Efficient Algorithms for Robust High Dimensional Learning

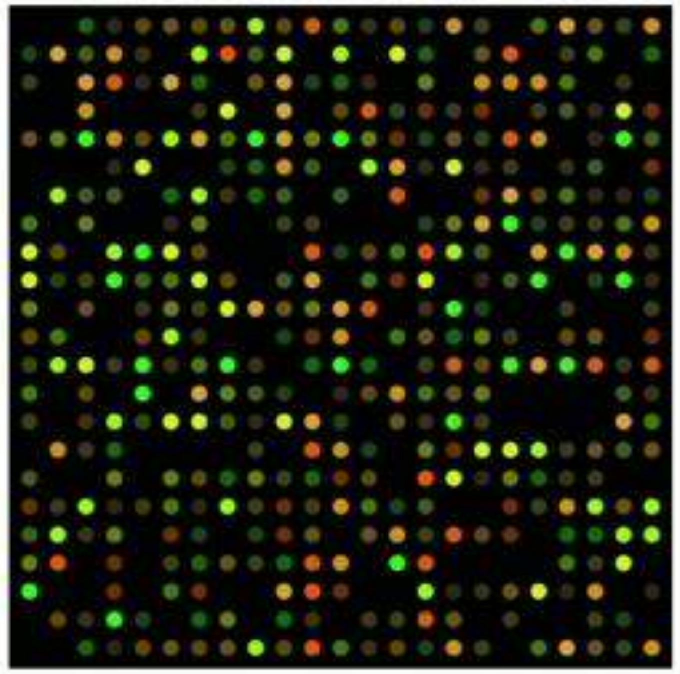
Jerry Li
MSR AI

Robustness: two motivating examples

Genetic data

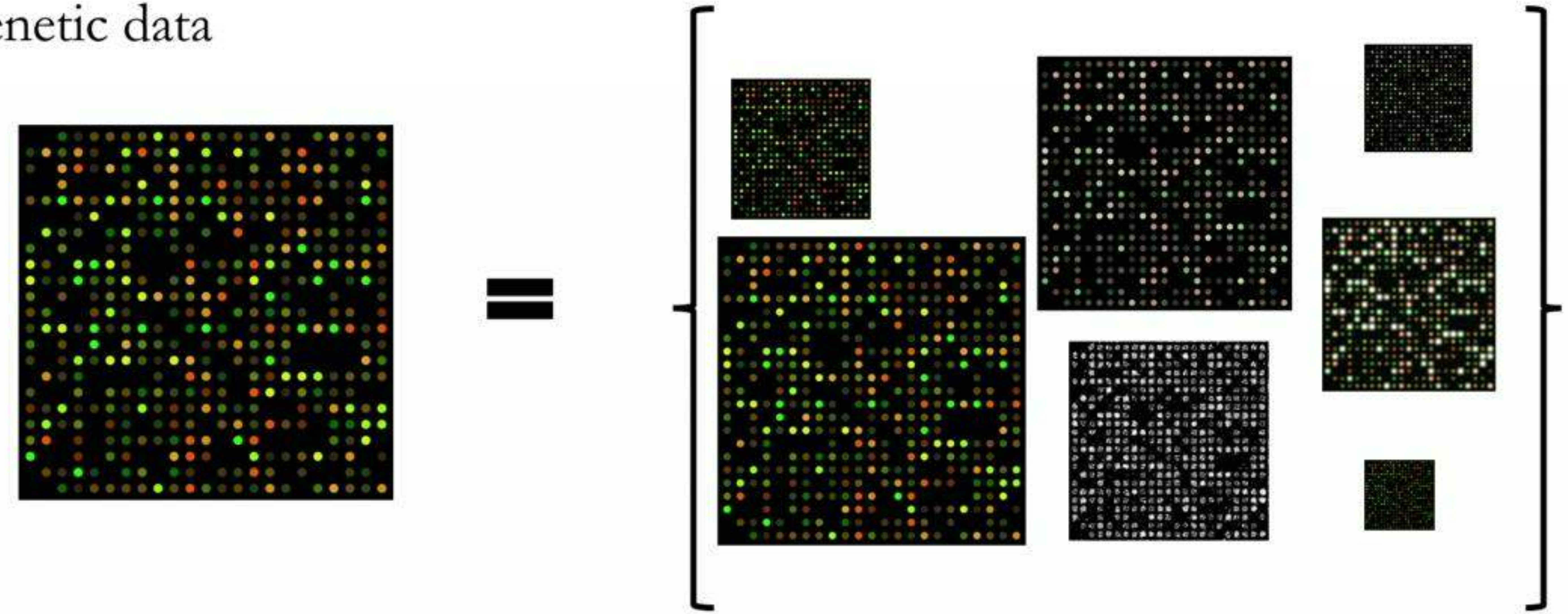
Robustness: two motivating examples

Genetic data



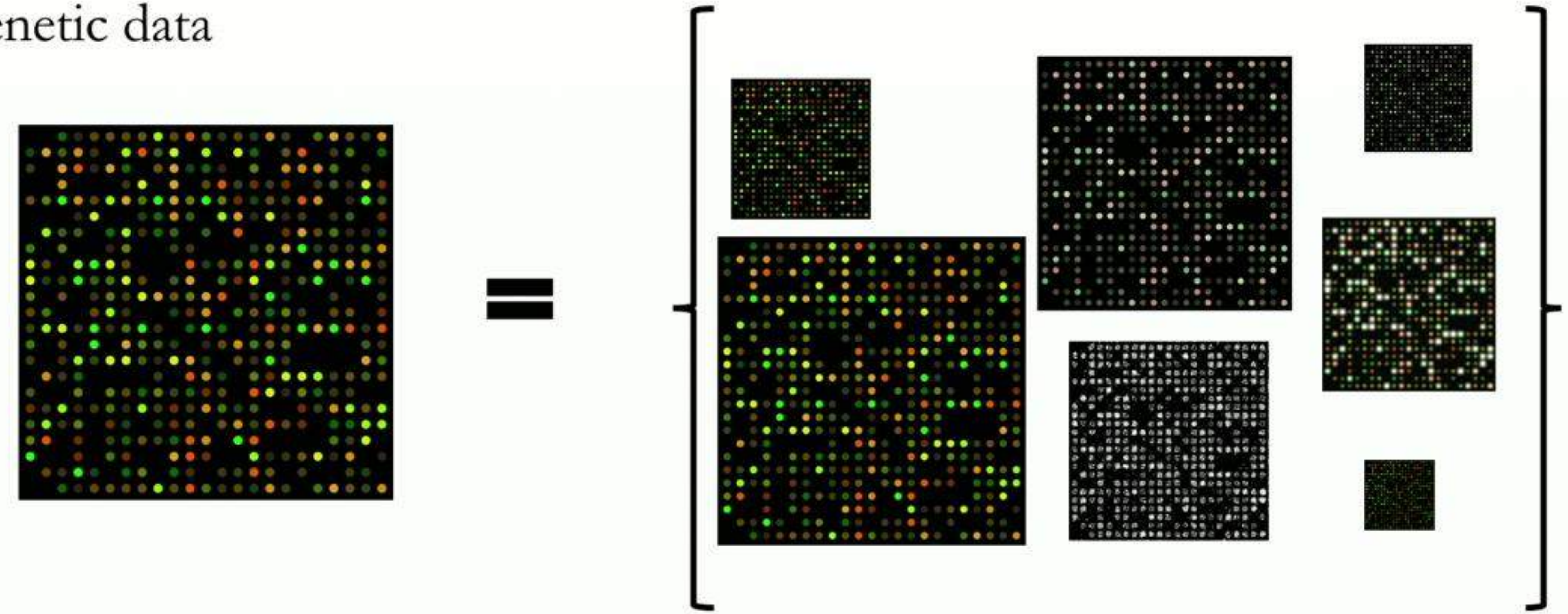
Robustness: two motivating examples

Genetic data



Robustness: two motivating examples

Genetic data



Data is often heterogeneous, causing uncontrolled systematic noise

Robustness: two motivating examples

Data poisoning / Adversarial machine learning

Robustness: two motivating examples

Data poisoning / Adversarial machine learning

Figure from [Gu, Dolan-Gavitt, Garg '17]

