Streaming Algorithms for High-Dimensional Robust Statistics

Ankit Pensia

Ilias Diakonikolas Daniel Kane Thanasis Pittas

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

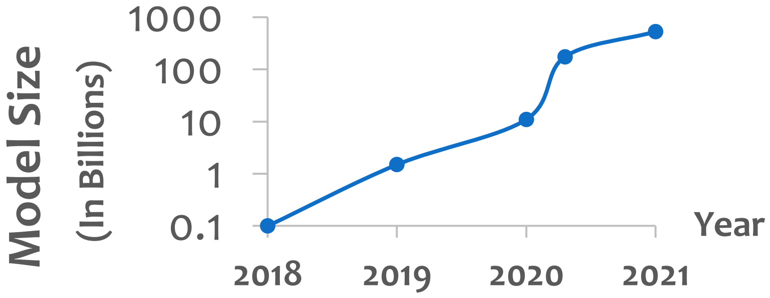
UC San Diego

Challenges in Modern Machine Learning

Challenges in Modern Machine Learning

Huge Models and Datasets

- Both number of samples and dimension

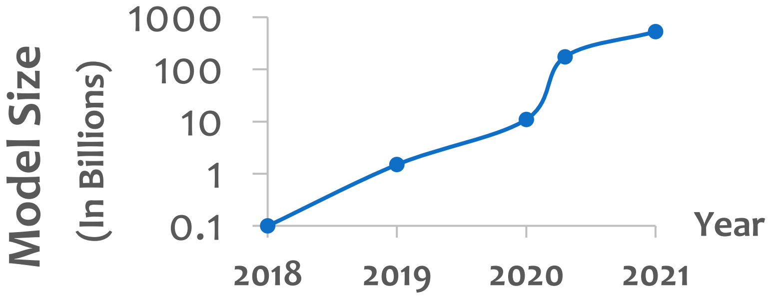


- Can’t store the whole dataset in memory

Challenges in Modern Machine Learning

Huge Models and Datasets

- Both number of samples and dimension



- Can't store the whole dataset in memory

Corrupt Datasets

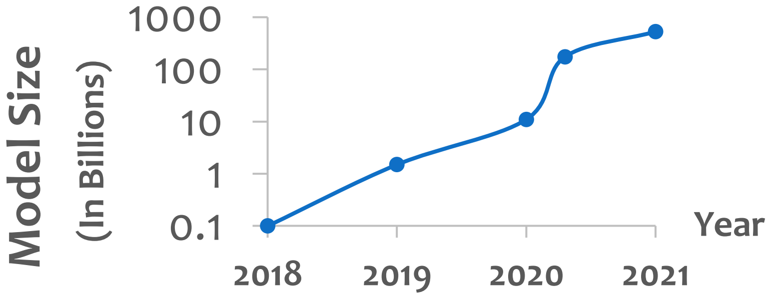
- A constant fraction of data may be corrupt:
  - Measurement errors
  - Adversarial corruption
- Need to use robust algorithms [DKKLMS16,LRV16]
- Current robust algs. store whole data in memory

[DKKLMS16] I.Diakonikolas, G. Kamath, D.M. Kane, J. Li, A. Moitra, A. Stewart. Robust Estimators in High Dimensions without the computational intractability. 2016.
[LRV16] K.A. Lai, A.B. Rao, S. Vempala. Agnostic Estimation of Mean and Covariance. 2016.

