A. Appendix

Well-Conditioned Basis: (Dasgupta et al., 2009) Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ a rank d matrix. For $p \geq 1$, it has a dual $q = p/(p-1)$. A matrix $\mathbf{U}$ is said to be (α, β, p) *well-conditioned basis* of $\mathbf{A}$, if $\mathbf{U}$ spans the column space of $\mathbf{A}$, $\sum_{i=1}^{d} \|\mathbf{u}_j\|_p \leq \alpha, \forall \mathbf{x} \in \mathbb{R}^d, \frac{\|\mathbf{x}\|_q}{\|\mathbf{U}\mathbf{x}\|_p} \leq \beta$ and (α, β) are $d^{O(1)}$ 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 $\forall i \in [n]$, $|x_i - \mathbb{E}[x_i]| \leq b$. 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 $t > 0$,*

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

Theorem A.2. ***Matrix Bernstein (Tropp et al., 2015)*** *Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are independent $d \times d$ random matrices such that $\forall i \in [n], \|\mathbf{X}_i\| \leq b$ and $var(\|\mathbf{X}\|) \leq \sigma^2$ where $\mathbf{X} = \sum_{i=1}^{n} \mathbf{X}_i$, then for some $t > 0$,*

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

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

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

Let $\|\Pi\mathbf{A}\mathbf{x}\|_p^p = \sum_{i=1}^{m} |\tilde{\mathbf{a}}_i^T \mathbf{x}|^p$, where Π is a sampling matrix which samples m rows from $\mathbf{A}$ with proper scaling. Now we argue this $\forall \mathbf{x} \in \mathbf{Q}$, i.e. $\|\Pi\mathbf{A}\mathbf{x}\|_p^p = (1 \pm \epsilon) \|\mathbf{A}\mathbf{x}\|_p^p$ which is same as $\|\Pi\mathbf{U}\mathbf{y}\|_p^p = (1 \pm \epsilon) \|\mathbf{U}\mathbf{y}\|_p^p$ where $\mathbf{U}\mathbf{y} = \mathbf{A}\mathbf{x}$. Now with an $\epsilon-$net, we argue $\forall \mathbf{x} \in \mathbf{Q}$, $\mathbf{Q} \subseteq \mathbb{R}^d$.

Let $\mathbf{B} = \{\mathbf{z} \in \mathbb{R}^n | \mathbf{z} = \mathbf{U}\mathbf{y}$ for some $y \in \mathbb{R}^k$ and $\|\mathbf{z}\|_p = 1\}$. From this set we intend to find a finite subset $\mathbf{N}$ which is an ϵ-net to the set. Now here we argue that if we can ensure $\|\Pi\mathbf{w}\|_p^p = (1 \pm \epsilon) \|\mathbf{w}\|_p^p, \forall \mathbf{w} \in \mathbf{N}$ then we can claim that $\|\Pi\mathbf{z}\|_p^p = (1 \pm \epsilon) \|\mathbf{z}\|_p^p, \forall \mathbf{z} \in \mathbf{B}$ which further imply that $\|\Pi\mathbf{A}\mathbf{x}\|_p^p = (1 \pm \epsilon) \|\mathbf{A}\mathbf{x}\|_p^p, \forall \mathbf{x} \in \mathbf{Q}$.

Let $\mathbf{v} \in \mathbf{B}$ whose closest ϵ-net point is $\mathbf{w}_1 \in \mathbf{N}$ such that $\|\mathbf{v} - \mathbf{w}_1\|_p \leq \epsilon$. Now note that,

$$\begin{aligned}
\|\Pi\mathbf{v}\|_p^p &= \|\Pi\mathbf{w}_1 + \Pi(\mathbf{v} - \mathbf{w}_1)\|_p^p \\
&\leq (\|\Pi\mathbf{w}_1\|_p + \|\Pi(\mathbf{v} - \mathbf{w}_1)\|_p)^p \\
&\leq (1 + \epsilon + \|\Pi(\mathbf{v} - \mathbf{w}_1)\|_p)^p \\
&= (1 + \epsilon + \|\Pi(\mathbf{w}_2/\alpha + \mathbf{v} - \mathbf{w}_1 - \mathbf{w}_2/\alpha)\|_p)^p \\
&\leq (1 + \epsilon + \epsilon(1+\epsilon) + \|\Pi(\mathbf{v} - \mathbf{w}_1 - \mathbf{w}_2/\alpha)\|_p)^p
\end{aligned}$$

Repeated application of this argument yields

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

By similar argument one can show that $\|\Pi\mathbf{v}\|_p^p \geq 1 - O(\epsilon)$. Finally by rescaling ϵ by some constant factor we achieve $\|\Pi\mathbf{z}\|_p^p \in 1 \pm \epsilon, \forall \mathbf{z} \in \mathbf{B}$.

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

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

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

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

$$(\mathbf{M} + \mathbf{x}\mathbf{x}^T)^\dagger = \mathbf{M}^\dagger - \frac{\mathbf{M}^\dagger \mathbf{x}\mathbf{x}^T \mathbf{M}^\dagger}{1 + \mathbf{x}^T\mathbf{M}^\dagger\mathbf{x}}$$

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

$$\begin{aligned}
(\mathbf{M} + \mathbf{x}\mathbf{x}^T)^\dagger &= (\mathbf{V}\Sigma\mathbf{V}^T + \mathbf{V}\mathbf{y}\mathbf{y}^T\mathbf{V}^T)^\dagger \\
&= \mathbf{V}(\Sigma + \mathbf{y}\mathbf{y}^T)^{-1}\mathbf{V}^T \\
&= \mathbf{V}\left(\Sigma^{-1} - \frac{\Sigma^{-1}\mathbf{y}\mathbf{y}^T\Sigma^{-1}}{1 + y^T\Sigma^{-1}\mathbf{y}}\right)\mathbf{V} \\
&= \mathbf{V}\left(\Sigma^{-1} - \frac{\Sigma^{-1}\mathbf{V}^T\mathbf{V}\mathbf{y}\mathbf{y}^T\mathbf{V}^T\mathbf{V}\Sigma^{-1}}{1 + y^T\mathbf{V}^T\mathbf{V}\Sigma^{-1}\mathbf{V}^T\mathbf{V}\mathbf{y}}\right)\mathbf{V} \\
&= \mathbf{M}^\dagger - \frac{\mathbf{M}^\dagger\mathbf{x}\mathbf{x}^T\mathbf{M}^\dagger}{1 + \mathbf{x}^T\mathbf{M}^\dagger\mathbf{x}}
\end{aligned}$$

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 Σ and the rank-1 update of $\mathbf{y}\mathbf{y}^T$. □

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,

$$\tilde{s}_i = \sup_{\mathbf{x}} \frac{|\mathbf{a}_i^T\mathbf{x}|^p}{\sum_{j=1}^{i}|\mathbf{a}_j^T\mathbf{x}|^p} = \sup_{\mathbf{y}} \frac{|\mathbf{u}_i^T\mathbf{y}|^p}{\sum_{j=1}^{i}|\mathbf{u}_j^T\mathbf{y}|^p}$$