Robustness: two motivating examples

Data poisoning / Adversarial machine learning

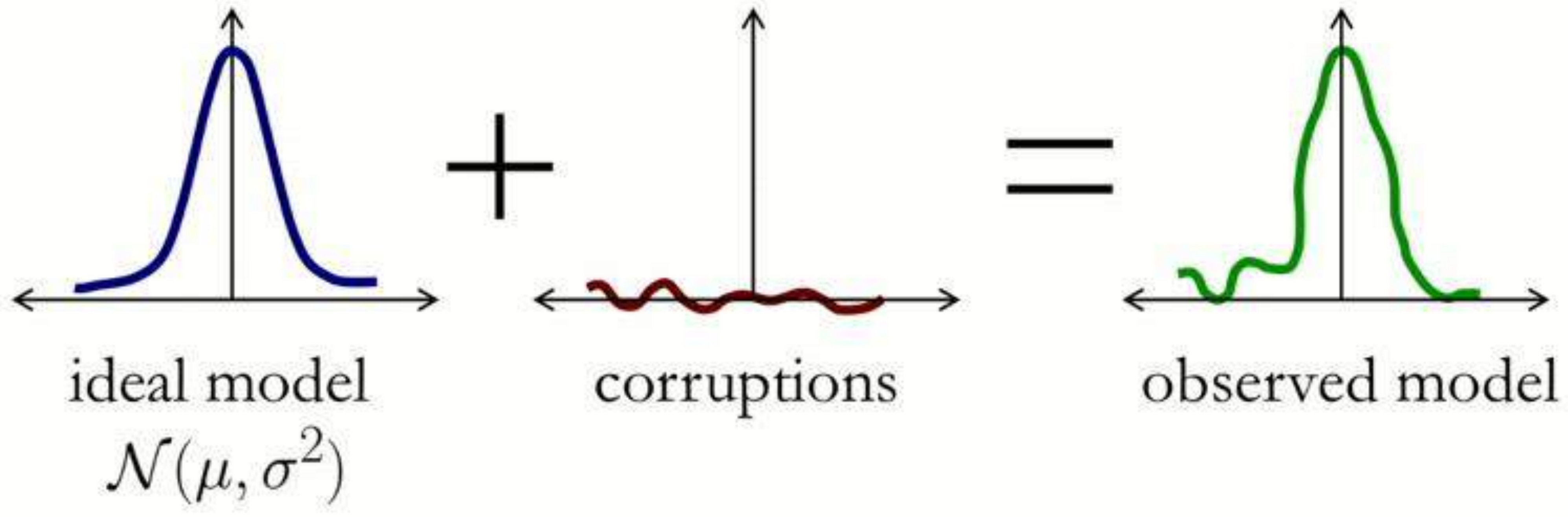
Figure from [Gu, Dolan-Gavitt, Garg '17]

Data can come from untrusted / tampered sources

Robustness: two motivating examples

Large data sets are often inherently noisy

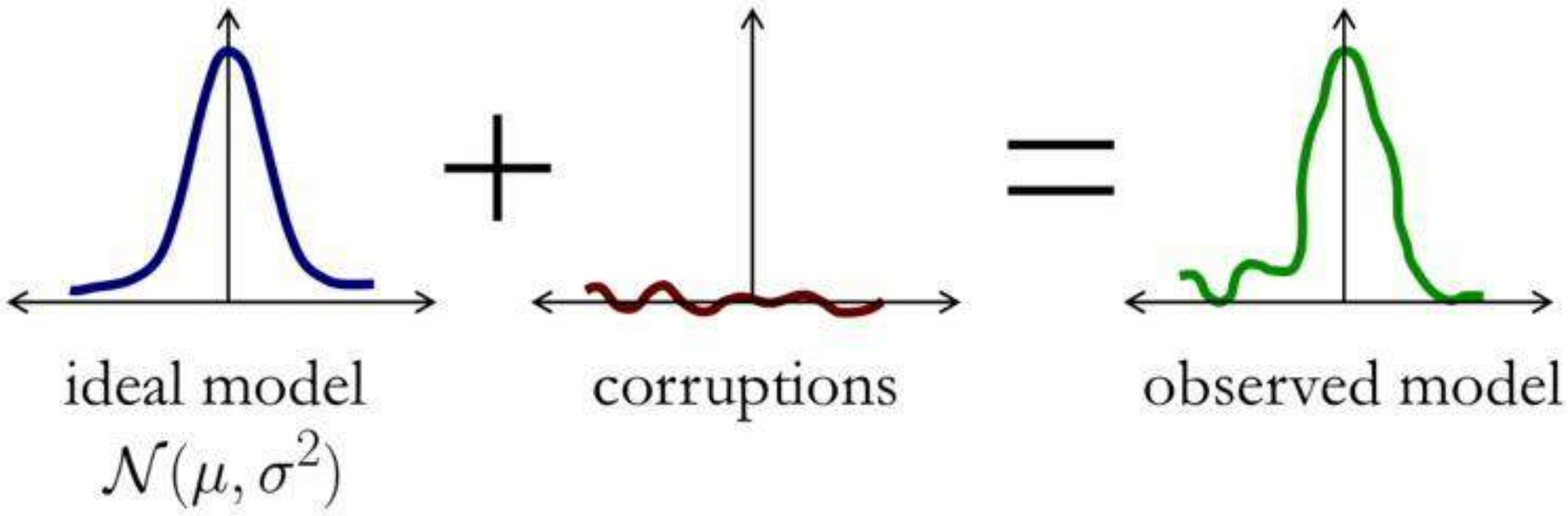
How can we learn from noisy high dimensional data?



Robustness: two motivating examples

Large data sets are often inherently noisy

How can we learn from noisy high dimensional data?



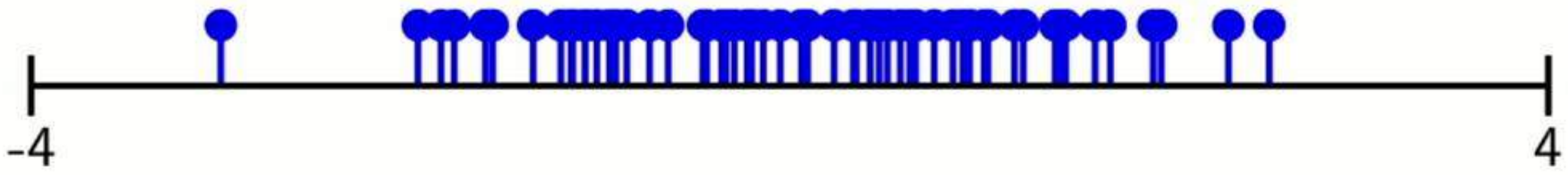
Challenge: Develop algorithms which are provably robust to worst case noise

Robust statistics [Huber], [Tukey] '60s

- Given samples from a distribution, where an adversary has moved an ε-fraction of the points arbitrarily, can you recover statistics of the original distribution?

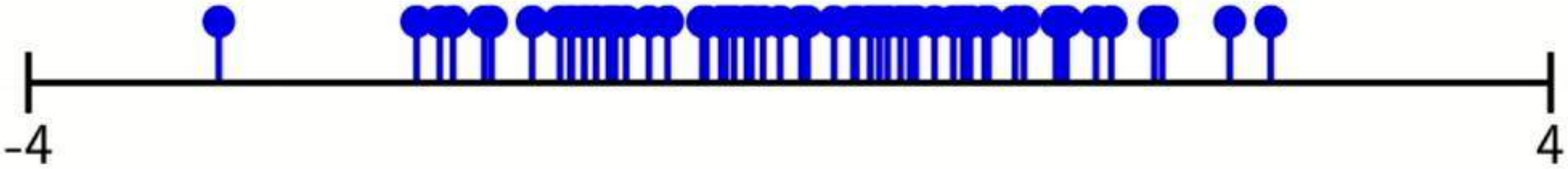
Robust statistics [Huber], [Tukey] '60s

- Given samples from a distribution, where an adversary has moved an ε-fraction of the points arbitrarily, can you recover statistics of the original distribution?



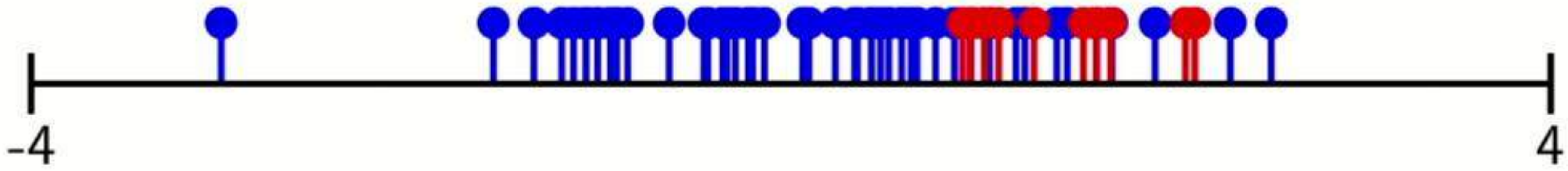
Robust statistics [Huber], [Tukey] ’60s

- Given samples from a distribution, where an adversary has moved an ε-fraction of the points arbitrarily, can you recover statistics of the original distribution?



Robust statistics [Huber], [Tukey] ’60s

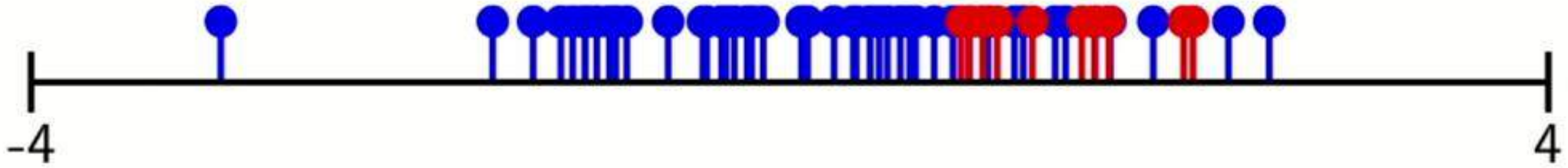
- Given samples from a distribution, where an adversary has moved an ε-fraction of the points arbitrarily, can you recover statistics of the original distribution?