Challenges in Modern Machine Learning

Huge Models and Datasets

- Both number of samples and dimension



- Can't store the whole dataset in memory

Corrupt Datasets

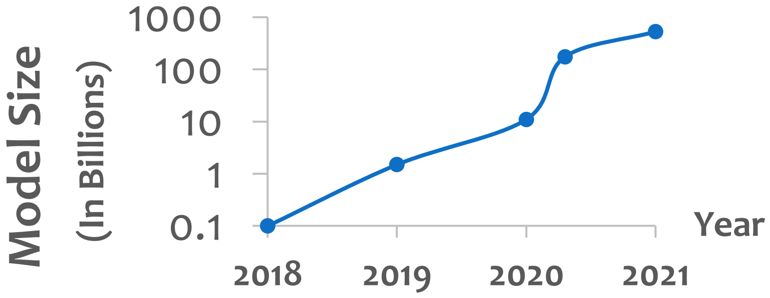
- A constant fraction of data may be corrupt:
  - Measurement errors
  - Adversarial corruption
- Need to use robust algorithms [DKKLMS16,LRV16]
- Current robust algs. store whole data in memory

[DKKLMS16] I.Diakonikolas, G. Kamath, D.M. Kane, J. Li, A. Moitra, A. Stewart. Robust Estimators in High Dimensions without the computational intractability. 2016.
[LRV16] K.A. Lai, A.B. Rao, S. Vempala. Agnostic Estimation of Mean and Covariance. 2016.

Challenges in Modern Machine Learning

Huge Models and Datasets

- Both number of samples and dimension



- Can't store the whole dataset in memory

Corrupt Datasets

- A constant fraction of data may be corrupt:
  - Measurement errors
  - Adversarial corruption
- Need to use robust algorithms [DKKLMS16,LRV16]
- Current robust algs. store whole data in memory

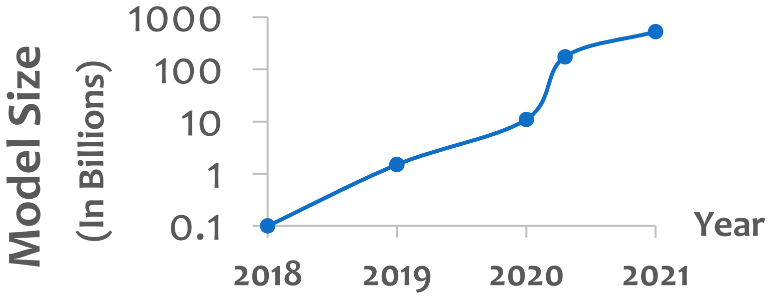
How do we handle this challenge?

[DKKLMS16] I.Diakonikolas, G. Kamath, D.M. Kane, J. Li, A. Moitra, A. Stewart. Robust Estimators in High Dimensions without the computational intractability. 2016.
[LRV16] K.A. Lai, A.B. Rao, S. Vempala. Agnostic Estimation of Mean and Covariance. 2016.

Challenges in Modern Machine Learning

Huge Models and Datasets

- Both number of samples and dimension



- Can't store the whole dataset in memory

Corrupt Datasets

- A constant fraction of data may be corrupt:
  - Measurement errors
  - Adversarial corruption
- Need to use robust algorithms [DKKLMS16,LRV16]
- Current robust algs. store whole data in memory

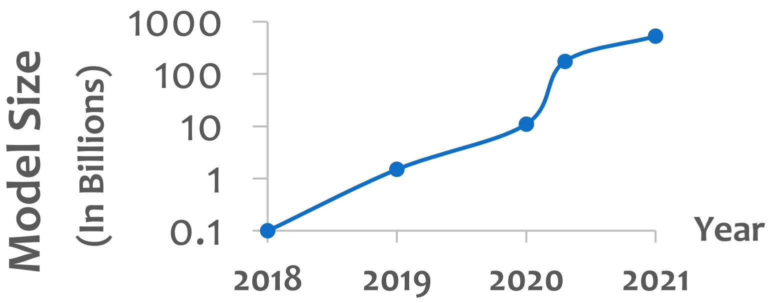
How do we handle this challenge?

[DKKLMS16] I.Diakonikolas, G. Kamath, D.M. Kane, J. Li, A. Moitra, A. Stewart. Robust Estimators in High Dimensions without the computational intractability. 2016.
[LRV16] K.A. Lai, A.B. Rao, S. Vempala. Agnostic Estimation of Mean and Covariance. 2016.