Here $\mathbf{y} = \Sigma\mathbf{V}^T\mathbf{x}$ where $[\mathbf{U},\Sigma,\mathbf{V}] = \text{svd}(\mathbf{A})$ and $\mathbf{u}_i^T$ is the i^{th} row of $\mathbf{U}$. Now at this i^{th} step we also define $[\mathbf{U}_i,\Sigma_i,\mathbf{V}_i] = \text{svd}(\mathbf{A}_i)$. So with $\mathbf{y} = \Sigma_i\mathbf{V}_i^T\mathbf{x}$ and $\tilde{\mathbf{u}}_i^T$ is the i^{th} row of $\mathbf{U}_i$ we rewrite the above optimization function as follows,

$$\begin{aligned}\tilde{s}_i &= \sup_{\mathbf{x}} \frac{|\mathbf{a}_i^T\mathbf{x}|^p}{\sum_{j=1}^{i}|\mathbf{a}_j^T\mathbf{x}|^p} = \sup_{\mathbf{y}} \frac{|\tilde{\mathbf{u}}_i^T\mathbf{y}|^p}{\|\mathbf{U}_i\mathbf{y}\|_p^p} \\ &= \sup_{\mathbf{y}} \frac{|\tilde{\mathbf{u}}_i^T\mathbf{y}|^p}{|\tilde{\mathbf{u}}_i^T\mathbf{y}|^p + \sum_{j=1}^{i-1}|\tilde{\mathbf{u}}_j^T\mathbf{y}|^p}\end{aligned}$$

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

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

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

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

$$\begin{aligned}\sum_{j=1}^{i-1}|\tilde{\mathbf{u}}_j^T\mathbf{y}|^2 &= \left(\sum_{j=1}^{i-1}|\tilde{\mathbf{u}}_j^T\mathbf{y}|^2\cdot 1\right) \\ &\overset{(i)}{\leq} \left(\sum_{j=1}^{i-1}|\tilde{\mathbf{u}}_j^T\mathbf{y}|^{2p/2}\right)^{2/p}\left(\sum_{j=1}^{i-1}1^{p/(p-2)}\right)^{1-2/p} \\ &\overset{(ii)}{\leq} \left(\sum_{j=1}^{i-1}|\tilde{\mathbf{u}}_j^T\mathbf{y}|^p\right)^{2/p}\cdot(i)^{1-2/p}\end{aligned}$$

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

$$\begin{aligned}\left(\sum_{j=1}^{i-1}|\tilde{\mathbf{u}}_j^T\mathbf{y}|^p\right)^{2/p} &\geq \left(\frac{1}{i}\right)^{1-2/p}(1-|\tilde{\mathbf{u}}_i^T\mathbf{y}|^2) \\ \left(\sum_{j=1}^{i-1}|\tilde{\mathbf{u}}_j^T\mathbf{y}|^p\right) &\geq \left(\frac{1}{i}\right)^{p/2-1}(1-|\tilde{\mathbf{u}}_i^T\mathbf{y}|^2)^{p/2}\end{aligned}$$

So we get $\tilde{s}_i \leq \sup_{\mathbf{y}} \frac{|\tilde{\mathbf{u}}_i^T\mathbf{y}|^p}{|\tilde{\mathbf{u}}_i^T\mathbf{y}|^p + (1/i)^{p/2-1}(1-|\tilde{\mathbf{u}}_i^T\mathbf{y}|^2)^{p/2}}$. Note that this function increases with value of $|\tilde{\mathbf{u}}_i^T\mathbf{y}|$, which is maximum when $\mathbf{y} = \frac{\tilde{\mathbf{u}}_i}{\|\tilde{\mathbf{u}}_i\|}$, which gives,

$$\tilde{s}_i \leq \frac{\|\tilde{\mathbf{u}}_i\|^p}{\|\tilde{\mathbf{u}}_i\|^p + (1/i)^{p/2-1}(1-\|\tilde{\mathbf{u}}_i\|^2)^{p/2}}$$

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

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 $i^{p/2-1}$ for every row. Although we have bound on the $\sum_i^n \tilde{l}_i$ from lemma 5.3, but this factor can be very huge as i increases which eventually sets many $\tilde{l}_i = 1$.

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

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

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

$$\begin{aligned}p_i &= r\tilde{l}_i/\sum_{j=1}^{i}\tilde{l}_j \geq \frac{r|\mathbf{u}_i^T\mathbf{y}|^p}{\sum_{j=1}^{i}\tilde{l}_j\sum_{j=1}^{i}|\mathbf{u}_j^T\mathbf{y}|^p} \\ &\geq \frac{r|\mathbf{u}_i^T\mathbf{y}|^p}{\sum_{j=1}^{n}\tilde{l}_j\sum_{j=1}^{n}|\mathbf{u}_j^T\mathbf{y}|^p}\end{aligned}$$

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 $|w_i - \mathbb{E}[w_i]| \leq \frac{|\mathbf{u}_i^T\mathbf{y}|^p}{p_i} \leq \frac{|\mathbf{u}_i^T\mathbf{y}|^p(\sum_{j=1}^n \tilde{l}_j)\sum_{j=1}^n |\mathbf{u}_j^T\mathbf{y}|^p}{r|\mathbf{u}_i^T\mathbf{y}|^p} = \frac{(\sum_{j=1}^n \tilde{l}_j)\sum_{j=1}^n |\mathbf{u}_j^T\mathbf{y}|^p}{r}$. Note for $p_i = 1, |w_i - \mathbb{E}[w_i]| = 0$.

Case 2: When w_i is 0 then $p_i < 1$. So we have $1 > \frac{r\tilde{l}_i}{\sum_{j=1}^i \tilde{l}_j} \geq \frac{r|\mathbf{u}_i^T\mathbf{y}|^p}{(\sum_{j=1}^n \tilde{l}_j)\sum_{j=1}^n |\mathbf{u}_j^T\mathbf{y}|^p}$. So $|w_i - \mathbb{E}[w_i]| = |\mathbb{E}[w_i]| = |(\mathbf{u}_i^T\mathbf{y})^p| < \frac{(\sum_{j=1}^n \tilde{l}_j)\sum_{j=1}^n |\mathbf{u}_j^T\mathbf{y}|^p}{r}$.