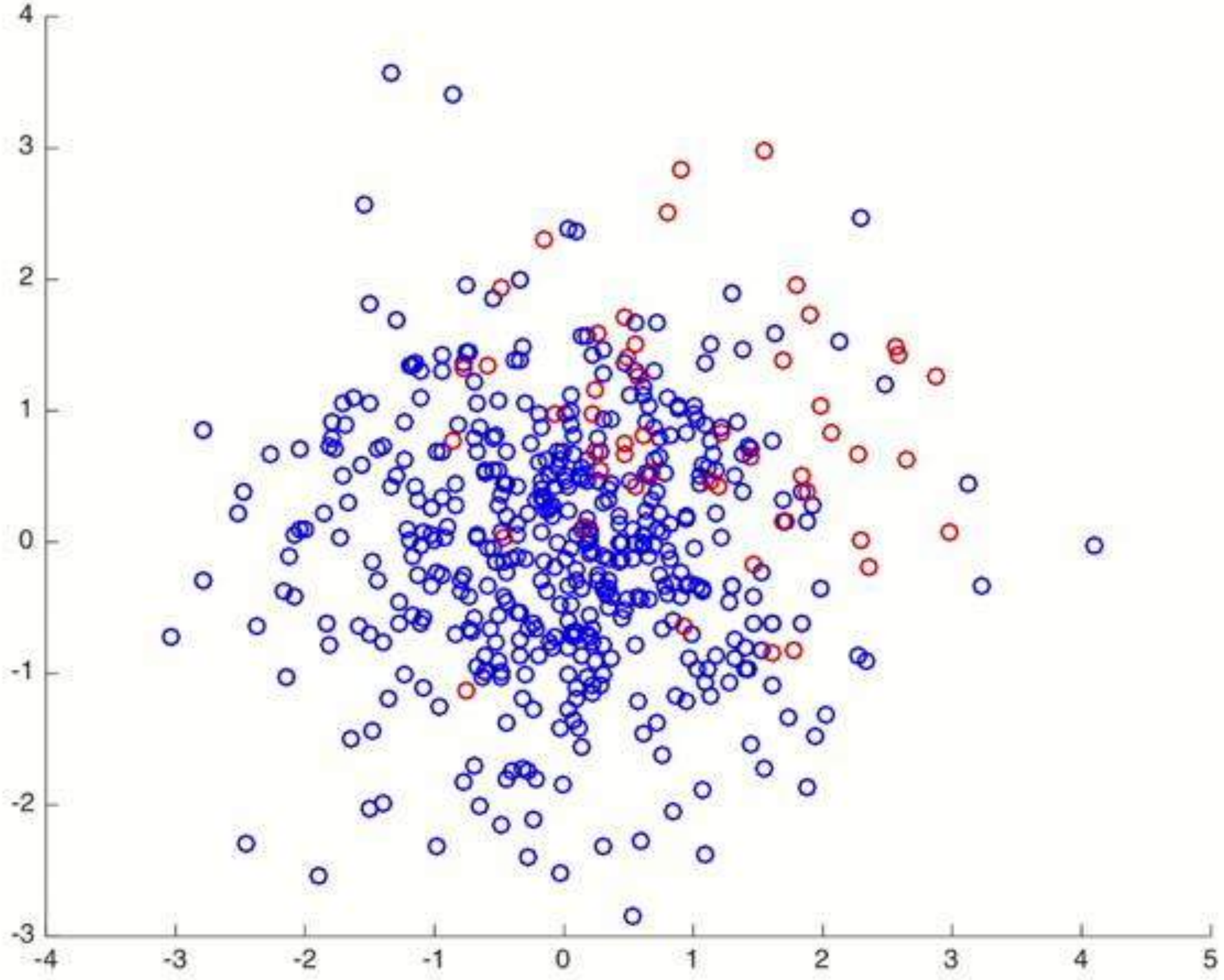
Robust statistics [Huber], [Tukey] '60s

- Given samples from a distribution, where an adversary has moved an ε-fraction of the points arbitrarily, can you recover statistics of the original distribution?