Challenges in Modern Machine Learning

Huge Models and Datasets

- Both number of samples and dimension



- Can’t store the whole dataset in memory

Corrupt Datasets

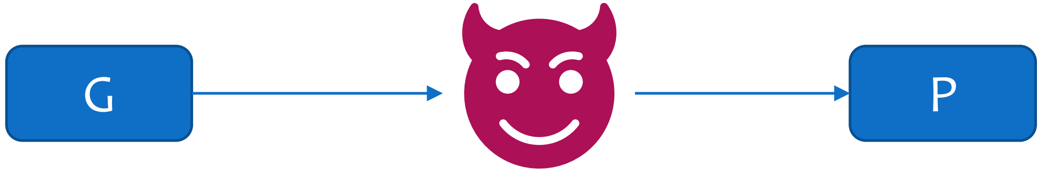
- A constant fraction of data may be corrupt:
  - Measurement errors
  - Adversarial corruption
- Need to use robust algorithms [DKKLMS16,LRV16]
- Current robust algs. store whole data in memory

How do we handle this challenge? Streaming

[DKKLMS16] I.Diakonikolas, G. Kamath, D.M. Kane, J. Li, A. Moitra, A. Stewart. Robust Estimators in High Dimensions without the computational intractability. 2016.
[LRV16] K.A. Lai, A.B. Rao, S. Vempala. Agnostic Estimation of Mean and Covariance. 2016.