So by setting $b = \frac{(\sum_{j=1}^n \tilde{l}_j)\sum_{j=1}^n |\mathbf{u}_j^T\mathbf{y}|^p}{r}$ we can bound the term $|w_i - \mathbb{E}[w_i]|$. Next we bound the variance of the sum, i.e. $\sum_{i=1}^n \tilde{l}_i$. Let $\sigma^2 = \text{var}\big(\sum_{i=1}^n w_i\big) = \sum_{i=1}^n \sigma_i^2$, since every incoming rows are independent of each other and here we consider $\sigma_i^2 = \text{var}(w_i)$

$$\begin{aligned}
\sigma^2 &= \sum_{i=1}^n \sigma_i^2 = \sum_{i=1}^n \mathbb{E}[w_i^2] - (\mathbb{E}[w_i])^2 \\
&\leq \sum_{i=1}^n \frac{|\mathbf{u}_i^T\mathbf{y}|^{2p}}{p_i} \\
&\leq \sum_{i=1}^n \frac{|\mathbf{u}_i^T\mathbf{y}|^{2p}(\sum_{k=1}^n \tilde{l}_k)\sum_{j=1}^n |\mathbf{u}_j^T\mathbf{y}|^p}{r|\mathbf{u}_i^T\mathbf{y}|^p} \\
&\leq \frac{(\sum_{k=1}^n \tilde{l}_k)(\sum_{j=1}^n |\mathbf{u}_j^T\mathbf{y}|^p)^2}{r}
\end{aligned}$$

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

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

which we bound as follows,

$$\begin{aligned}
\mathbb{P} &\leq \exp\left(\frac{\left(\epsilon\sum_{j=1}^n |\mathbf{u}_j^T\mathbf{y}|^p\right)^2}{2\sigma^2 + bt/3}\right) \\
&\leq \exp\left(\frac{-r\epsilon^2(\|\mathbf{U}\mathbf{y}\|_p^p)^2}{(\|\mathbf{U}\mathbf{y}\|_p^p)^2\sum_{j=1}^n \tilde{l}_j(2+\epsilon/3)}\right) \\
&= \exp\left(\frac{-r\epsilon^2}{(2+\epsilon/3)\sum_{j=1}^n \tilde{l}_j}\right)
\end{aligned}$$

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

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

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

and follow the above proof. Finally by setting r as $O(\frac{k\sum_{j=1}^n \tilde{l}_j}{\epsilon^2})$ one can get

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

Since for 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 ℓ_p subspace embedding as in equation (2). □

Now we give the detail proof of sum of upper bounds of sensitivity scores, $\sum_{i=1}^n \tilde{l}_i$. 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 $\mathbf{A}_i$ denotes the $i \times d$ matrix of the first i incoming rows. `LineFilter` maintains the covariance matrix $\mathbf{M}$. At the $(i-1)^{th}$ step we have $\mathbf{M} = \mathbf{A}_{i-1}^T\mathbf{A}_{i-1}$. This is then used to define the score $\tilde{l}_i$ for the next step i, as $\tilde{l}_i = \min\{i^{p/2-1}\tilde{e}_i^{p/2}, 1\}$, where $\tilde{e}_i = \mathbf{a}_i^T(\mathbf{M}+\mathbf{a}_i\mathbf{a}_i^T)^\dagger\mathbf{a}_i = \mathbf{a}_i^T(\mathbf{A}_i^T\mathbf{A}_i)^\dagger\mathbf{a}_i$ and $\mathbf{a}_i^T$ is the i^{th} row. The scores $\tilde{e}_i$ are also called online leverage scores. We first give a bound on $\sum_{i=1}^n \tilde{e}_i$. A similar bound is given in the online matrix row sampling by (Cohen et al., 2016), albeit for a regularized version of the scores $\tilde{e}_i$. As the rows are coming, the rank of $\mathbf{M}$ increases from 1 to at most d. We say that the algorithm is in phase-k if the rank of $\mathbf{M}$ equals k. For each phase $k \in [1, d-1]$, let i_k denote the index where row $\mathbf{a}_{i_k}$ caused a phase-change in $\mathbf{M}$ i.e. rank of $(\mathbf{A}_{i_k-1}^T\mathbf{A}_{i_k-1})$ is $k-1$, while rank of $(\mathbf{A}_{i_k}^T\mathbf{A}_{i_k})$ is k. For each such i_k, the online leverage score $\tilde{e}_{i_k} = 1$, since row $\mathbf{a}_{i_k}$ does not lie in the row space of $\mathbf{A}_{i_k-1}$. There are at most d such indices i_k.

We now bound the $\sum_{i\in[i_k,i_{k+1}-1]} \tilde{e}_i$. Suppose the thin-SVD$(\mathbf{A}_{i_k}^T\mathbf{A}_{i_k}) = \mathbf{V}\Sigma_{i_k}\mathbf{V}^T$, all entries in Σ_{i_k} being positive. Furthermore, for any i in this phase, i.e. for $i \in [i_k+1, i_{k+1}-1]$, $\mathbf{V}$ forms the basis of the row space of $\mathbf{A}_i$. Define $\mathbf{X}_i = \mathbf{V}^T(\mathbf{A}_i^T\mathbf{A}_i)\mathbf{V}$ and the i^{th} row $\mathbf{a}_i = \mathbf{V}\mathbf{b}_i$. Notice that each $\mathbf{X}_i \in \mathbb{R}^{k\times k}$, and $\mathbf{X}_{i_k} = \Sigma_{i_k}$. Also, $\mathbf{X}_{i_k}$ is positive definite. Now for each $i \in [i_k+1, i_{k+1}-1]$, we have $\mathbf{X}_i = \mathbf{X}_{i-1} + \mathbf{b}_i\mathbf{b}_i^T$.

So we have, $\tilde{e}_i = \mathbf{a}_i^T(\mathbf{A}_i^T\mathbf{A}_i)^\dagger\mathbf{a}_i = \mathbf{b}_i^T\mathbf{V}^T(\mathbf{V}(\mathbf{X}_{i-1} + \mathbf{b}_i\mathbf{b}_i^T)\mathbf{V}^T)^\dagger\mathbf{V}\mathbf{b}_i = \mathbf{b}_i^T(\mathbf{X}_{i-1} + \mathbf{b}_i\mathbf{b}_i^T)^{-1}\mathbf{b}_i$ where the

last equality uses the invertibility of the matrix. Since $\mathbf{X}_{i-1}$ is not rank deficient so by using matrix determinant lemma (Harville, 1997) on $\det(\mathbf{X}_{i-1}+\mathbf{b}_i\mathbf{b}_i^T)$ we get,

$$\begin{aligned} &= \det(\mathbf{X}_{i-1})(1+\mathbf{b}_i^T(\mathbf{X}_{i-1})^{-1}\mathbf{b}_i) \\ &\overset{(i)}{\geq} \det(\mathbf{X}_{i-1})(1+\mathbf{b}_i^T(\mathbf{X}_{i-1}+\mathbf{b}_i\mathbf{b}_i^T)^{-1}\mathbf{b}_i) \\ &= \det(\mathbf{X}_{i-1})(1+\tilde{e}_i) \\ &\overset{(ii)}{\geq} \det(\mathbf{X}_{i-1})\exp(\tilde{e}_i/2) \\ \exp(\tilde{e}_i/2) &\leq \frac{\det(\mathbf{X}_{i-1}+\mathbf{b}_i\mathbf{b}_i^T)}{\det(\mathbf{X}_{i-1})} \end{aligned}$$

