ICML

International Conference On Machine Learning

On the Robustness of CountSketch to Adaptive inputs

Edith Cohen

Xin Lyu

Jelani Nelson

Tamás Sarlós

Moshe Shechner

Uri Stemmer

CountSketch

CountSketch [CCF ‘02] (Feature hashing [MD ‘89]):

- Dim reduction method using linear projections:

CountSketch

CountSketch [CCF '02] (Feature hashing [MD '89]):

- Dim reduction method using linear projections:

$$\boldsymbol{v} \in \mathbb{R}^n \rightarrow sketch(\boldsymbol{v}) \in \mathbb{R}^d, \qquad d \ll n$$

CountSketch

CountSketch [CCF ‘02] (Feature hashing [MD ‘89]):

- Dim reduction method using linear projections:

$$\boldsymbol{v} \in \mathbb{R}^n \rightarrow sketch(\boldsymbol{v}) \in \mathbb{R}^d, \qquad d \ll n$$

- Usage – Recovering ***heavy hitters*** of $\boldsymbol{v}$.

CountSketch

CountSketch [CCF '02] (Feature hashing [MD '89]):

- Dim reduction method using linear projections:

$$\boldsymbol{v} \in \mathbb{R}^n \rightarrow sketch(\boldsymbol{v}) \in \mathbb{R}^d, \qquad d \ll n$$

- Usage – Recovering ***heavy hitters*** of $\boldsymbol{v}$.

Heavy hitters of $\boldsymbol{v}$ are "preserved" in $sketch(\boldsymbol{v})$

CountSketch

CountSketch [CCF '02] (Feature hashing [MD '89]):

- Dim reduction method using linear projections:

$$\boldsymbol{v} \in \mathbb{R}^n \rightarrow sketch(\boldsymbol{v}) \in \mathbb{R}^d, \qquad d \ll n$$

- Applications:

CountSketch

CountSketch [CCF '02] (Feature hashing [MD '89]):

- Dim reduction method using linear projections:

$$\boldsymbol{v} \in \mathbb{R}^n \ \rightarrow sketch(\boldsymbol{v}) \in \mathbb{R}^d, \qquad d \ll n$$

- Applications:
  - Streaming (memory)

CountSketch

CountSketch [CCF '02] (Feature hashing [MD '89]):

- Dim reduction method using linear projections:

$$\boldsymbol{v} \in \mathbb{R}^n \rightarrow sketch(\boldsymbol{v}) \in \mathbb{R}^d, \qquad d \ll n$$

- Applications:
  - Streaming (memory)
  - Distributed Aggregation (communication)

CountSketch

CountSketch [CCF ‘02] (Feature hashing [MD ‘89]):

- Dim reduction method using linear projections:

$$\boldsymbol{v} \in \mathbb{R}^n \rightarrow sketch(\boldsymbol{v}) \in \mathbb{R}^d, \qquad d \ll n$$

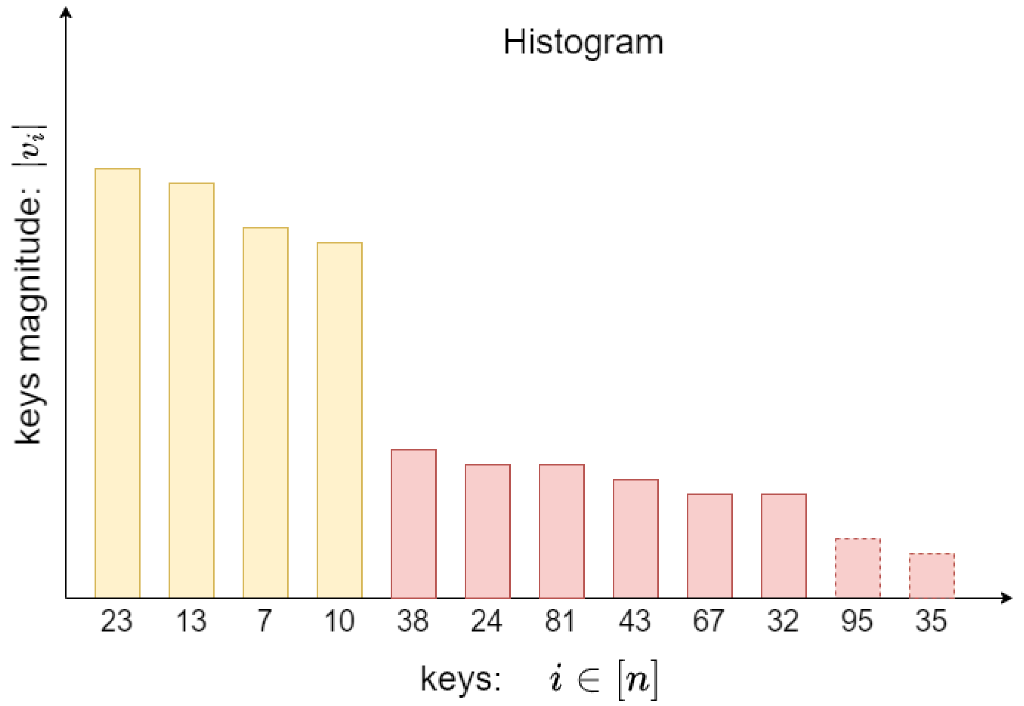
- Applications:
  - Streaming (memory)
  - Distributed Aggregation (communication)
  - Compression (parameters)

HeavyHitters problem

Definition l_2-Heavy Hitters.
For $\boldsymbol{v} \in \mathbb{R}^n$, and parameter k, the l_2-heavy Hitters of $\boldsymbol{v}$ are *keys* $i \in [n]$ s.t.

$$v_i^2 \geq \frac{1}{k} \|v_{tail}\|_2^2$$

Where v_{tail} is obtained from v by replacing the k largest entries with 0.