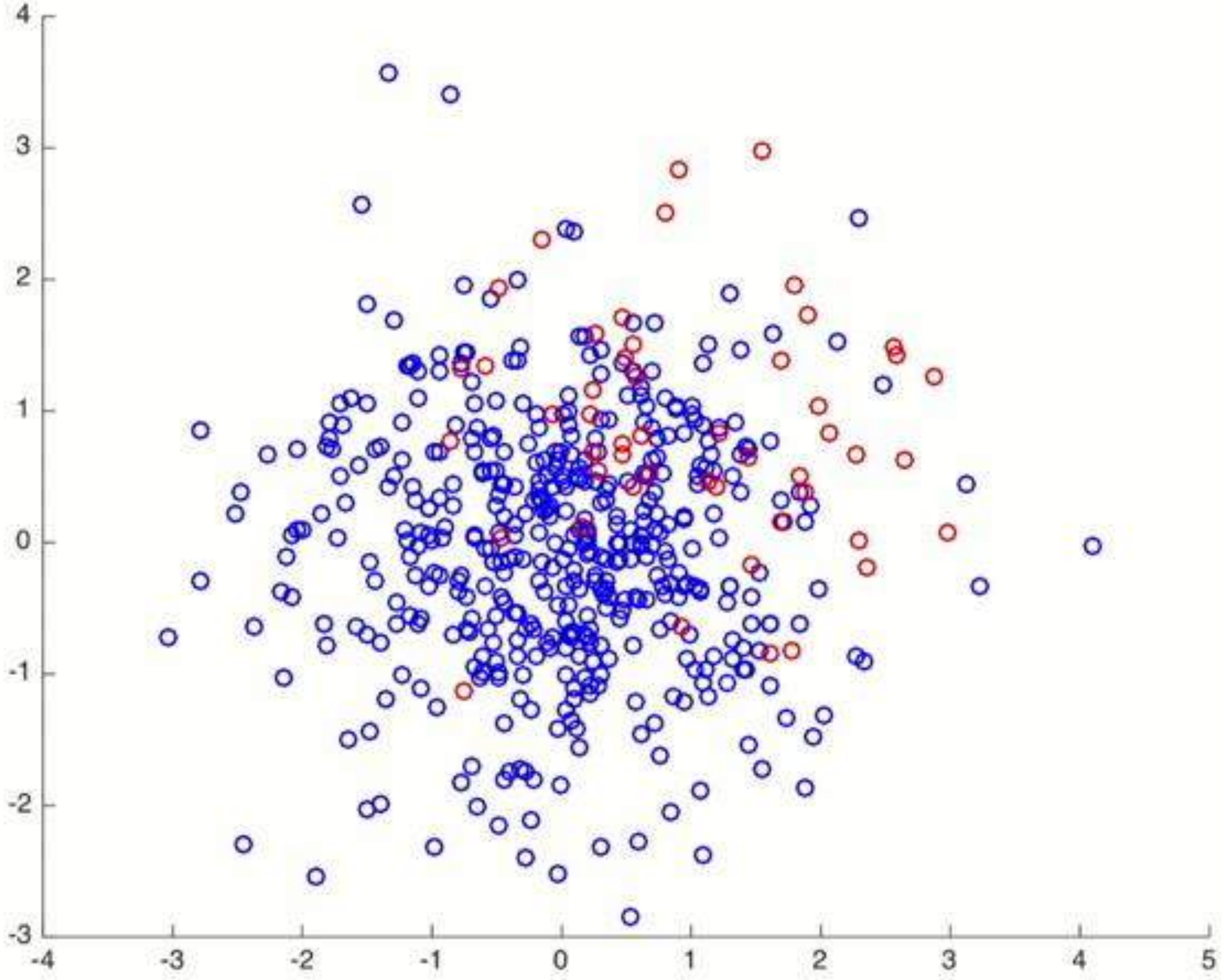
$\boxed{\varepsilon\text{-corrupted}}$



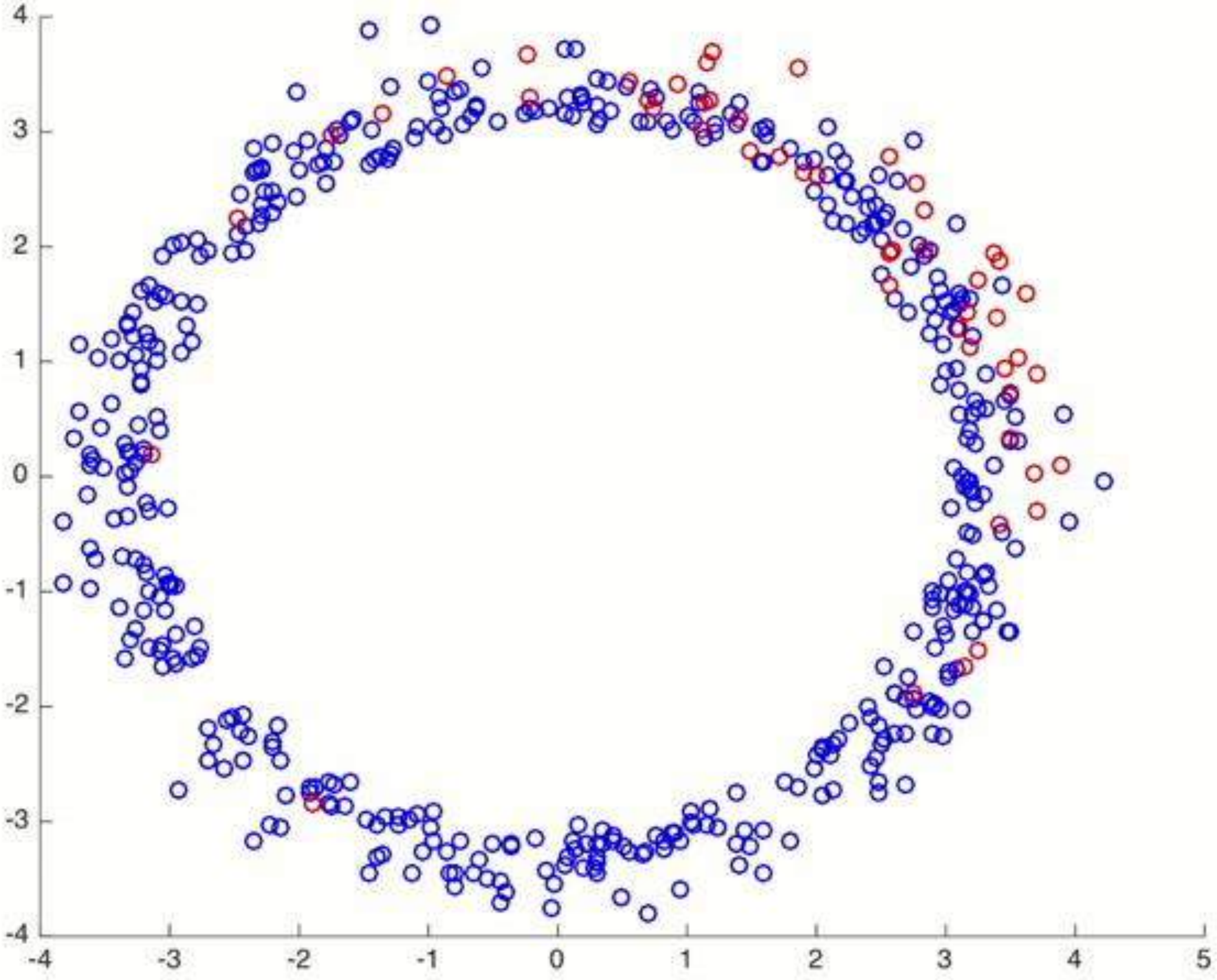
Corruptions in 2 dimensions



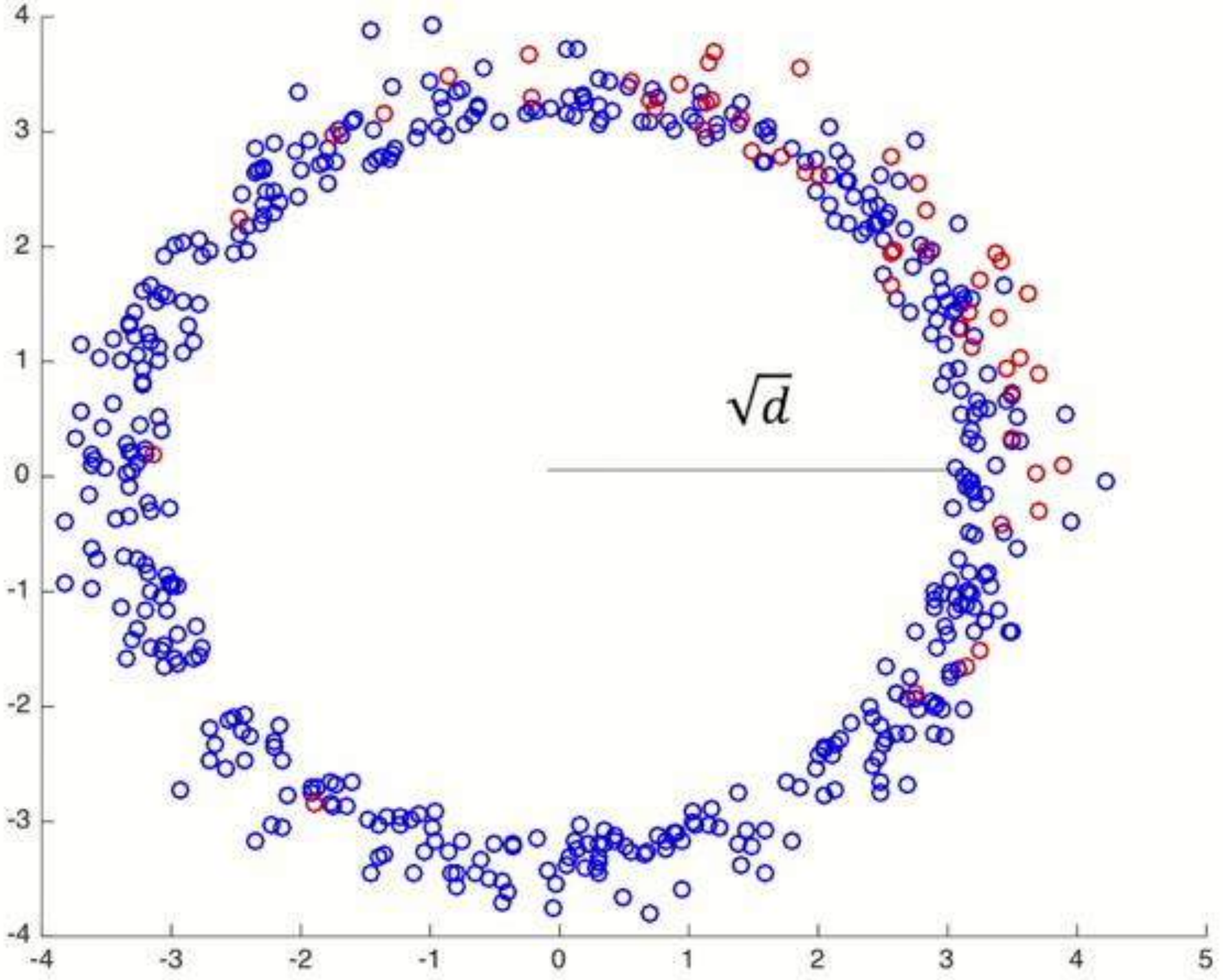
Corruptions in 2 dimensions



Corruptions in high dimensions



Corruptions in high dimensions



Any method looking for outliers will lose dimension factors

Must look for corruptions globally

A curse of dimensionality?

All known approaches for high-dimensional mean estimation either

1. Are computationally intractable in high dimensions; or
2. Lose accuracy factors which depend polynomially on the dimension

A curse of dimensionality?

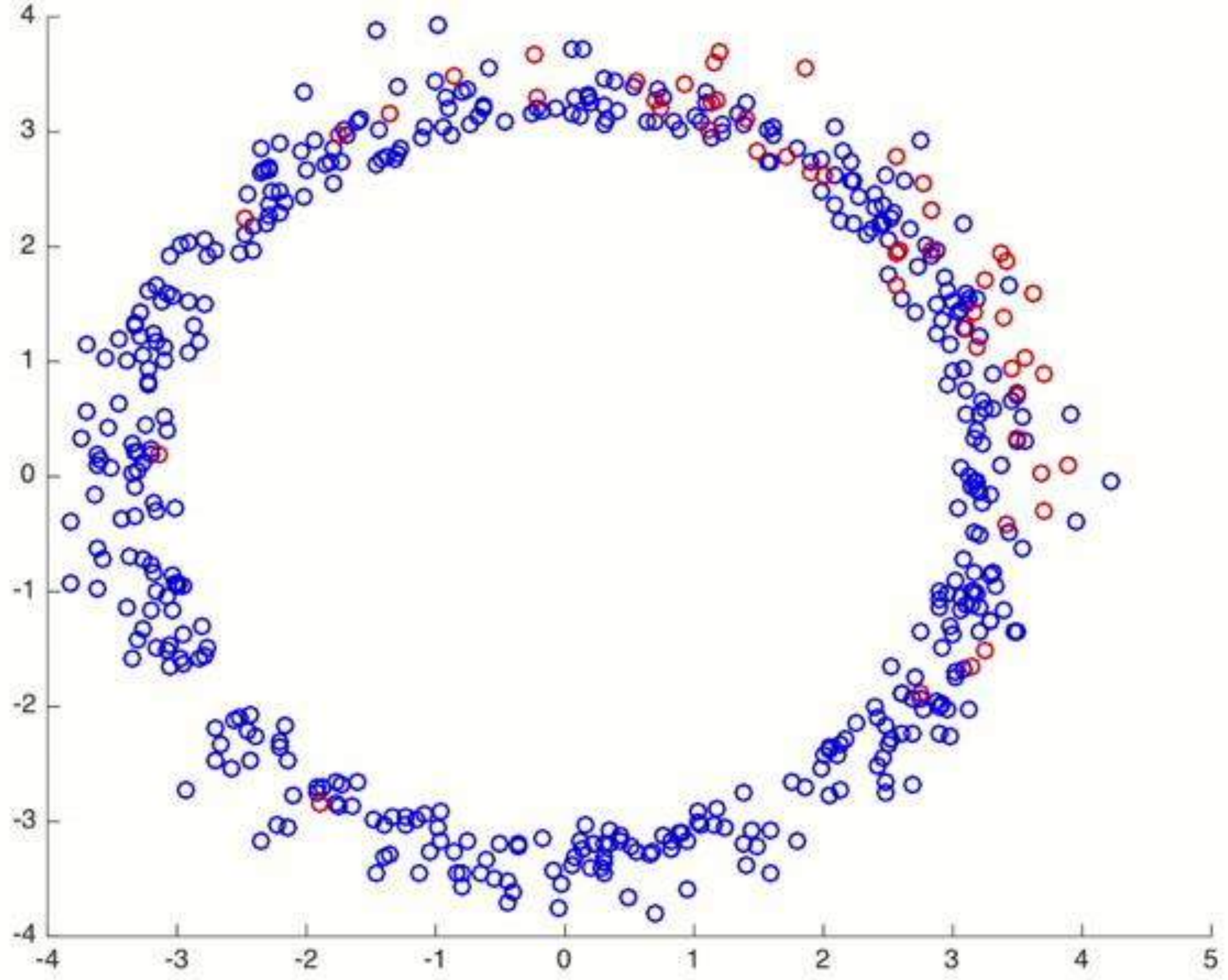
All known approaches for high-dimensional mean estimation either