Inequality (i) follows as $\mathbf{X}_{i-1}^{-1}-(\mathbf{X}_{i-1}+\mathbf{b}\mathbf{b}^T)^{-1}\succeq 0$ (i.e. p.s.d.). Inequality (ii) follows from the fact that $1+x\geq\exp(x/2)$ for $x\leq 1$. Now with $\tilde{e}_{i_k}=1$, we analyze the product of the remaining terms of the phase k i.e.,

$$\textstyle\prod_{i\in[i_k+1,i_{k+1}-1]}\exp(\tilde{e}_i/2)$$

which is,

$$\leq \prod_{i\in[i_k+1,i_{k+1}-1]}\frac{\det(\mathbf{X}_i)}{\det(\mathbf{X}_{i-1})}\leq\frac{\det(\mathbf{X}_{i_{k+1}-1})}{\det(\mathbf{X}_{i_k+1})}.$$

Now by taking the product over all phases the term $\exp\left(\sum_{i\in[1,i_d-1]}\tilde{e}_i/2\right)$ gets,

$$\begin{aligned} &= \exp((d-1)/2)\Big(\prod_{k\in[1,d-1]}\prod_{i\in[i_k+1,i_{k+1}-1]}\exp(\tilde{e}_i/2)\Big) \\ &= \exp((d-1)/2)\Big(\prod_{k\in[1,d-1]}\frac{\det(\mathbf{X}_{i_{k+1}-1})}{\det(\mathbf{X}_{i_k+1})}\Big) \\ &= \exp((d-1)/2)\Big(\frac{\det(X_{i_2-1})}{\det(X_{i_1})}\prod_{k\in[2,d-1]}\frac{\det(\mathbf{X}_{i_{k+1}-1})}{\det(\mathbf{X}_{i_k+1})}\Big) \end{aligned}$$

Because we know that for any phase k we have $(\mathbf{A}_{i_{k+1}-1}^T\mathbf{A}_{i_{k+1}-1})\succeq(\mathbf{A}_{i_k+1}^T\mathbf{A}_{i_k+1})$ so we get, $\det(\mathbf{X}_{i_{k+1}-1})\geq\det(\mathbf{X}_{i_k+1})$. Further between inter phases terms, i.e. between the last term of phase $k-1$ and the second term of phase k we have $\det(\mathbf{X}_{i_k-1})\leq\det(\mathbf{X}_{i_k+1})$. 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-1 yields a telescopic product, which gives,

$$\begin{aligned}\exp\left(\sum_{i\in[1,i_d-1]}\tilde{e}_i/2\right)&\leq\frac{\exp((d-1)/2)\det(\mathbf{X}_{i_d-1})}{\det(\mathbf{X}_{i_1+1})}\\&\leq\frac{\exp((d-1)/2)\det(\mathbf{A}_{i_d}^T\mathbf{A}_{i_d})}{\det(\mathbf{X}_{i_1+1})}.\end{aligned}$$

Furthermore, we know $\tilde{e}_{i_d}=1$, 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,

$$\begin{aligned}\exp\left(\sum_{i\in[i_d,n]}\tilde{e}_i/2\right)&\leq\frac{\exp(1/2)\det(\mathbf{A}^T\mathbf{A})}{\det(\mathbf{A}_{i_d+1}^T\mathbf{A}_{i_d+1})}\\&\leq\frac{\exp(1/2)\|\mathbf{A}\|^d}{\det(\mathbf{A}_{i_d+1}^T\mathbf{A}_{i_d+1})}\end{aligned}$$

Let $\mathbf{a}_{i_1+1}$ 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 $k\in[2,d]$,

$$\begin{aligned}\sum_{i\leq n}\tilde{e}_i&\leq d/2+2d\log\|\mathbf{A}\|-2\log\|\mathbf{a}_{i_1+1}\|\\&\leq d/2+2d\log\|\mathbf{A}\|-\min_i 2\log\|\mathbf{a}_i\|.\end{aligned}$$

Now, we give a bound on $\sum_{i=1}^n\tilde{l}_i$ where $\tilde{l}_i=\min\{1,i^{p/2-1}\tilde{e}_i^{p/2}\}\leq\min\{1,n^{p/2-1}\tilde{e}_i^{p/2}\}$. We consider two cases. When $\tilde{e}_i^{p/2}\geq n^{1-p/2}$ then $\tilde{l}_i=1$, this implies that $\tilde{e}_i\geq n^{2/p-1}$. But we know $\sum_{i=1}^n\tilde{e}_i\leq O(d+d\log\|\mathbf{A}\|-\min_i\log\|\mathbf{a}_i\|)$ and hence there are atmost $O(n^{1-2/p}(d+d\log\|\mathbf{A}\|-\min_i\log\|\mathbf{a}_i\|))$ indices with $\tilde{l}_i=1$. Now for the case where $\tilde{e}_i^{p/2}<n^{1-p/2}$, we get $\tilde{e}_i^{p/2-1}\leq(n)^{(1-p/2)(1-2/p)}$. Then $\sum_{i=1}^n n^{p/2-1}\tilde{e}_i^{p/2}=\sum_{i=1}^n n^{p/2-1}\tilde{e}_i^{p/2-1}\tilde{e}_i\leq\sum_{i=1}^n n^{1-2/p}\tilde{e}_i$ is $O(n^{1-2/p}(d+d\log\|\mathbf{A}\|-\min_i\log\|\mathbf{a}_i\|))$. □

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 ϵ 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 feed to the streaming version of the algorithm in (Cohen & Peng, 2015), i.e. `StreamingLW` for ℓ_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 $\mathbf{P}$ of size n takes $O(nd^C\log n)$ time to return a coreset $\mathbf{Q}$ of size $O(d^{p/2}(\log d)\epsilon^{-5})$ where C is a constant. Note that for the `StreamingLW` in section 7 of (Har-Peled & Mazumdar, 2004) we set $M=O(d^{p/2}(\log d)\epsilon^{-5})$. The method

returns $\mathbf{Q}_i$ as the $(1+\delta_i)$ coreset for the partition $\mathbf{P}_i$ where $|\mathbf{P}_i|$ is either $2^i M$ or 0, here $\rho_j = \epsilon/(c(j+1)^2)$ such that $1+\delta_i = \prod_{j=0}^{i}(1+\rho_j) \leq 1+\epsilon/2, \forall j \in \lceil \log n \rceil$. Thus we have $|\mathbf{Q}_i|$ 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 $O(d^{p/2}(\log^{11} n)(\log d)\epsilon^{-5})$. The `StreamingLW` never actually uses the entire P_i and run offline Lewis Weight based sampling. Instead it uses all Q_j , where $j < i$. Now the amortized time spent per update is,

$$= \sum_{i=1}^{\lceil \log(n/M) \rceil} \frac{1}{2^i M}(|\mathbf{Q}_i| d^C \log |\mathbf{Q}_i|) \\ \sum_{i=1}^{\lceil \log(n/M) \rceil} \frac{1}{2^i M}(M(i+1)^4 d^C \log |\mathbf{Q}_i|) \leq d^C p \log d$$