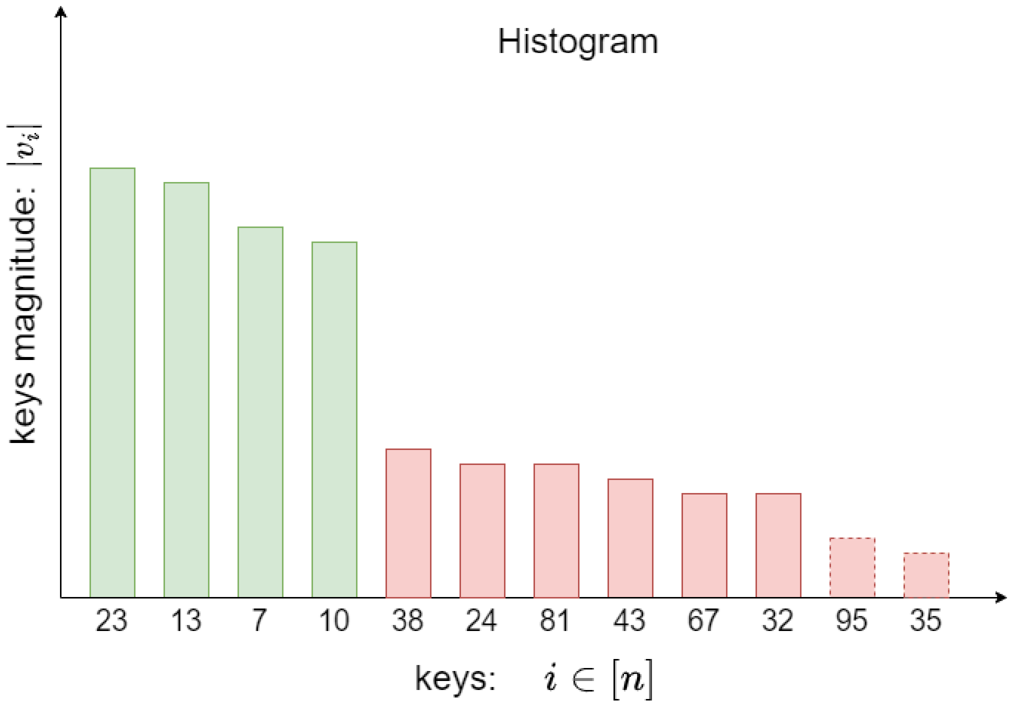


HeavyHitters problem

Definition l_2-Heavy Hitters.
For $\boldsymbol{v} \in \mathbb{R}^n$, and parameter k, the l_2-heavy Hitters of $\boldsymbol{v}$ are *keys* $i \in [n]$ s.t.

$$v_i^2 \geq \frac{1}{k} \|v_{tail}\|_2^2$$

Where v_{tail} is obtained from v by replacing the k largest entries with 0.

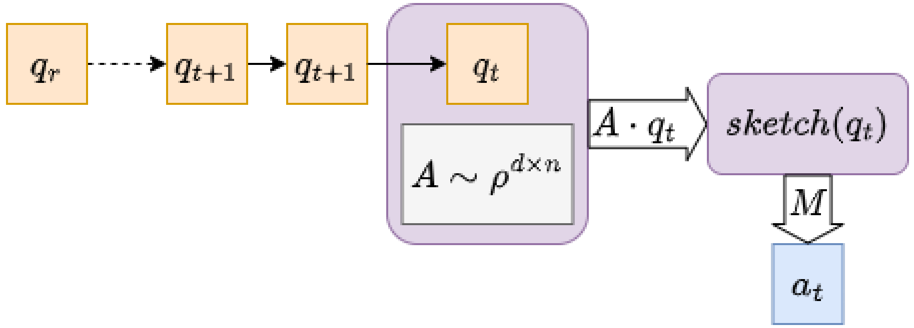


Heavy Hitters Problem (l_2):

Goal: Given $\boldsymbol{v} \in \mathbb{R}^n$, return a set of *keys* $H \subset [n]$ of size $O(k)$ that includes **all** l_2-heavy hitters of $\boldsymbol{v}$.