1. Are computationally intractable in high dimensions; or
2. Lose accuracy factors which depend polynomially on the dimension

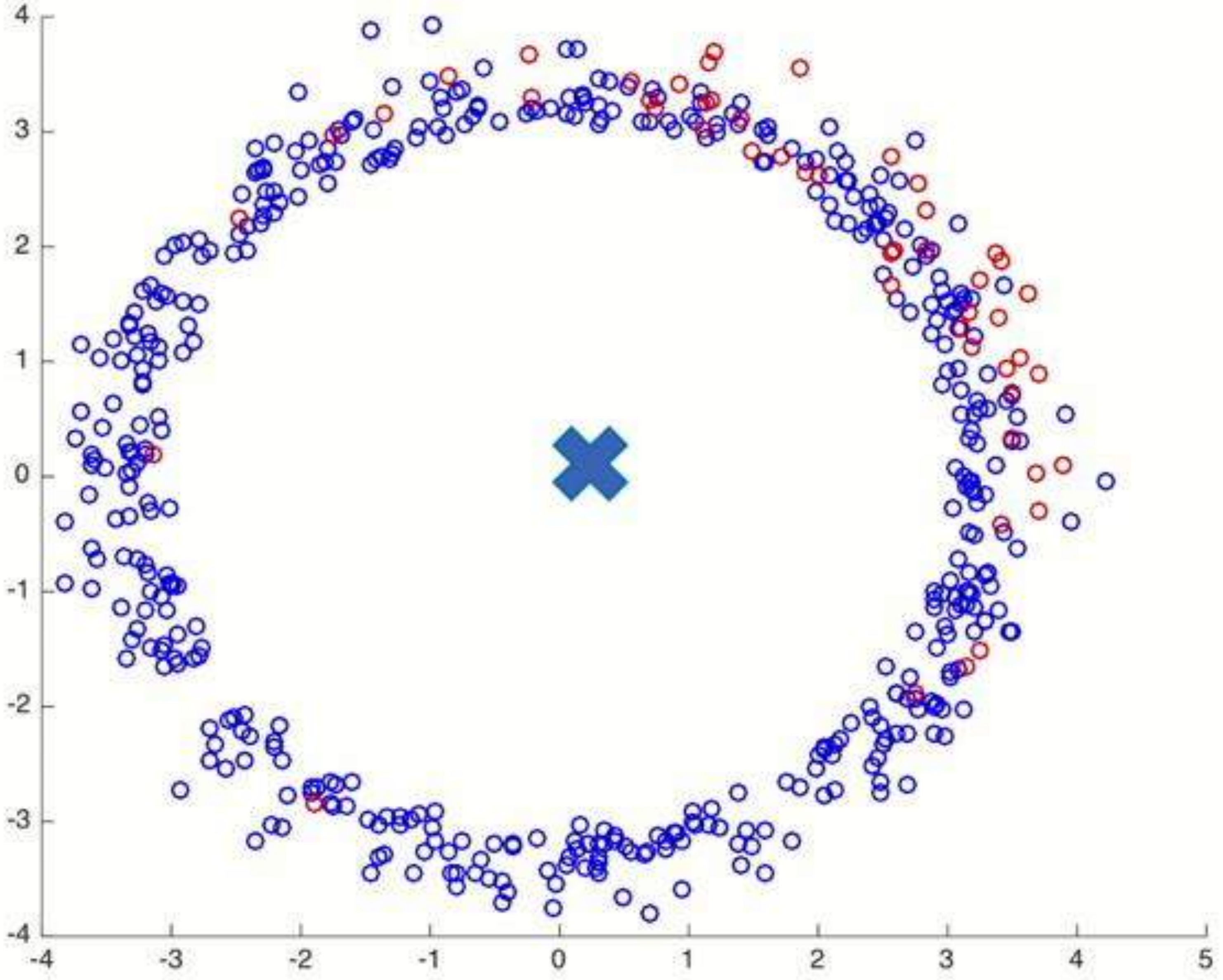
Is efficient robust estimation possible in high dimensions? **Yes!**

Global corruptions?

Idea: If the corruptions move the mean…



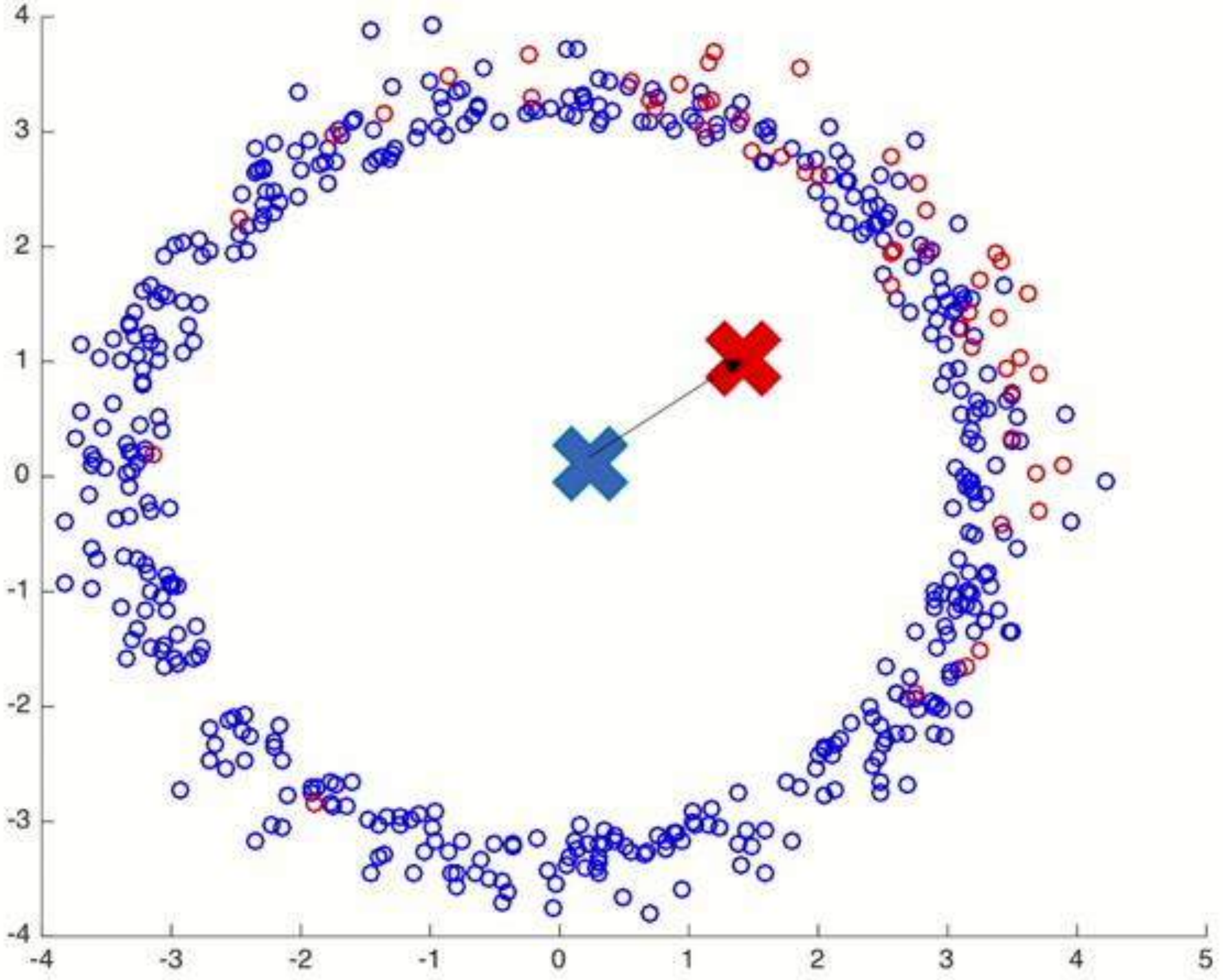
Global corruptions?



Idea: If the corruptions move the mean…

They also shift the covariance matrix!