So finally the algorithm return $\mathbf{Q}$ as the final coreset of $O(d^{p/2}(\log^{10} n)(\log d)\epsilon^{-5})$ rows and uses $O(d^C p \log d)$ 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 $\tilde{O}(n^{1-2/p}d)$, 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 $M = O(d^{p/2}(\log d)\epsilon^{-5})$. The method returns $\mathbf{Q}_i$ as the $(1+\delta_i)$ coreset for the partition $\mathbf{P}_i$ where $|\mathbf{P}_i|$ is either $2^i M$ or 0, here $\rho_j = \epsilon/(c(j+1)^2)$ such that $1+\delta_i = \prod_{j=0}^{i}(1+\rho_j) \leq 1+\epsilon/2, \forall j \in \lceil \log n \rceil$. Thus we have $|\mathbf{Q}_i|$ is $O(d^{p/2}(\log d)(i+1)^{10}\epsilon^{-5})$. Hence the total working space is $O((1-2/p)^{11}d^{p/2}(\log^{11} n)(\log d)\epsilon^{-5})$. So finally `LineFilter+StreamingLW` returns a coreset $\mathbf{Q}$ 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 depending on the value of p.

A.4. Proof of Lemma 4.2

Proof. The term $|\mathbf{x}^T\mathbf{y}|^p = |\mathbf{x}^T\mathbf{y}|^{\lfloor p/2 \rfloor}|\mathbf{x}^T\mathbf{y}|^{\lceil p/2 \rceil}$. We define $|\mathbf{x}^T\mathbf{y}|^{\lfloor p/2 \rfloor} = |\grave{\mathbf{x}}_i^T\grave{\mathbf{y}}| = |\langle \mathbf{x}^{\otimes \lfloor p/2 \rfloor}, \mathbf{y}^{\otimes \lfloor p/2 \rfloor} \rangle|^2$ and $|\mathbf{x}^T\mathbf{y}|^{\lceil p/2 \rceil} = |\acute{\mathbf{x}}_i^T\acute{\mathbf{y}}| = |\langle \mathbf{x}^{\otimes \lceil p/2 \rceil}, \mathbf{y}^{\otimes \lceil p/2 \rceil} \rangle|^2$. Here the $\grave{\mathbf{x}}$ and $\acute{\mathbf{x}}$ are the higher dimensional representation of $\mathbf{x}$ and similarly $\grave{\mathbf{y}}$ and $\acute{\mathbf{y}}$ are defined from $\mathbf{y}$. For even valued p we know $\lfloor p/2 \rfloor = \lceil p/2 \rceil$, so for simplicity we write as $|\mathbf{x}^T\mathbf{y}|^{p/2} = |\acute{\mathbf{x}}_i^T\acute{\mathbf{y}}|$. Hence we get $|\mathbf{x}^T\mathbf{y}|^p = |\langle \mathbf{x}^{\otimes p/2}, \mathbf{y}^{\otimes p/2} \rangle|^2 = |\acute{\mathbf{x}}^T\acute{\mathbf{y}}|^2$ which is same as in (Schechtman, 2011). Here the vector $\acute{\mathbf{x}}$ is the higher dimensional vector, where $\acute{\mathbf{x}} = \text{vec}(\mathbf{x}^{\otimes p/2}) \in \mathbb{R}^{p/2}$ and similarly $\acute{\mathbf{y}}$ is also defined from $\mathbf{y}$. Now for odd value of p we have $\grave{\mathbf{x}} = \text{vec}(\mathbf{x}^{\otimes (p-1)/2}) \in \mathbb{R}^{(p-1)/2}$ and $\acute{\mathbf{x}} = \text{vec}(\mathbf{x}^{\otimes (p+1)/2}) \in \mathbb{R}^{(p+1)/2}$. Similarly $\grave{\mathbf{y}}$ and $\acute{\mathbf{y}}$ are defined from $\mathbf{y}$. Further note that $|\grave{\mathbf{x}}^T\grave{\mathbf{y}}| = |\acute{\mathbf{x}}^T\acute{\mathbf{y}}|^{(p-1)/(p+1)}$ which gives $|\mathbf{x}^T\mathbf{y}|^p = |\langle \mathbf{x}^{\otimes (p-1)/2}, \mathbf{y}^{\otimes (p-1)/2} \rangle| \cdot |\langle \mathbf{x}^{\otimes (p+1)/2}, \mathbf{y}^{\otimes (p+1)/2} \rangle| = |\grave{\mathbf{x}}^T\grave{\mathbf{y}}| \cdot |\acute{\mathbf{x}}^T\acute{\mathbf{y}}| = |\acute{\mathbf{x}}^T\acute{\mathbf{y}}|^{2p/(p+1)}$. 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 $\tilde{s}_i$ for each point i as follows,

$$\tilde{s}_i = \sup_{\{\mathbf{x} | \|\mathbf{x}\|=1\}} \frac{|\mathbf{a}_i^T\mathbf{x}|^p}{\|\mathbf{A}_i\mathbf{x}\|_p^p}$$

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

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

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 $\tilde{s}_i \leq \|\acute{\mathbf{u}}_i\|^2$ for even value p as defined in `KernelFilter`.

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

$$\begin{aligned}
\tilde{s}_i &= \sup_{\{\mathbf{x}|\|\mathbf{x}\|=1\}} \frac{|\mathbf{a}_i^T\mathbf{x}|^p}{\|\mathbf{A}_i\mathbf{x}\|_p^p} \\
&\overset{i}{=} \sup_{\{\acute{\mathbf{x}}|\|\acute{\mathbf{x}}\|=1\}} \frac{|\acute{\mathbf{a}}_i^T\acute{\mathbf{x}}|^{2p/(p+1)}}{\sum_{j\leq i}|\acute{\mathbf{a}}_j^T\acute{\mathbf{x}}|^{2p/(p+1)}} \\
&= \sup_{\{\acute{\mathbf{x}}|\|\acute{\mathbf{x}}\|=1\}} \frac{|\acute{\mathbf{a}}_i^T\acute{\mathbf{x}}|^{2p/(p+1)}}{\|\acute{\mathbf{A}}_i\acute{\mathbf{x}}\|_{2p/(p+1)}^{2p/(p+1)}} \\
&= \sup_{\{\acute{\mathbf{y}}|\|\acute{\mathbf{y}}\|=1\}} \frac{|\acute{\mathbf{u}}_i^T\acute{\mathbf{y}}|^{2p/(p+1)}}{\|\acute{\mathbf{U}}_i\acute{\mathbf{y}}\|_{2p/(p+1)}^{2p/(p+1)}} \\
&\overset{ii}{\leq} \sup_{\{\acute{\mathbf{y}}|\|\acute{\mathbf{y}}\|=1\}} \frac{|\acute{\mathbf{u}}_i^T\acute{\mathbf{y}}|^{2p/(p+1)}}{\|\acute{\mathbf{U}}_i\acute{\mathbf{y}}\|^{2p/(p+1)}} \\
&= \sup_{\{\acute{\mathbf{y}}|\|\acute{\mathbf{y}}\|=1\}} |\acute{\mathbf{u}}_i^T\acute{\mathbf{y}}|^{2p/(p+1)} \\
&= \|\acute{\mathbf{u}}_i\|^{2p/(p+1)}
\end{aligned}$$