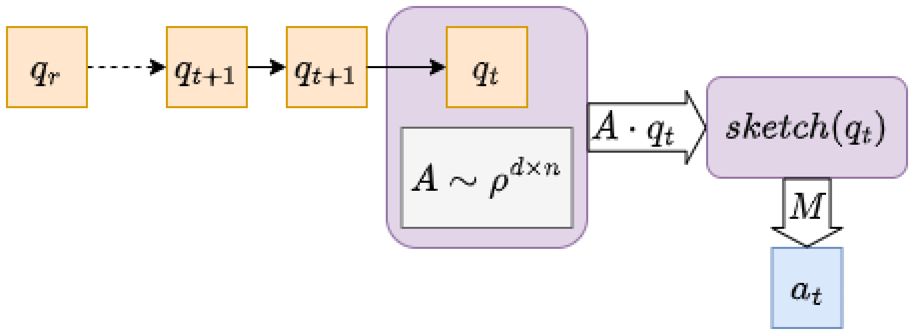
CountSketch Performance

Oblivious setting



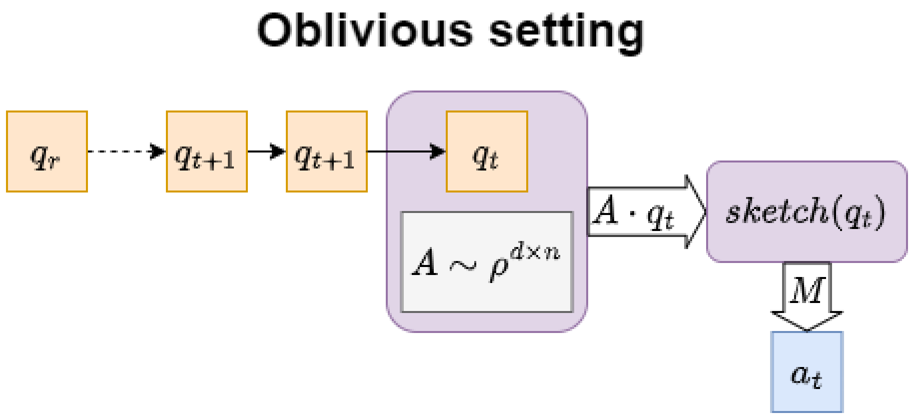
CountSketch Performance

Oblivious setting



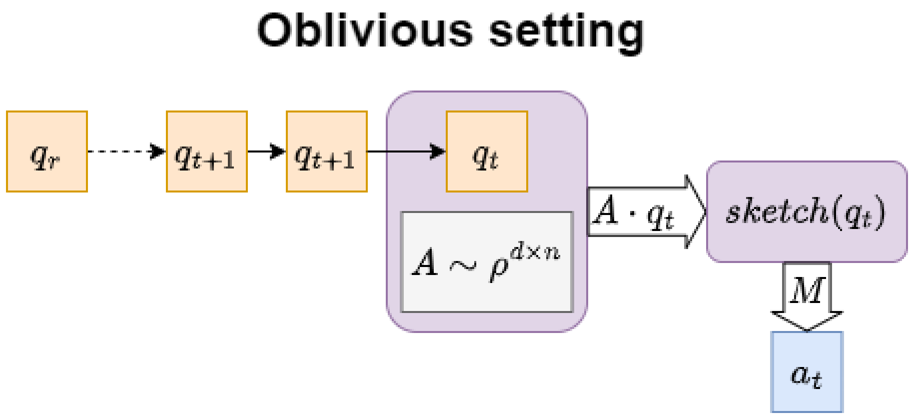
- Sketch matrix $\boldsymbol{A}$ is drawn from distribution $\boldsymbol{\rho}$.

CountSketch Performance



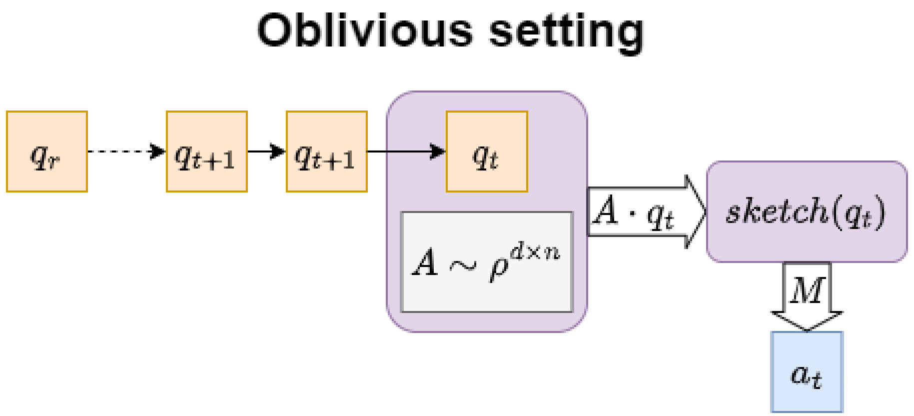
- Sketch matrix $\boldsymbol{A}$ is drawn from distribution $\boldsymbol{\rho}$.
- For r rounds:
  output $a_t \leftarrow M(sketch(q_t))$ for query q_t.

CountSketch Performance



- Sketch matrix $\boldsymbol{A}$ is drawn from distribution $\boldsymbol{\rho}$.
- For r rounds:
  output $a_t \leftarrow M(sketch(q_t))$ for query q_t.
- **Assumption:** $\{q_t\}_{t \in [r]}$ are **fixed in advance**

CountSketch Performance



- Sketch matrix $\boldsymbol{A}$ is drawn from distribution $\boldsymbol{\rho}$.
- For r rounds:
  output $a_t \leftarrow M(sketch(q_t))$ for query q_t.
- **Assumption:** $\{q_t\}_{t\in[r]}$ are **fixed in advance**

Performance:

For $r = 2^{\Omega(\ell)}$, a_t are correct (W.H.P).

Where $\ell \times k$ is the size of *sketch*.

Motivating questions

For l_2-HeavyHitters:

- There are no deterministic sketches [KPW '21].