Problem Setup: Contamination & Streaming

Data Contamination Model



True Distribution

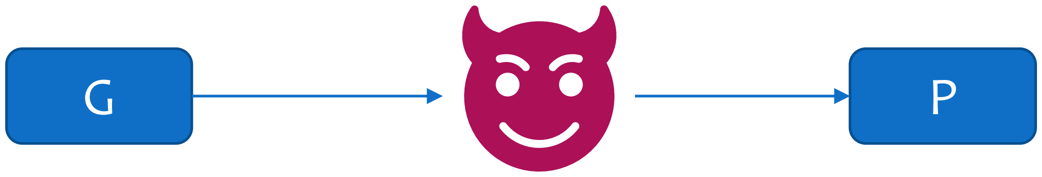
Corrupted Distribution

$\mathrm{d_{TV}}(P, G) \leq \epsilon$

Total-variation contamination

Problem Setup: Contamination & Streaming

Data Contamination Model



True Distribution

Corrupted Distribution

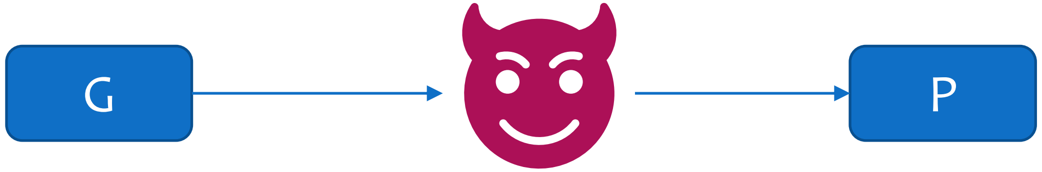
$d_{TV}(P, G) \leq \epsilon$

Total-variation contamination

Streaming Algorithm Model

Problem Setup: Contamination & Streaming

Data Contamination Model



True Distribution

Corrupted Distribution

$\mathrm{d}_{\mathrm{TV}}(P, G) \leq \epsilon$

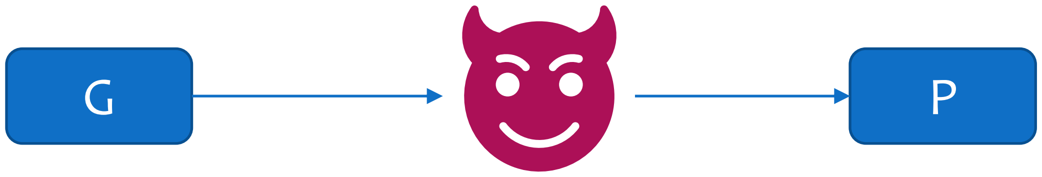
Total-variation contamination

Streaming Algorithm Model

- Initialize memory state S

Problem Setup: Contamination & Streaming

Data Contamination Model



True Distribution

Corrupted Distribution

$\mathrm{d_{TV}}(P, G) \le \epsilon$

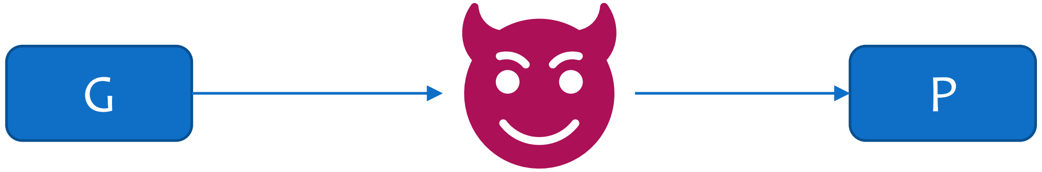
Total-variation contamination

Streaming Algorithm Model

- Initialize memory state S
- For $i = 1, \ldots, n$

Problem Setup: Contamination & Streaming

Data Contamination Model



True Distribution

Corrupted Distribution

$d_{TV}(P, G) \leq \epsilon$

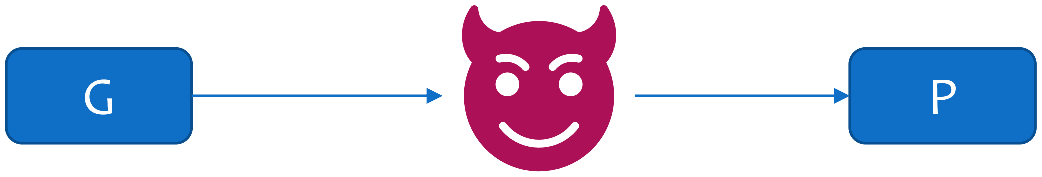
Total-variation contamination

Streaming Algorithm Model

- Initialize memory state S
- For $i = 1, \ldots, n$
  - Observe X_i from P
  - Update memory $S \leftarrow f(S, X_i, i)$

Problem Setup: Contamination & Streaming

Data Contamination Model



True Distribution

Corrupted Distribution

$\mathrm{d}_{\mathrm{TV}}(P, G) \leq \epsilon$

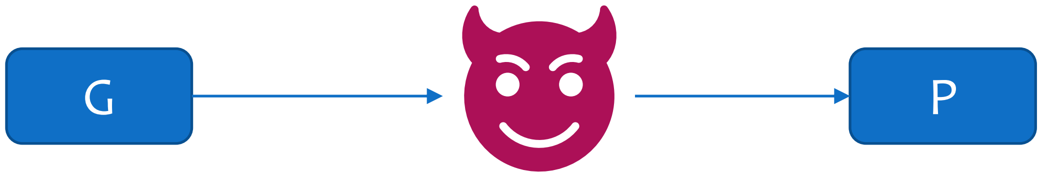
Total-variation contamination

Streaming Algorithm Model

- Initialize memory state S
- For $i = 1, \ldots, n$
  - Observe X_i from P
  - Update memory $S \leftarrow f(S, X_i, i)$
- Output $\hat{\theta}$ as a function of S

Problem Setup: Contamination & Streaming

Data Contamination Model



True Distribution

Corrupted Distribution

$d_{TV}(P, G) \leq \epsilon$

Total-variation contamination

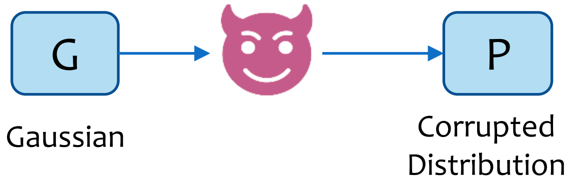
Streaming Algorithm Model

- Initialize memory state S
- For $i = 1, \ldots, n$
  - Observe X_i from P
  - Update memory $S \leftarrow f(S, X_i, i)$
- Output $\hat{\theta}$ as a function of S