The equality (i) is by lemma 4.2. Next with similar assumption as above let $[\acute{\mathbf{U}}_i, \acute{\mathbf{\Sigma}}_i, \acute{\mathbf{V}}_i] = \text{svd}(\acute{\mathbf{A}}_i)$. The inequality (ii) is because $\|\acute{\mathbf{U}}_i\acute{\mathbf{y}}\|_{2p/(p+1)} \geq \|\acute{\mathbf{U}}_i\acute{\mathbf{y}}\|$ and finally we get $\tilde{s}_i \leq \tilde{l}_i$ 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 ℓ_p subspace embedding. Now for some fixed $\mathbf{x} \in \mathbb{R}^d$ consider the following random variable for every row i.

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

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 $\text{var}(W) \leq \sigma^2$.

Now for the i^{th} timestamp `KernelFilter` defines $p_i = \min\{1, r\tilde{l}_i/\sum_{j\leq i}\tilde{l}_j\}$ where r is some constant. If $p_i = 1$ then $|w_i| = 0$, else if $p_i < 1$ and `KernelFilter` samples the row then $|w_i| \leq |\mathbf{a}_i^T\mathbf{x}|^p/p_i = |\mathbf{a}_i^T\mathbf{x}|^p \sum_{j=1}^i \tilde{l}_j/(r\tilde{l}_i) \leq \|\mathbf{A}_i\mathbf{x}\|_p^p |\mathbf{a}_i^T\mathbf{x}|^p \sum_{j=1}^i \tilde{l}_j/(r|\mathbf{a}_i^T\mathbf{x}|^p) \leq \|\mathbf{A}\mathbf{x}\|_p^p \sum_{j=1}^n \tilde{l}_j/r$. Next when `KernelFilter` does not sample the i^{th} row, it means that $p_i < 1$, then we have $1 > r\tilde{l}_i/\sum_{j=1}^i \tilde{l}_j \geq r|\mathbf{a}_i^T\mathbf{x}|^p/(\|\mathbf{A}_i\mathbf{x}\|_p^p \sum_{j=1}^i \tilde{l}_j) \geq r|\mathbf{a}_i^T\mathbf{x}|^p/(\|\mathbf{A}\mathbf{x}\|_p^p \sum_{j=1}^n \tilde{l}_j)$. Finally we get $|\mathbf{a}_i^T\mathbf{x}|^p \leq \|\mathbf{A}\mathbf{x}\|_p^p \sum_{j=1}^n \tilde{l}_j/r$. So for each i we get $|w_i| \leq \|\mathbf{A}\mathbf{x}\|_p^p \sum_{j=1}^n \tilde{l}_j/r$.

Next we bound the variance of sum of the random variable, i.e. $W = \sum_{i=1}^n w_i$. Let, $\sigma^2 = \text{var}(W) = \sum_{i=1}^n \text{var}(w_i) = \sum_{i=1}^n \mathbb{E}[w_i^2]$ as follows,

$$\begin{aligned}
\sigma^2 &= \sum_{i=1}^n \mathbb{E}[w_i^2] = \sum_{i=1}^n |\mathbf{a}_i^T\mathbf{x}|^{2p}/p_i \\
&\leq \sum_{i=1}^n |\mathbf{a}_i^T\mathbf{x}|^{2p} \sum_{j=1}^i \tilde{l}_j/(r\tilde{l}_i) \\
&= \|\mathbf{A}_i\mathbf{x}\|_p^p \sum_{i=1}^n |\mathbf{a}_i^T\mathbf{x}|^{2p} \sum_{j=1}^i \tilde{l}_j/(r|\mathbf{a}_i^T\mathbf{x}|^p) \\
&\leq \|\mathbf{A}\mathbf{x}\|_p^{2p} \sum_{j=1}^n \tilde{l}_j/r
\end{aligned}$$

Now we can apply Bernstein A.1 to bound the probability $\mathbb{P} = \Pr(|W| \geq \epsilon\|\mathbf{A}\mathbf{x}\|_p^p)$, Here we have $b = \|\mathbf{A}\mathbf{x}\|_p^p \sum_{j=1}^n \tilde{l}_j/r, \sigma^2 = \|\mathbf{A}\mathbf{x}\|_{2p}^p \sum_{j=1}^n \tilde{l}_j/r$ and we set $t = \epsilon\|\mathbf{A}\mathbf{x}\|_p^p$, then we get

$$\begin{aligned}
\mathbb{P} &\leq \exp\left(\frac{-(\epsilon\|\mathbf{A}\mathbf{x}\|_p^p)^2}{2\|\mathbf{A}\mathbf{x}\|_p^{2p}\sum_{j=1}^n \tilde{l}_j/r + \epsilon\|\mathbf{A}\mathbf{x}\|_p^{2p}\sum_{j=1}^n \tilde{l}_j/3r}\right) \\
&= \exp\left(\frac{-r\epsilon^2\|\mathbf{A}\mathbf{x}\|_p^{2p}}{(2+\epsilon/3)\|\mathbf{A}\mathbf{x}\|_p^{2p}\sum_{j=1}^n \tilde{l}_j}\right) \\
&= \exp\left(\frac{-r\epsilon^2}{(2+\epsilon/3)\sum_{j=1}^n \tilde{l}_j}\right)
\end{aligned}$$

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

Now to ensure the guarantee for tensor contraction as equation (1) one can define

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

and follow the above proof. By setting the $r = O(k\epsilon^{-2}\sum_{j=1}^n \tilde{l}_j)$ one can get following $\forall \mathbf{x} \in \mathbf{Q}$,

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

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 ℓ_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 $\acute{c}_i = \|\acute{\mathbf{u}}_i\|$. Now for even value p we have $\sum_{i=1}^n \tilde{l}_i = \sum_{i=1}^n \acute{c}_i^2$. From lemma 5.3 we get $\sum_{i=1}^n \acute{c}_i^2$ is $O(d^{p/2}(1 + \log\|\acute{\mathbf{A}}\| - d^{-p/2}\min_i \log\|\acute{\mathbf{a}}_i\|)$. Now with $[\mathbf{u}, \mathbf{\Sigma}, \mathbf{V}] = \text{svd}(\mathbf{A})$ we have $\acute{\mathbf{a}}^T = \text{vec}(\mathbf{a}_i^T \otimes^{p/2}) = \text{vec}((\mathbf{u}_i^T \mathbf{\Sigma} \mathbf{V}^T)^{p/2})$. So we get $\|\acute{\mathbf{A}}\| \leq \sigma_1^{p/2}$. Hence $\sum_{i=1}^n \tilde{l}_i$ is $O(d^{p/2}(1+p(\log\|\mathbf{A}\| - d^{-p/2}\min_i \log\|\mathbf{a}_i\|)))$.