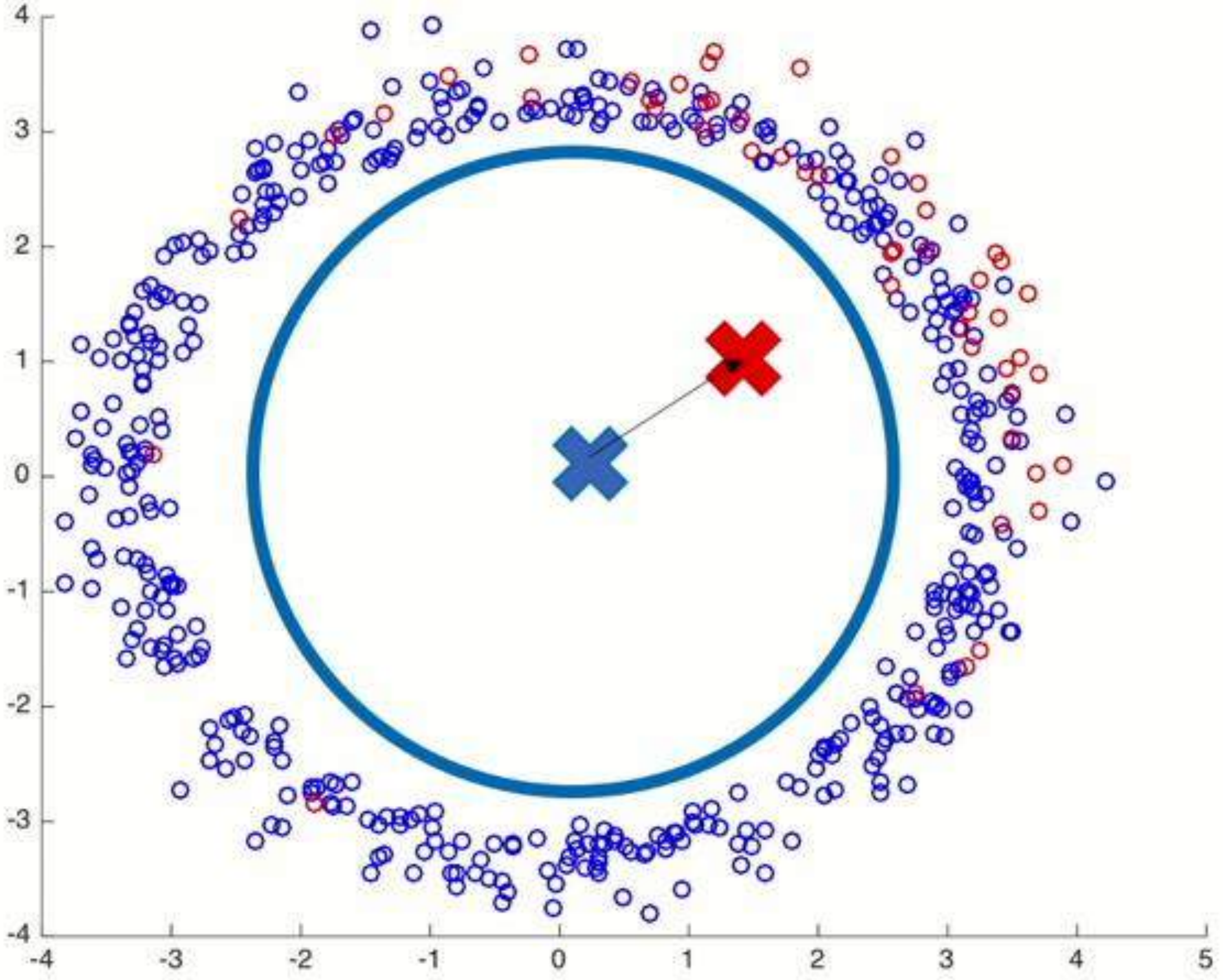
Global corruptions?



Idea: If the corruptions move the mean…

They also shift the covariance matrix!

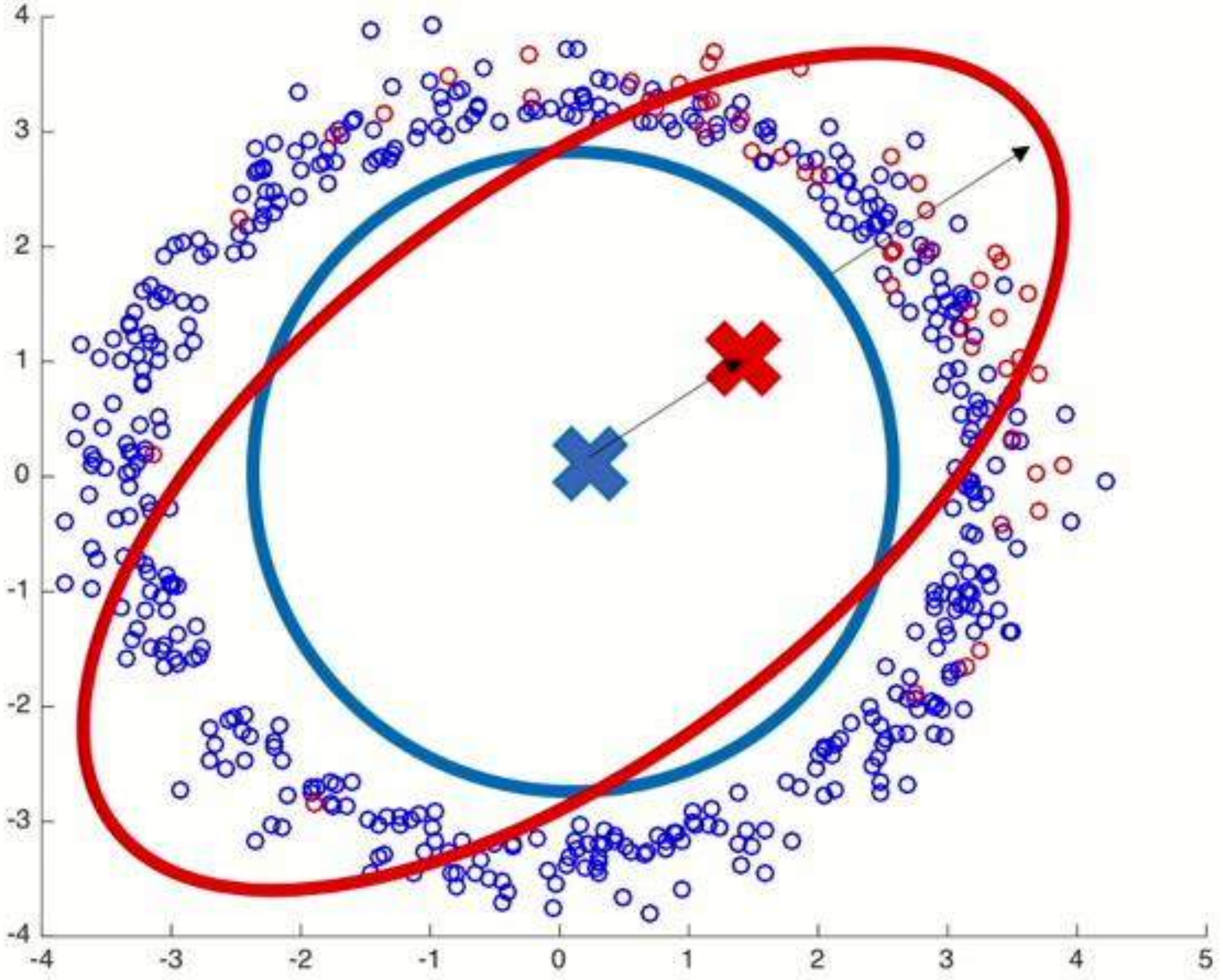
Global corruptions?



Idea: If the corruptions move the mean…

They also shift the covariance matrix!

Global corruptions?



Idea: If the corruptions move the mean…

They also shift the covariance matrix!

Efficient algorithms via spectral signatures

"Fundamental Lemma of Efficient Robust Estimation": Suppose you have an ε-corrupted data set of size n in d dimensions, and the good data comes from a distribution with mean μ and covariance $\Sigma \preccurlyeq I$.

Let $\hat{\mu}$ and $\hat{\Sigma}$ be the mean and covariance of the corrupted data. Then with high probability, we have

$$\|\hat{\mu} - \mu\|_2 \leq \tilde{O}\left(\sqrt{\frac{d}{n}}\right) + O\left(\sqrt{\varepsilon \cdot \|\hat{\Sigma}\|_2}\right).$$

Efficient algorithms via spectral signatures

Two consequences of the lemma:

1. If the top eigenvalue of the empirical covariance of your corrupted data is small, then the corruptions aren't "too bad".
   - Can just output the empirical mean!

2. If the top eigenvalue is large, then it can only be large because the bad points are too big in this direction.
   - The top eigenvector gives a direction where the bad points are prominent!