Goal: Design an algorithm that is robust, fast, and memory-efficient

Task: Robust High-dimensional Mean Estimation

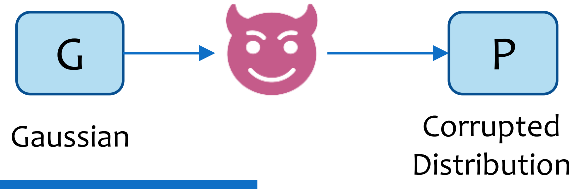
- Let $G = \mathcal{N}(\mu, I)$ be a Gaussian distribution in $\mathbb{R}^d$ with unknown mean



$d_{TV}(P, G) \leq \epsilon$

Task: Robust High-dimensional Mean Estimation

- Let $G = \mathcal{N}(\mu, I)$ be a Gaussian distribution in $\mathbb{R}^d$ with unknown mean

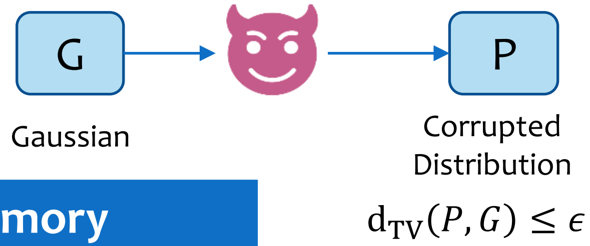


$d_{TV}(P, G) \le \epsilon$

Known Polynomial-time Algorithms	Error Guarantee	Memory

Task: Robust High-dimensional Mean Estimation

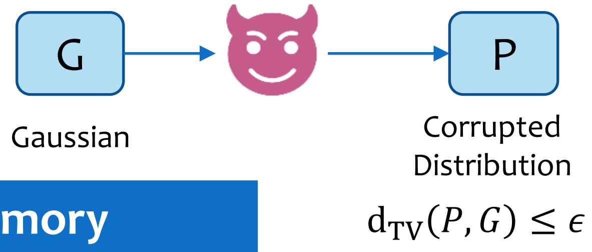
- Let $G = \mathcal{N}(\mu, I)$ be a Gaussian distribution in $\mathbb{R}^d$ with unknown mean



Known Polynomial-time Algorithms	Error Guarantee	Memory
Naïve algorithms (clipping, random subspace, …)	$\epsilon \cdot \mathrm{poly}(d)$	$\tilde{O}(d)$

Task: Robust High-dimensional Mean Estimation

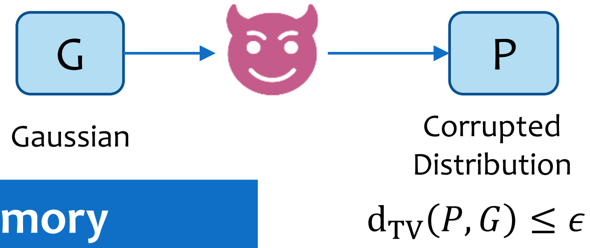
- Let $G = \mathcal{N}(\mu, I)$ be a Gaussian distribution in $\mathbb{R}^d$ with unknown mean



Known Polynomial-time Algorithms	Error Guarantee	Memory
Naïve algorithms (clipping, random subspace, …)	$\epsilon \cdot \mathrm{poly}(d)$	$\tilde{O}(d)$
Existing robust algorithms (filtering, convex programming, gradient descent)	$\tilde{O}(\epsilon)$	$\frac{d^2}{\epsilon^2}$

Task: Robust High-dimensional Mean Estimation

- Let $G = \mathcal{N}(\mu, I)$ be a Gaussian distribution in $\mathbb{R}^d$ with unknown mean



Known Polynomial-time Algorithms	Error Guarantee	Memory
Naïve algorithms (clipping, random subspace, …)	$\epsilon \cdot \mathrm{poly}(d)$	$\tilde{O}(d)$
Existing robust algorithms (filtering, convex programming, gradient descent)	$\tilde{O}(\epsilon)$	$\frac{d^2}{\epsilon^2}$

Is there an efficient algorithm that has error $\tilde{O}(\epsilon)$ and uses memory $\tilde{O}(d)$?