Now for the odd p case $\sum_{i=1}^n \tilde{l}_i = \sum_{i=1}^n \acute{c}_i^{2p/(p+1)}$. From lemma 5.3 we get $\sum_{i=1}^n \acute{c}_i^2$ is $O(d^{\lceil p/2 \rceil}(1 + \log\|\acute{\mathbf{A}}\| - d^{-\lceil p/2 \rceil}\min_i \log\|\acute{\mathbf{a}}_i\|)$. Now with $[\mathbf{u}, \mathbf{\Sigma}, \mathbf{V}] = \text{svd}(\mathbf{A})$ we have $\acute{\mathbf{a}}^T = \text{vec}(\mathbf{a}_i^T \otimes^{\lceil p/2 \rceil}) = \text{vec}((\mathbf{u}_i^T \mathbf{\Sigma} \mathbf{V}^T)^{\lceil p/2 \rceil})$. So we get $\|\acute{\mathbf{A}}\| \leq \sigma_1^{(p+1)/2}$. Hence $\sum_{i=1}^n \acute{c}_i^2$ is $O(d^{\lceil p/2 \rceil}(1 + (p+1)(\log\|\mathbf{A}\| - d^{-\lceil p/2 \rceil}\min_i \log\|\mathbf{a}_i\|)))$. Now let $\acute{\mathbf{c}}$ is a vector with each index $\acute{\mathbf{c}}_i$ is defined as above. Then in this case we have $\sum_{i=1}^n \tilde{l}_i = \|\acute{\mathbf{c}}\|_{2p/(p+1)}^{2p/(p+1)} \leq n^{1/(p+1)}\|\acute{\mathbf{c}}\|^{2p/(p+1)}$ which is $O(n^{1/(p+1)}d^{p/2}(1 + (p+1)(\log\|\mathbf{A}\| - d^{-\lceil p/2 \rceil}\min_i \log\|\mathbf{a}_i\|))^{p/(p+1)})$. □

Now for `LineFilter+KernelFilter` first we pass every incoming row to `LineFilter` by setting r such that a coreset returned from it will give an $1/3$ approximation factor. Next when we pass the sampled rows to `KernelFilter`, there we set r to get the final coreset $\mathbf{C}$ with ϵ approximation. Note that due a factor of n in the coreset size for odd value p returned by `KernelFilter`, hence we get a $O(n^{(p-2)/(p^2+p)})$ along with extra factors of $(dk)^{1/4}$ in the final coreset size.

A.5. $p = 2$ case

In this subsection we state the guarantees of our algorithm in the matrix case, where the rows of the matrix are coming in online manner. First we give a corollary stating one would get by following the analysis mentioned above, i.e. by using the scalar Bernstein inequality A.1.

Corollary A.1. *Given a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ with rows coming one at a time, for $p = 2$ our algorithm uses $O(d^2)$ update time and samples $O(\frac{d}{\epsilon^2}(d + d\log\|\mathbf{A}\| - \min_i \log\|\mathbf{a}_i\|))$ rows and preserves the following with probability at least 0.99, $\forall \mathbf{x} \in \mathbb{R}^d$ $(1-\epsilon)\|\mathbf{A}\mathbf{x}\|^2 \leq \|\mathbf{C}\mathbf{x}\|^2 \leq (1+\epsilon)\|\mathbf{A}\mathbf{x}\|^2$.*

Just by using Matrix Bernstein inequality (Tropp et al., 2011) we can slightly improve the sampling complexity from factor of $O(d^2)$ to factor of $O(d\log d)$. For simplicity we modify the sampling probability to $p_i = \min\{r\tilde{l}_i, 1\}$ and get the following guarantee.

Theorem A.3. *The above modified algorithm samples $O(\frac{d\log d}{\epsilon^2}(1 + \log\|\mathbf{A}\| - d^{-1}\min_i \log\|\mathbf{a}_i\|))$ rows and preserves the following with probability at least 0.99, $\forall \mathbf{x} \in \mathbb{R}^d$*

$$(1-\epsilon)\|\mathbf{A}\mathbf{x}\|^2 \leq \|\mathbf{C}\mathbf{x}\|^2 \leq (1+\epsilon)\|\mathbf{A}\mathbf{x}\|^2$$

Proof. We prove this theorem in 2 parts. First we show that sampling $\mathbf{a}_i$ with probability $p_i = \min\{r\tilde{l}_i, 1\}$ where $\tilde{l}_i = \mathbf{a}_i^T(\mathbf{A}_i^T\mathbf{A}_i)^\dagger\mathbf{a}_i$ preserves $\|\mathbf{C}^T\mathbf{C}\| \leq (1 \pm \epsilon)\|\mathbf{A}^T\mathbf{A}\|$. Next we give the bound on expected sample size.

We define, $\mathbf{u}_i = (\mathbf{A}^T\mathbf{A})^{-1/2}\mathbf{a}_i$ and we define a random matrix $\mathbf{W}_i$ corresponding to each streaming row as,

$$\mathbf{W}_i = \begin{cases} (1/p_i - 1)\mathbf{u}_i\mathbf{u}_i^T & \text{with probability } p_i \\ -\mathbf{u}_i\mathbf{u}_i^T & \text{with probability } (1-p_i) \end{cases}$$

Now we have,

$$\begin{aligned} \tilde{l}_i &= \mathbf{a}_i^T(\mathbf{A}_{i-1}^T\mathbf{A}_{i-1} + \mathbf{a}_i\mathbf{a}_i^T)^\dagger\mathbf{a}_i \\ &\geq \mathbf{a}_i^T(\mathbf{A}^T\mathbf{A})^\dagger\mathbf{a}_i = \mathbf{u}_i^T\mathbf{u}_i \end{aligned}$$

For $p_i \geq \min\{r\mathbf{u}_i^T\mathbf{u}_i, 1\}$, if $p_i = 1$, then $\|\mathbf{W}_i\| = 0$, else $p_i = r\mathbf{u}_i^T\mathbf{u}_i < 1$. So we get $\|\mathbf{W}_i\| \leq 1/r$. Next we bound $\mathbb{E}[\mathbf{W}_i^2]$.

$$\begin{aligned} \mathbb{E}[\mathbf{W}_i^2] &= p_i(1/p_i - 1)^2(\mathbf{u}_i\mathbf{u}_i^T)^2 + (1-p_i)(\mathbf{u}_i\mathbf{u}_i^T)^2 \\ &\preceq (\mathbf{u}_i\mathbf{u}_i^T)^2/p_i \preceq \mathbf{u}_i\mathbf{u}_i^T/r \end{aligned}$$