Motivating questions

For l_2-HeavyHitters:

- There are no deterministic sketches [KPW '21].
- The only known sketching algorithms are CountSketch (and variants)

Motivating questions

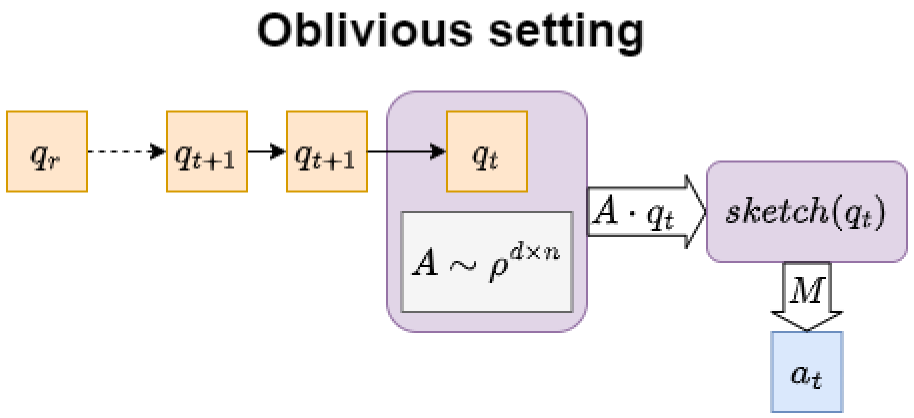
For l_2-HeavyHitters:

- There are no deterministic sketches [KPW '21].
- The only known sketching algorithms are CountSketch (and variants)

What can be said on their robustness to adaptive inputs?

(when input **depend** on previous outputs and randomness)

CountSketch Performance



- Sketch matrix $\boldsymbol{A}$ is drawn from distribution $\boldsymbol{\rho}$.
- For r rounds:
  output $a_t \leftarrow M(sketch(q_t))$ for query q_t.
- **Assumption:** $\{q_t\}_{t \in [r]}$ are **fixed in advance**

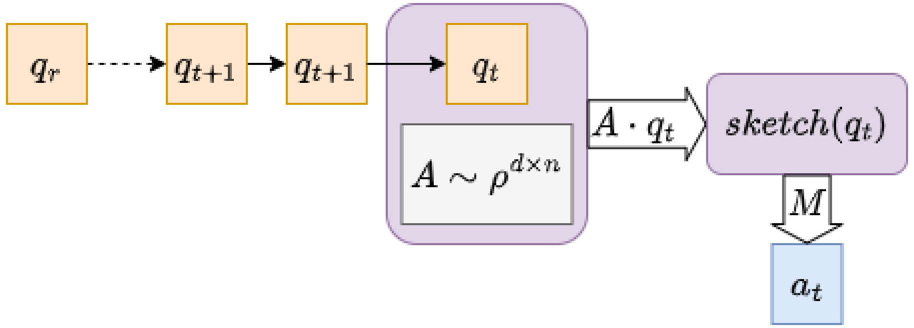
Performance:

For $r = 2^{\Omega(\ell)}$, a_t are correct (W.H.P).

Where $\ell \times k$ is the size of *sketch*.

CountSketch Performance

Oblivious setting



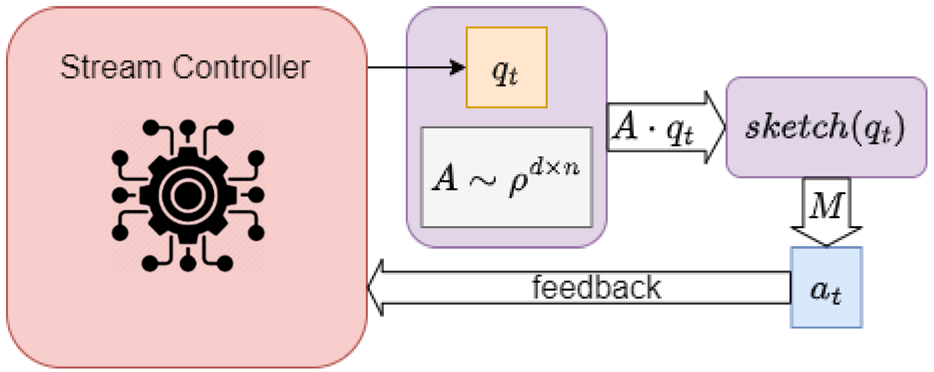
- Sketch matrix $\boldsymbol{A}$ is drawn from distribution $\boldsymbol{\rho}$.
- For r rounds:
  output $a_t \leftarrow M(sketch(q_t))$ for query q_t.
- **Assumption:** $\{q_t\}_{t\in[r]}$ are **fixed in advance**

Performance:

For $r = 2^{\Omega(\ell)}$, a_t are correct (W.H.P).

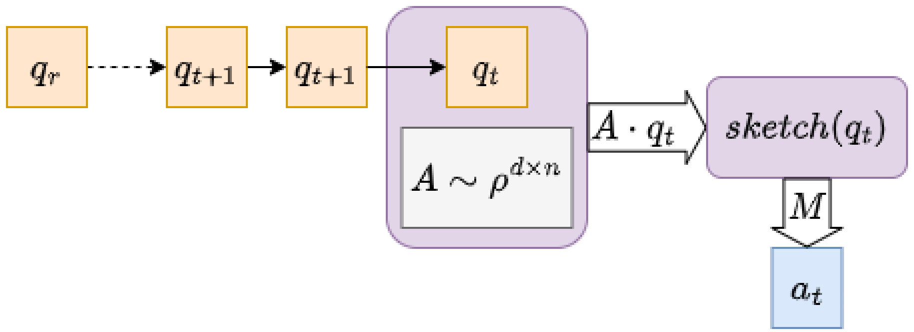
Where $\ell \times k$ is the size of *sketch*.