Filtering: A Simple Meta-Algorithm

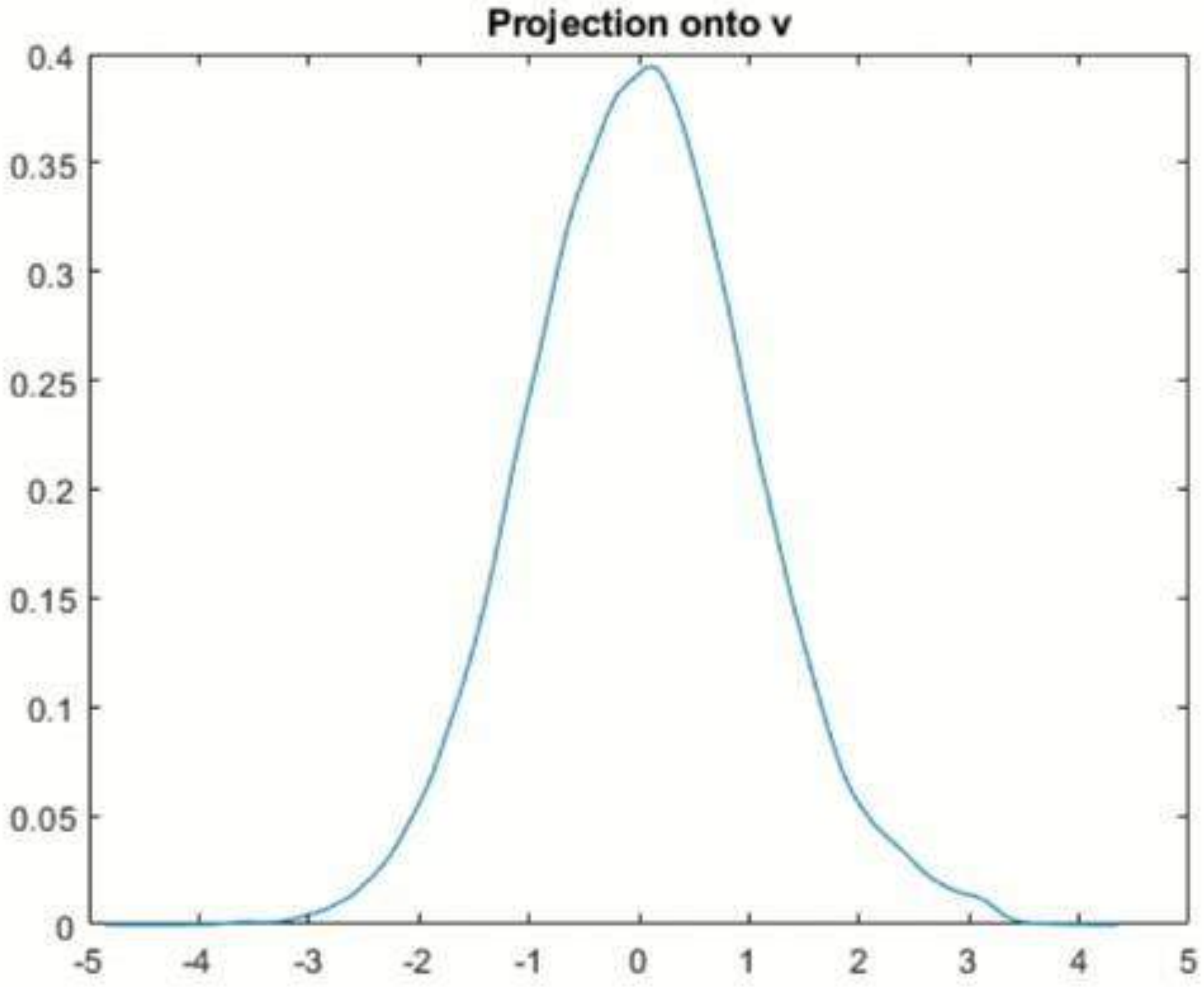
Given corrupted dataset S

- Let $\hat{\mu}$ be the empirical mean of S
- Let $\hat{\Sigma}$ be the empirical covariance of S
- $(\lambda, v) \leftarrow$ top eigenvalue/vector of $\hat{\Sigma}$
- If λ is not too large
  - Output $\hat{\mu}$
- Otherwise,
  - Project the data points in the direction of v
  - Remove (or downweight) the largest data points in this direction

Filtering: A Simple Meta-Algorithm

Given corrupted dataset S

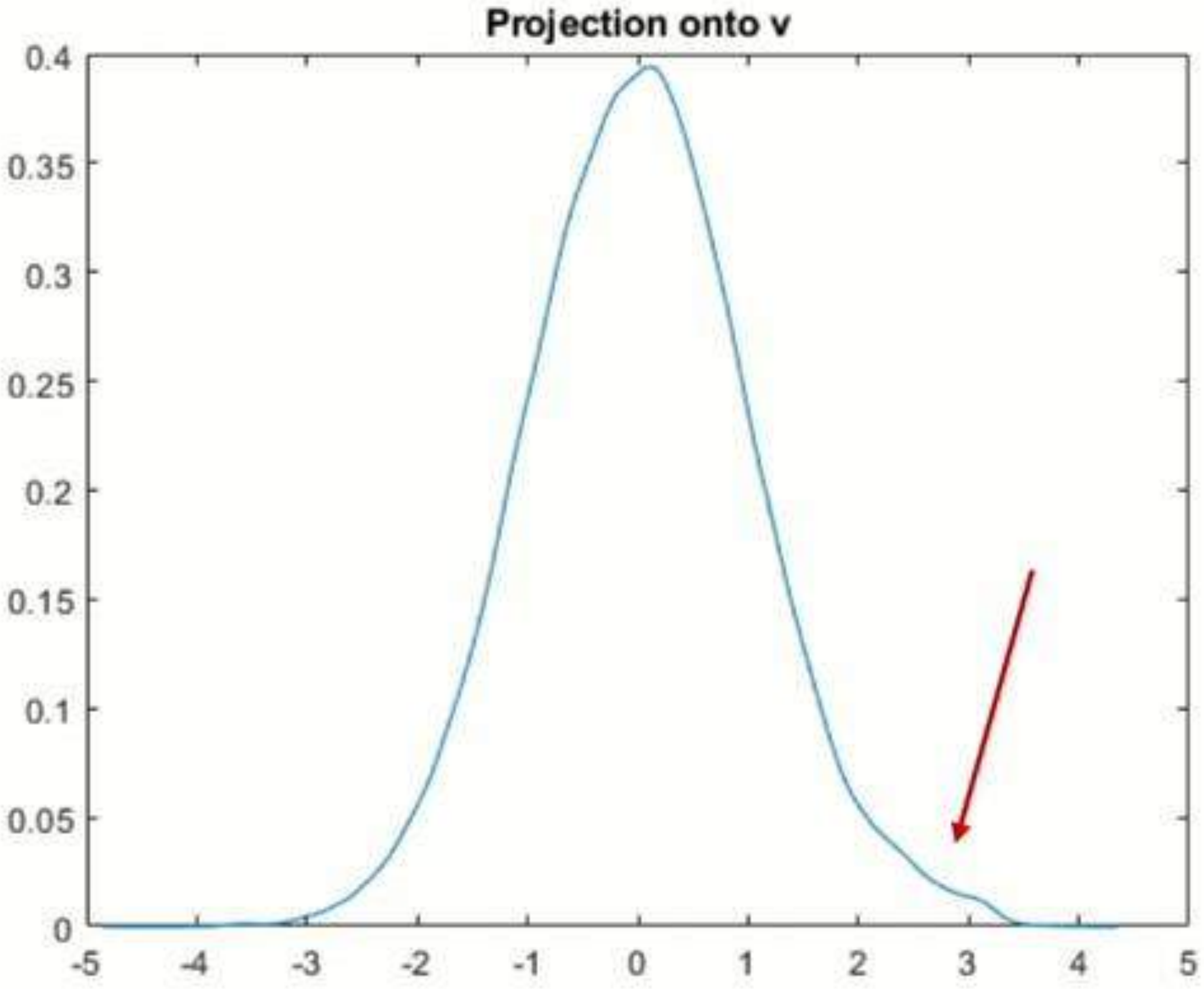
- Let $\hat{\mu}$ be the empirical mean of S
- Let $\hat{\Sigma}$ be the empirical covariance of S
- $(\lambda, v) \leftarrow$ top eigenvalue/vector of $\hat{\Sigma}$
- If λ is not too large
  - Output $\hat{\mu}$
- Otherwise,
  - Project the data points in the direction of v
  - Remove (or downweight) the largest data points in this direction



Filtering: A Simple Meta-Algorithm

Given corrupted dataset S

- Let $\hat{\mu}$ be the empirical mean of S
- Let $\hat{\Sigma}$ be the empirical covariance of S
- $(\lambda, v) \leftarrow$ top eigenvalue/vector of $\hat{\Sigma}$
- If λ is not too large
  - Output $\hat{\mu}$
- Otherwise,
  - Project the data points in the direction of v
  - Remove (or downweight) the largest data points in this direction



A single iteration runs in nearly linear time!

Filtering with bounded covariance

Given corrupted dataset S

- Let $\hat{\mu}$ be the empirical mean of S
- Let $\hat{\Sigma}$ be the empirical covariance of S
- $(\lambda, v) \leftarrow$ top eigenvalue/vector of $\hat{\Sigma}$
- If λ is not too large
  - Output $\hat{\mu}$
- Otherwise,
  - Project the data points in the direction of v
  - Remove the largest data points in this direction

Let D be a distribution with mean μ and covariance $\Sigma \preccurlyeq I$.

Assume: S is an ε-corrupted set of samples from D

Goal: Recover $\hat{\mu}$ so that whp

$$\|\hat{\mu} - \mu\| \leq C \cdot \sqrt{\varepsilon}$$

Filtering with bounded covariance

Given corrupted dataset S

- Let $\hat{\mu}$ be the empirical mean of S
- Let $\hat{\Sigma}$ be the empirical covariance of S
- $(\lambda, v) \leftarrow$ top eigenvalue/vector of $\hat{\Sigma}$
- If $\lambda \leq 9$
  - Output $\hat{\mu}$
- Otherwise,
  - Let $\tau(X) = \langle v, X - \hat{\mu} \rangle^2$, let $\tau_{\max} = \max_{X \in S} \tau(X)$
  - Remove the largest data points in this direction

Let D be a distribution with mean μ and covariance $\Sigma \preccurlyeq I$.