Our Results: Robust Mean Estimation

Efficient Algorithms	Error	Memory
Naïve algs.	$\epsilon \cdot \mathrm{poly}(d)$	d
Existing robust algs.	ϵ	$\frac{d^2}{\epsilon^2}$
This paper	$\boldsymbol{\epsilon}$	$\mathbf{d}$

Theorem[DKPP22] Let P be an ϵ-corruption of $\mathcal{N}(\mu, I)$. Given poly$\left(d, \frac{1}{\epsilon}\right)$ i.i.d. samples from P in the streaming model, there is a nearly-linear time algorithm to compute $\hat{\mu}$ such that w.h.p.

(i) Memory usage = $\tilde{O}(d)$ and (ii) $\|\hat{\mu} - \mu\|_2 = \tilde{O}(\epsilon)$

Our Results: Robust Mean Estimation

Efficient Algorithms	Error	Memory
Naïve algs.	$\epsilon \cdot \text{poly}(d)$	d
Existing robust algs.	ϵ	$\frac{d^2}{\epsilon^2}$
This paper	$\boldsymbol{\epsilon}$	$\mathbf{d}$

Theorem[DKPP22] Let P be an ϵ-corruption of $\mathcal{N}(\mu, I)$. Given poly$\left(d, \frac{1}{\epsilon}\right)$ i.i.d. samples from P in the streaming model, there is a nearly-linear time algorithm to compute $\hat{\mu}$ such that w.h.p.

(i) Memory usage = $\tilde{O}(d)$ and (ii) $\|\hat{\mu} - \mu\|_2 = \tilde{O}(\epsilon)$

- Near-optimal error even with infinite samples and memory

Our Results: Robust Mean Estimation

Efficient Algorithms	Error	Memory
Naïve algs.	$\epsilon \cdot \mathrm{poly}(d)$	d
Existing robust algs.	ϵ	$\frac{d^2}{\epsilon^2}$
This paper	$\boldsymbol{\epsilon}$	$\boldsymbol{d}$

Theorem[DKPP22] Let P be an ϵ-corruption of $\mathcal{N}(\mu, I)$. Given $\mathrm{poly}\left(d, \frac{1}{\epsilon}\right)$ i.i.d. samples from P in the streaming model, there is a nearly-linear time algorithm to compute $\hat{\mu}$ such that w.h.p.

(i) Memory usage = $\tilde{O}(d)$ and (ii) $\|\hat{\mu} - \mu\|_2 = \tilde{O}(\epsilon)$

- Near-optimal error even with infinite samples and memory
- Extends to other well-behaved distributions:
  - Bounded covariance distributions
  - More generally, "stable" distributions

Our Results: Beyond Robust Mean Estimation

Problem	Data Distribution (Before Corruption)	Memory	Error rate

Our Results: Beyond Robust Mean Estimation

Problem	Data Distribution (Before Corruption)	Memory	Error rate
Robust Covariance Estimation	Bdd. 4-th moment	$\tilde{O}(d^2)$	$\left\|\hat{\Sigma} - \Sigma\right\|_F = O(\sqrt{\epsilon})$
	Gaussian Distribution	$\tilde{O}(d^2)$	$\left\|\Sigma^{-0.5}\hat{\Sigma}\Sigma^{-0.5} - I\right\|_F = \tilde{O}(\epsilon)$