Adaptive setting



CountSketch Performance

Oblivious setting



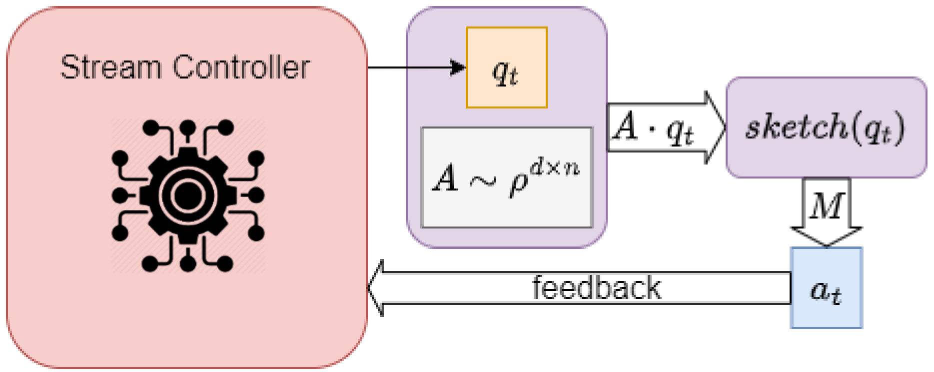
- Sketch matrix $\boldsymbol{A}$ is drawn from distribution $\boldsymbol{\rho}$.
- For r rounds:
  output $a_t \leftarrow M(sketch(q_t))$ for query q_t.
- **Assumption:** $\{q_t\}_{t\in[r]}$ are **fixed in advance**

Performance:

For $r = 2^{\Omega(\ell)}$, a_t are correct (W.H.P).

Where $\ell \times k$ is the size of *sketch*.

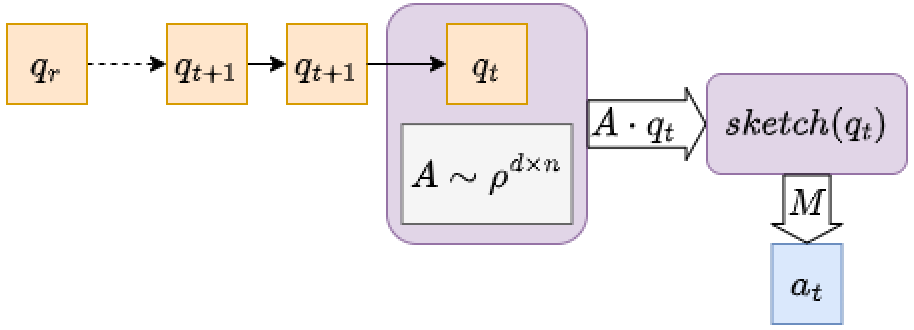
Adaptive setting



- Sketch matrix $\boldsymbol{A}$ is drawn from distribution $\boldsymbol{\rho}$.
- For r rounds:
  output $a_t \leftarrow M(sketch(q_t))$ for query q_t.

CountSketch Performance

Oblivious setting



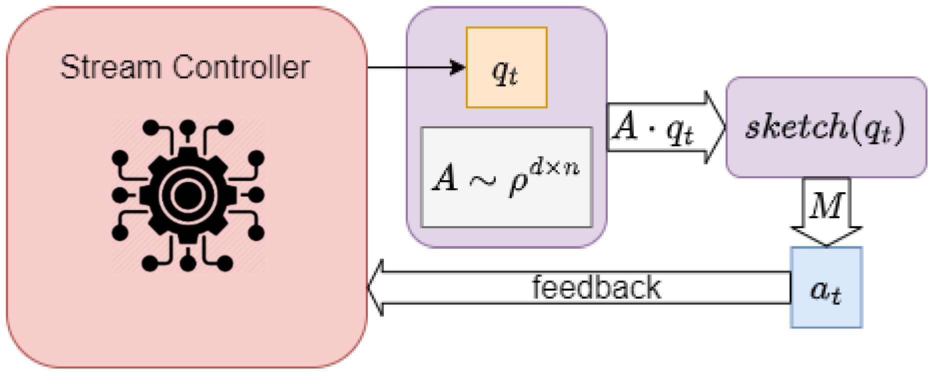
- Sketch matrix $\boldsymbol{A}$ is drawn from distribution $\boldsymbol{\rho}$.
- For r rounds:
  output $a_t \leftarrow M(sketch(q_t))$ for query q_t.
- **Assumption:** $\{q_t\}_{t \in [r]}$ are **fixed in advance**

Performance:

For $r = 2^{\Omega(\ell)}$, a_t are correct (W.H.P).

Where $\ell \times k$ is the size of *sketch*.