Let $\mathbf{W} = \sum_{i=1}^n \mathbf{W}_i$, then variance of $\|\mathbf{W}\|$

$$\begin{aligned} \text{var}(\|\mathbf{W}\|) &= \sum_{i=1}^n \text{var}(\|\mathbf{W}_i\|) \leq \sum_{i=1}^n \mathbb{E}[\|\mathbf{W}_i\|^2] \\ &\leq \left\| \sum_{j=1}^n \mathbf{u}_j\mathbf{u}_j^T/r \right\| \leq 1/r \end{aligned}$$

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

$$\Pr(\|\mathbf{W}\| \geq \epsilon) \leq d\exp\left(\frac{-\epsilon^2/2}{2/r + \epsilon/(3r)}\right) \leq 0.01$$

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

Then the expected number of samples to preserve ℓ_2 subspace embedding is $O(\sum_{i=1}^n \tilde{l}_i(\log d)/\epsilon^2)$. Now from lemma 5.3 we know that for $p = 2, \sum_{i=1}^n \tilde{l}_i$ is $O(d(1 + \log\|\mathbf{A}\|) - \min_i\|\mathbf{a}_i\|)$. Finally to get $\Pr(\|\mathbf{W}\| \geq \epsilon) \leq 0.01$ the algorithm samples $O(\frac{d\log d}{\epsilon^2}(1 + \log\|\mathbf{A}\| - d^{-1}\min_i \log\|\mathbf{a}_i\|))$ rows. □

A.6. Latent Variable Modeling

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 $\widetilde{\mathcal{T}_3}$ to realize the latent variables (Anandkumar et al., 2014). The tensor is reduced to an orthogonally decomposable tensor by

multiplying a matrix called whitening matrix $\mathbf{W} \in \mathbb{R}^{d\times k}$, such that $\mathbf{W}^T\mathbf{M}_2\mathbf{W} = \mathbf{I}_k$. Here k is the number of latent variables we are interested and $\mathbf{M}_2 \in \mathbb{R}^{d\times d}$ is the 2^{nd} order moment. Now the reduced tensor $\widetilde{\mathcal{T}}_r = \widetilde{\mathcal{T}}_3(\mathbf{W}, \mathbf{W}, \mathbf{W})$ is a $k \times k \times k$ sized orthogonally decomposable tensor. Next by running robust tensor power iteration (RTPI) on $\widetilde{\mathcal{T}}_r$ 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 $\widetilde{\mathcal{T}}_r \in \mathbb{R}^{k\times k\times k}$. Now rephrasing the main theorem 5.1 of (Anandkumar et al., 2014) we get that the $\|\mathcal{M}_3 - \widetilde{\mathcal{T}}_3\| \leq \varepsilon\|\mathbf{W}\|^{-3}$ where $\mathcal{M}_3$ is the true 3-order tensor with no noise and $\widetilde{\mathcal{T}}_3$ 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 $\mathbf{A} \in \mathbb{R}^{n\times d}$ with rows coming in streaming fashion and the algorithm* `LineFilter+KernelFilter` *returns a coreset $\mathbf{C}$ which guarantees (1) such that if for all unit vector $\mathbf{x} \in \mathbf{Q}$, it ensures $\epsilon \sum_{i\leq n} |\mathbf{a}^T\mathbf{x}|^3 \leq \varepsilon\|\mathbf{W}\|^{-3}$. Then applying the RTPI on the sampled coreset $\mathbf{C}$ returns k eigenpairs $\{\lambda_i, \mathbf{v}_i\}$ of the reduced (orthogonally decomposable) tensor, such that it ensures $\forall i \in [k]$,*

$$\|\mathbf{v}_{\pi(i)} - \mathbf{v}_i\| \leq 8\varepsilon/\lambda_i \quad \textit{and} \quad |\lambda_{\pi(i)} - \lambda_i| \leq 5\varepsilon$$

Here precisely we have $\mathbf{Q}$ as the column space of the $\mathbf{W}^\dagger$, where $\mathbf{W}$ 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 generate a dataset with 200K rows in $\mathbb{R}^{30}$. Each coordinate of the row is 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-order tensors $\hat{\mathcal{T}}$ and we also create the tensor $\mathcal{T}$ using the entire dataset. 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 $\forall \mathbf{x} \in \mathbf{Q}$.

`LineFilter`(2): Here we sample rows based on online leverage scores $c_i = \mathbf{a}_i^T(\mathbf{A}_i^T\mathbf{A}_i)^{-1}\mathbf{a}_i$. We define a sampling probability for an incoming row i as $p_i = c_i/(\sum_{j=1}^{i} c_j)$. 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, i.e., $|\mathcal{T}(\mathbf{x},\mathbf{x},\mathbf{x},\mathbf{x}) - \hat{\mathcal{T}}(\mathbf{x},\mathbf{x},\mathbf{x},\mathbf{x})|/\mathcal{T}(\mathbf{x},\mathbf{x},\mathbf{x},\mathbf{x})$, between all three sampling schemes mentioned above. Here we have $\mathcal{T}(\mathbf{x},\mathbf{x},\mathbf{x},\mathbf{x}) = \sum_{i=1}^{n}(\mathbf{a}_i^T\mathbf{x})^4$ and $\hat{\mathcal{T}}(\mathbf{x},\mathbf{x},\mathbf{x},\mathbf{x}) = \sum_{\mathbf{c}_i\in\mathbf{C}}(\mathbf{c}_i^T\mathbf{x})^4$. In table (3), $\mathbf{Q}$ is set of right singular vectors of $\mathbf{A}$ corresponding to the 5 smallest singular values. This table reports the relative error approximation $|\sum_{\mathbf{x}\in[\mathbf{Q}]}\mathcal{T}(\mathbf{x},\mathbf{x},\mathbf{x},\mathbf{x}) - \sum_{\mathbf{x}\in[\mathbf{Q}]}\hat{\mathcal{T}}(\mathbf{x},\mathbf{x},\mathbf{x},\mathbf{x})|/\sum_{\mathbf{x}\in[\mathbf{Q}]}\mathcal{T}(\mathbf{x},\mathbf{x},\mathbf{x},\mathbf{x})$. The table (4) reports for $\mathbf{x}$ as the right singular vector of the smallest singular value of $\mathbf{A}$. Here we choose this $\mathbf{x}$ 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 $\mathbf{x} \in \mathbf{Q}$

Sample	Uniform	LineFilter(2)	LineFilter +KernelFilter
200	1.1663	0.2286	**0.1576**
250	0.4187	0.1169	**0.0855**
300	0.6098	0.1195	**0.0611**
350	0.5704	0.0470	**0.0436**

Table 4. Error with $\mathbf{x}$ as right singular vector of the smallest singular value

Sample	Uniform	LineFilter(2)	LineFilter +KernelFilter
100	1.3584	0.8842	**0.6879**
200	0.8886	0.5005	**0.3952**
300	0.8742	0.4195	**0.3696**
500	0.9187	0.3574	**0.2000**