Our Results: Beyond Robust Mean Estimation

Problem	Data Distribution (Before Corruption)	Memory	Error rate
Robust Covariance Estimation	Bdd. 4-th moment	$\tilde{O}(d^2)$	$\left\|\hat{\Sigma} - \Sigma\right\|_F = O(\sqrt{\epsilon})$
	Gaussian Distribution	$\tilde{O}(d^2)$	$\left\|\Sigma^{-0.5}\,\hat{\Sigma}\Sigma^{-0.5} - I\right\|_F = \tilde{O}(\epsilon)$
Robust Linear Regression	$Y = X^\top\theta^* + Z$ • $X \sim \mathcal{N}(0, I)$ • $X \perp Z,\ Z \sim \mathcal{N}(0,1)$ • θ^* bdd.	$\tilde{O}(d)$	$\left\|\hat{\theta} - \theta^*\right\|_2 = O(\sqrt{\epsilon})$
Robust Logistic Regression	…	$\tilde{O}(d)$	$\left\|\hat{\theta} - \theta^*\right\|_2 = O(\sqrt{\epsilon})$

Our Results: Beyond Robust Mean Estimation

Problem	Data Distribution (Before Corruption)	Memory	Error rate
Robust Covariance Estimation	Bdd. 4-th moment	$\tilde{O}(d^2)$	$\|\hat{\Sigma} - \Sigma\|_F = O(\sqrt{\epsilon})$
	Gaussian Distribution	$\tilde{O}(d^2)$	$\|\Sigma^{-0.5}\hat{\Sigma}\Sigma^{-0.5} - I\|_F = \tilde{O}(\epsilon)$
Robust Linear Regression	$Y = X^\top\theta^* + Z$ • $X \sim \mathcal{N}(0, I)$ • $X \perp Z,\ Z \sim \mathcal{N}(0,1)$ • θ^* bdd.	$\tilde{O}(d)$	$\|\hat{\theta} - \theta^*\|_2 = O(\sqrt{\epsilon})$
Robust Logistic Regression	…	$\tilde{O}(d)$	$\|\hat{\theta} - \theta^*\|_2 = O(\sqrt{\epsilon})$
Robust Stochastic Convex Optimization	$\min_{\theta \in \mathbb{R}^d} F(\theta)$ • $F(\theta) := \mathbb{E}_Z[f(\theta; Z)]$ • Well-conditioned • $\mathrm{Cov}(\nabla f(\theta; Z))$ bdd.	$\tilde{O}(d)$	$\|\hat{\theta} - \theta^*\|_2 = O(\sqrt{\epsilon})$

Summary

- Developed the first **streaming** algorithms for **high-dimensional robust** statistics

Summary

- Developed the first **streaming** algorithms for **high-dimensional robust** statistics
- **Near-optimal space** complexities for various robust tasks:
  - mean and covariance estimation
  - linear regression and logistic regression
  - stochastic optimization

Summary

- Developed the first **streaming** algorithms for **high-dimensional robust** statistics
- **Near-optimal space** complexities for various robust tasks:
  - mean and covariance estimation
  - linear regression and logistic regression
  - stochastic optimization

Open Questions

Summary

- Developed the first **streaming** algorithms for **high-dimensional robust** statistics
- **Near-optimal space** complexities for various robust tasks:
  - mean and covariance estimation
  - linear regression and logistic regression
  - stochastic optimization

Open Questions

- Your favorite robust statistical tasks in the streaming setting
- Sample-Memory tradeoff
- Stronger (adaptive) adversaries?

Summary

- Developed the first **streaming** algorithms for **high-dimensional robust** statistics
- **Near-optimal space** complexities for various robust tasks:
  - mean and covariance estimation
  - linear regression and logistic regression
  - stochastic optimization

Open Questions

- Your favorite robust statistical tasks in the streaming setting
- Sample-Memory tradeoff
- Stronger (adaptive) adversaries?

Please visit our poster for more details!

Thank You!