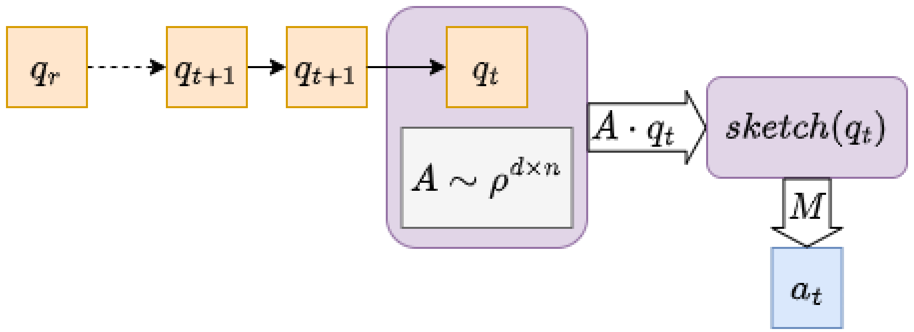
Adaptive setting



- Sketch matrix $\boldsymbol{A}$ is drawn from distribution $\boldsymbol{\rho}$.
- For r rounds:
  output $a_t \leftarrow M(sketch(q_t))$ for query q_t.
- **Assumption:** q_t may **depend** on $\{a_i\}_{i \le t-1}$

CountSketch Performance

Oblivious setting



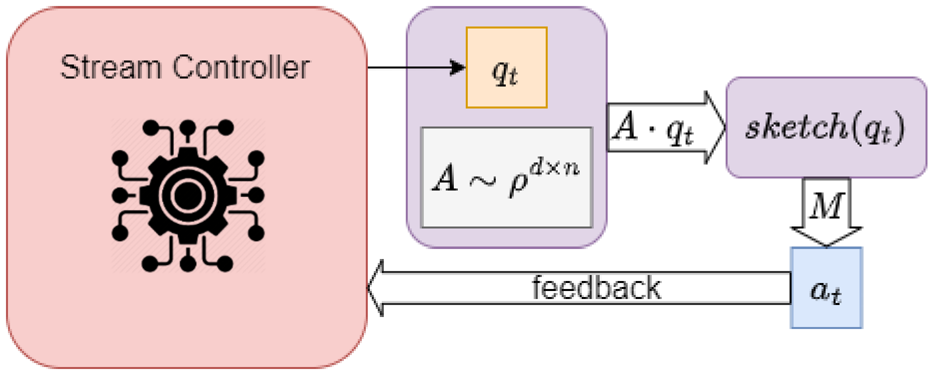
- Sketch matrix $\boldsymbol{A}$ is drawn from distribution $\boldsymbol{\rho}$.
- For r rounds:
  output $a_t \leftarrow M(sketch(q_t))$ for query q_t.
- **Assumption:** $\{q_t\}_{t\in[r]}$ are **fixed in advance**

Performance:

For $r = 2^{\Omega(\ell)}$, a_t are correct (W.H.P).

Where $\ell \times k$ is the size of *sketch*.

Adaptive setting

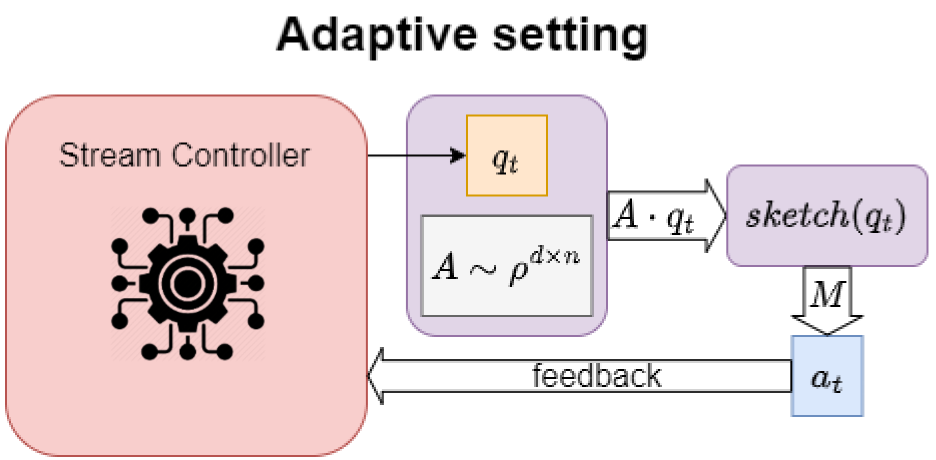


- Sketch matrix $\boldsymbol{A}$ is drawn from distribution $\boldsymbol{\rho}$.
- For r rounds:
  output $a_t \leftarrow M(sketch(q_t))$ for query q_t.
- **Assumption:** q_t may **depend** on $\{a_i\}_{i\leq t-1}$

Performance:

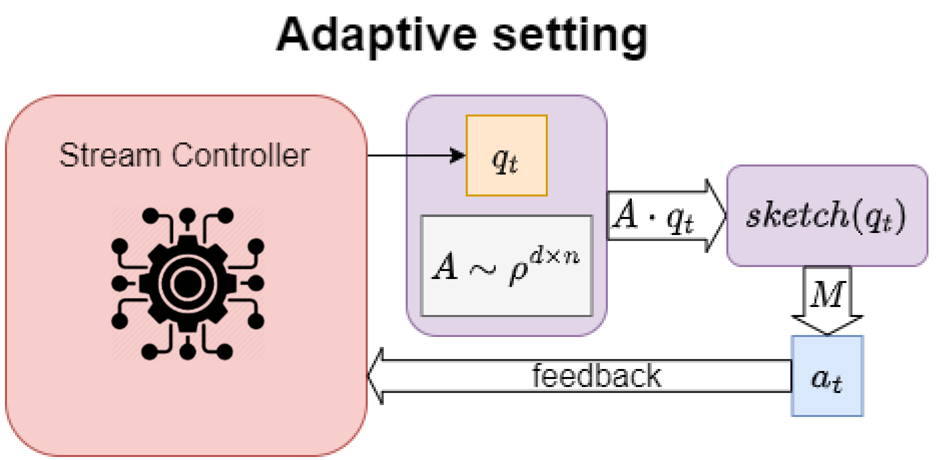
?

The Adaptive Inputs Setting



Why do we care about the adaptive setting?

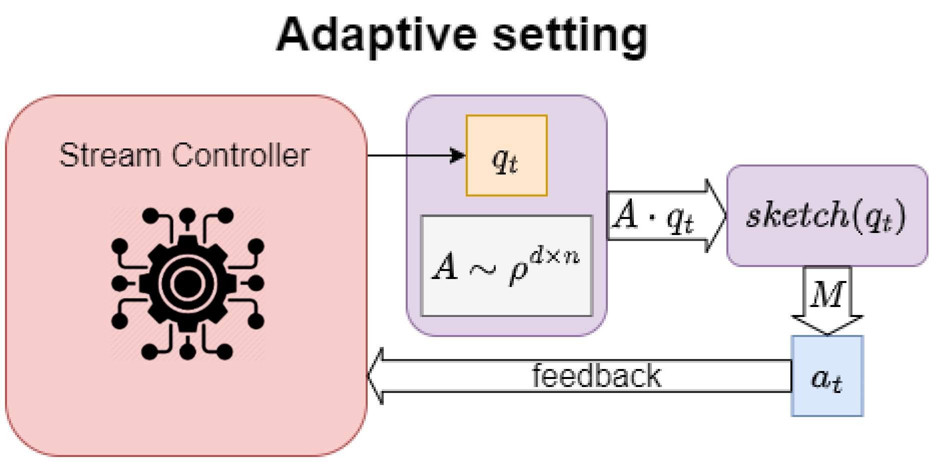
The Adaptive Inputs Setting



Why do we care about the adaptive setting?

- May appear naturally in **systems with feedback** (see e.g. [SKMS '19], [RPUISBGA '20]).

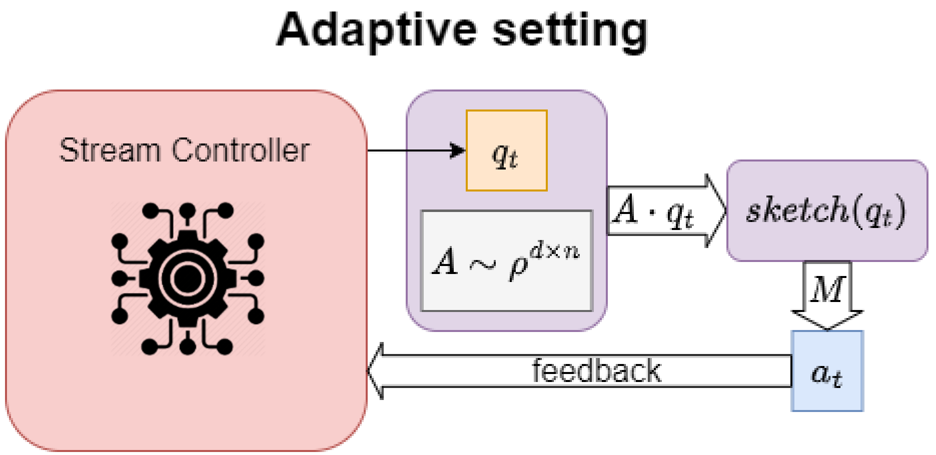
The Adaptive Inputs Setting



Why do we care about the adaptive setting?

- May appear naturally in **systems with feedback** (see e.g. [SKMS '19], [RPUISBGA '20]).
- Adversarial input selection: assuming input controller **tries to Fail the sketching algorithm**.

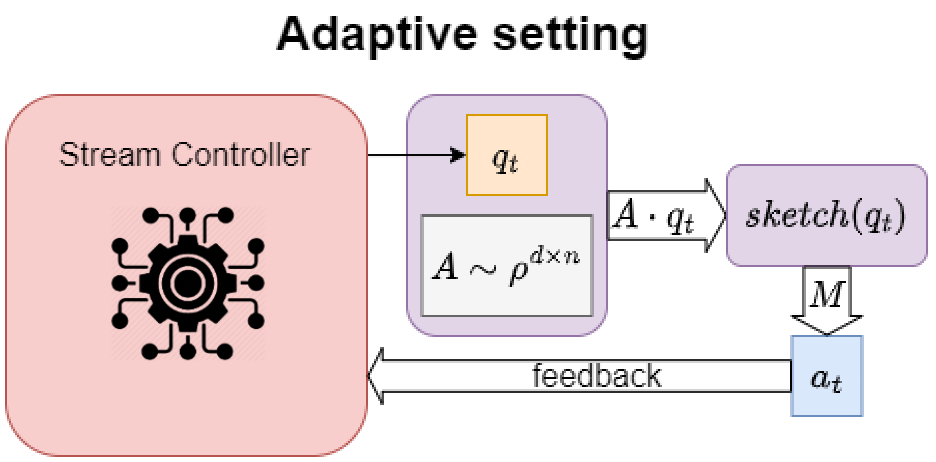
The Adaptive Inputs Setting



Adaptive Setting (Prior Work).

- Recent line of work has used wrapping methods over Oblivious sketching algorithms to achieve robustness (see e.g. [BJWY ‘20], [HKMMS ‘20], [WZ ‘21], [ACSS ‘21], [BEO ‘21]).

The Adaptive Inputs Setting



Adaptive Setting (Prior Work).

- Recent line of work has used wrapping methods over Oblivious sketching algorithms to achieve robustness (see e.g. [BJWY '20], [HKMMS '20], [WZ '21], [ACSS '21], [BEO '21]).

Using wrapping method ([HKMMS '20]):

Can answer $\boldsymbol{r} = \boldsymbol{\Omega}(\ell^2)$ queries correctly (W.H.P).

Where $sketch$ size is $O(\ell \times k^{1.5})$, ℓ is a size-parameter, k is the HeavyHitters parameter.

Main Results

Question1. Is CountSketch already robust to adaptive inputs?

Question2. Can we do better (space-wise) then existing wrapper-robustification results?

Main Results

Question1. Is CountSketch already robust to adaptive inputs?

No. We show an attack on CountSketch with median estimator.

Question2. Can we do better (space-wise) then existing wrapper-robustification results?

Main Results

Question1. Is CountSketch already robust to adaptive inputs?

No. We show an attack on CountSketch with median estimator.
Attack variants apply for variants of sketches and estimators.

Question2. Can we do better (space-wise) then existing wrapper-robustification results?

Main Results

Question1. Is CountSketch already robust to adaptive inputs?

No. We show an attack on CountSketch with median estimator.
Attack variants apply for variants of sketches and estimators.

Question2. Can we do better (space-wise) then existing wrapper-robustification results?

Yes:

Main Results

Question1. Is CountSketch already robust to adaptive inputs?

No. We show an attack on CountSketch with median estimator.
Attack variants apply for variants of sketches and estimators.

Question2. Can we do better (space-wise) then existing wrapper-robustification results?

Yes: BCountSketch

Main Results

Question1. Is CountSketch already robust to adaptive inputs?

No. **We show an attack on CountSketch with median estimator.**
Attack variants apply for variants of sketches and estimators.

Question2. Can we do better (space-wise) then existing wrapper-robustification results?

Yes: BCountSketch + Novel Estimator

Main Results

Question1. Is CountSketch already robust to adaptive inputs?

No. We show an attack on CountSketch with median estimator.
Attack variants apply for variants of sketches and estimators.

Question2. Can we do better (space-wise) then existing wrapper-robustification results?

Yes: BCountSketch + Novel Estimator + DP Technique

Main Results

Question1. Is CountSketch already robust to adaptive inputs?

No. We show an attack on CountSketch with median estimator.
Attack variants apply for variants of sketches and estimators.

Question2. Can we do better (space-wise) then existing wrapper-robustification results?

Yes: BCountSketch + Novel Estimator + DP Technique

For $\Omega(\ell^2)$ queries, this construction has a space complexity of $O(\ell \times k)$. Improvement by a factor of $\sqrt{k}$ upon previous results.

Bibliography: Citation mentioned in the talk

[CCF ‘02] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In Proceedings of the 29th International Colloquium on Automata, Languages and Programming, ICALP ’02, page 693–703. Springer-Verlag, 2002.

[MD ‘89] John E. Moody and Christian J. Darken. Fast learning in networks of locally-tuned processing units. Neural Comput., 1(2):281–294, 1989.

[RPUISBGA ‘20] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. FetchSGD: Communication-efficient federated learning with sketching. In Hal Daumé III and Aarti Singh, editors, Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 8253–8265. PMLR, 13–18 Jul 2020.

[SKMS ‘19] Ryan Spring, Anastasios Kyrillidis, Vijai Mohan, and Anshumali Shrivastava. Compressing gradient optimizers via count-sketches. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 5946–5955. PMLR, 09–15 Jun 2019.

[OJWY2020] Omri Ben-Eliezer, Rajesh Jayaram, David P.Woodruff, and Eylon Yogev. A framework for adversarially robust streaming algorithms. SIGMOD Rec., 50(1):6–13, 2021.

[HKMMS ‘20] Avinatan Hassidim, Haim Kaplan, Yishay Mansour, Yossi Matias, and Uri Stemmer. Adversarially robust streaming algorithms via differential privacy. In Annual Conference on Advances in Neural Information Processing Systems (NeurIPS), 2020.

[WZ ‘21] David P. Woodruff and Samson Zhou. Tight bounds for adversarially robust streams and sliding windows via difference estimators. In Proceedings of the 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS), 2021.

[ACSS ‘21] Idan Attias, Edith Cohen, Moshe Shechner, and Uri Stemmer. A framework for adversarial streaming via differential privacy and difference estimators. CoRR, abs/2107.14527, 2021.

[BEO ‘21] Omri Ben-Eliezer, Talya Eden, and Krzysztof Onak. Adversarially robust streaming via dense-sparse tradeoffs. CoRR, abs/2109.03785, 2021.

[KPW ‘21] Akshay Kamath, Eric Price, and David P. Woodruff. A Simple Proof of a New Set Disjointness with Applications to Data Streams. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, DEU, 2021

Thank you

Edith Cohen
Tamás Sarlós
Xin Lyu
Moshe Shechner
Jelani Nelson
Uri Stemmer