Assume: S is an ε-corrupted set of samples from D

Goal: Recover $\hat{\mu}$ so that whp

$$\|\hat{\mu} - \mu\| \leq C \cdot \sqrt{\varepsilon}$$

Filtering with bounded covariance

Given corrupted dataset S

- Let $\hat{\mu}$ be the empirical mean of S
- Let $\hat{\Sigma}$ be the empirical covariance of S
- $(\lambda, v) \leftarrow$ top eigenvalue/vector of $\hat{\Sigma}$
- If $\lambda \leq 9$
  - Output $\hat{\mu}$
- Otherwise,
  - Let $\tau(X) = \langle v, X - \hat{\mu} \rangle^2$, let $\tau_{\max} = \max_{X \in S} \tau(X)$
  - Remove (or downweight) each point $X \in S$ independently with probability $\tau(X)/\tau_{\max}$
  - Output the remaining set of points

Let D be a distribution with mean μ and covariance $\Sigma \preccurlyeq I$.

Assume: S is an ε-corrupted set of samples from D

Goal: Recover $\hat{\mu}$ so that whp

$$\|\hat{\mu} - \mu\| \leq C \cdot \sqrt{\varepsilon}$$

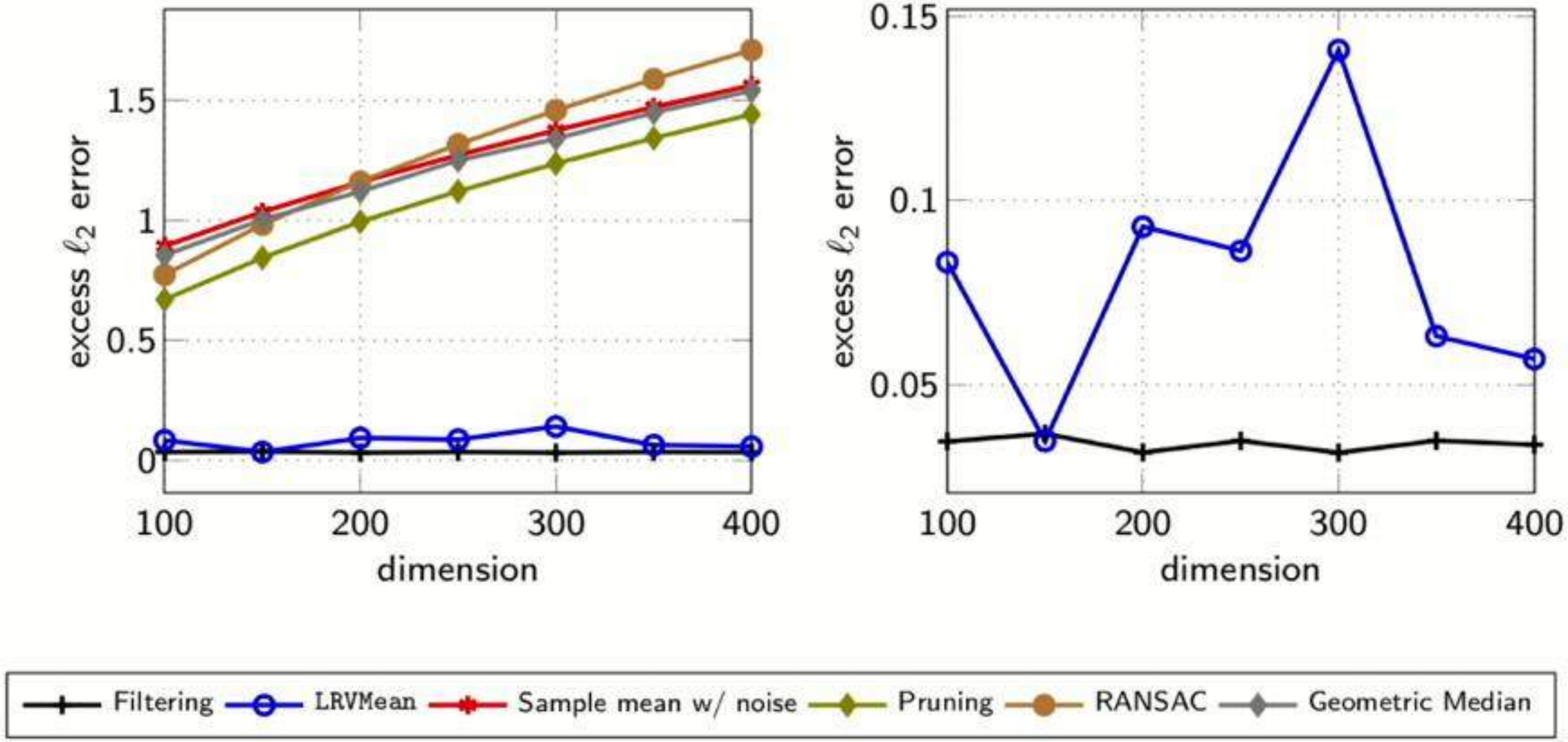
Our Results

Given an ε-corrupted set of samples that is sufficiently large from...	...we can efficiently get an estimate of the true mean to ℓ_2 error:
a distribution with bounded second moment	$O(\sqrt{\varepsilon})$ [LRV16, DKKLMS16, DKKLMS17]
a Gaussian (or sub-Gaussian distribution) with identity covariance	$O(\varepsilon\sqrt{\log 1/\varepsilon})$ [DKKLMS17,SCV17]
a Gaussian with unknown covariance	$O(\varepsilon \log 1/\varepsilon)$ [DKKLMS16]
a "nice" distribution with bounded t-th moments	$O(\varepsilon^{1-1/t})$ [HL18, KS18]

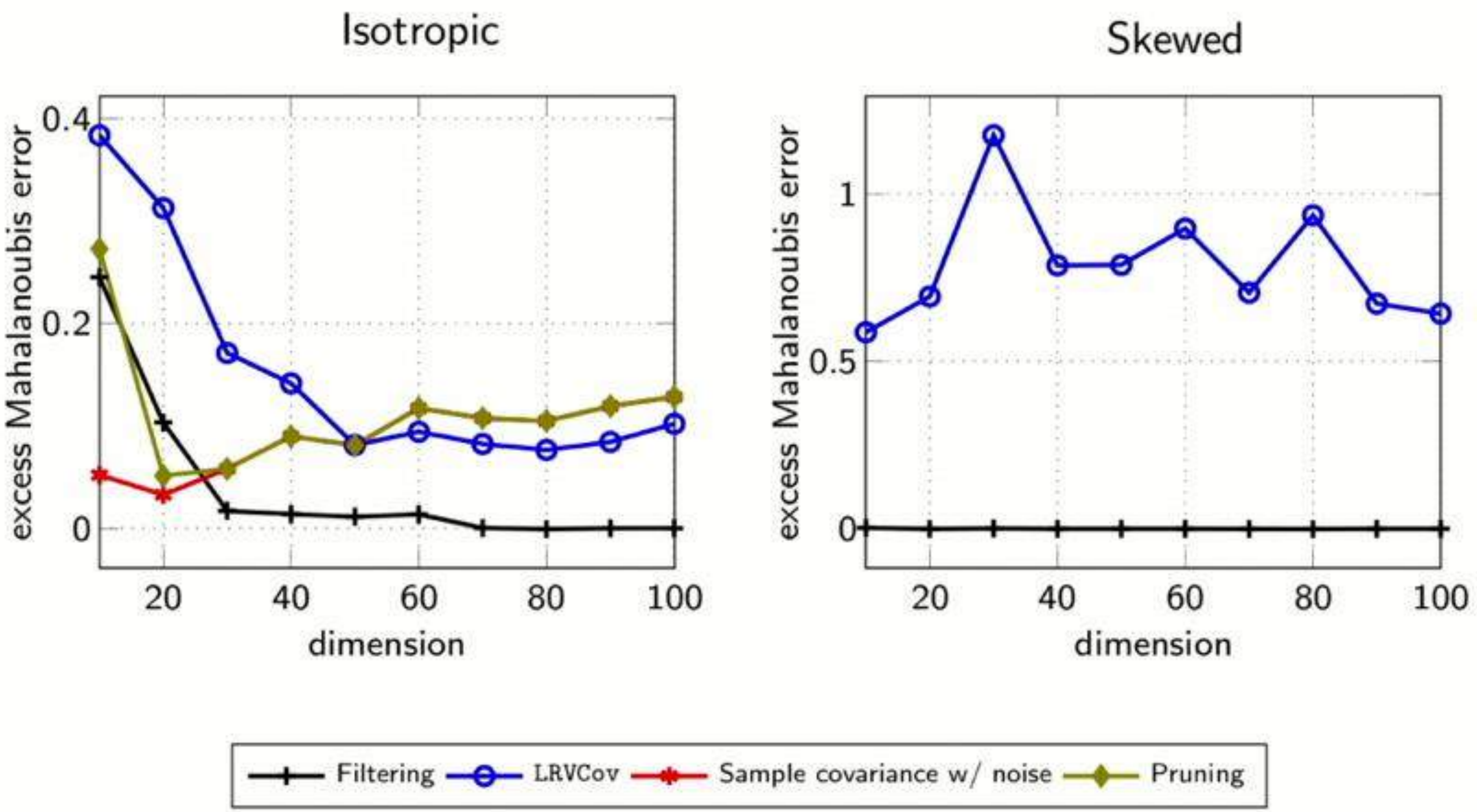
For all cases, these are the first efficient dimension-independent guarantees!

Also sparsity [L17, DBS17], list learning [CSV17, MV17], graphical models [DKS18], general norms [SCV17], federated learning [QV17], sparse regression [KKM18, CLL19] etc…

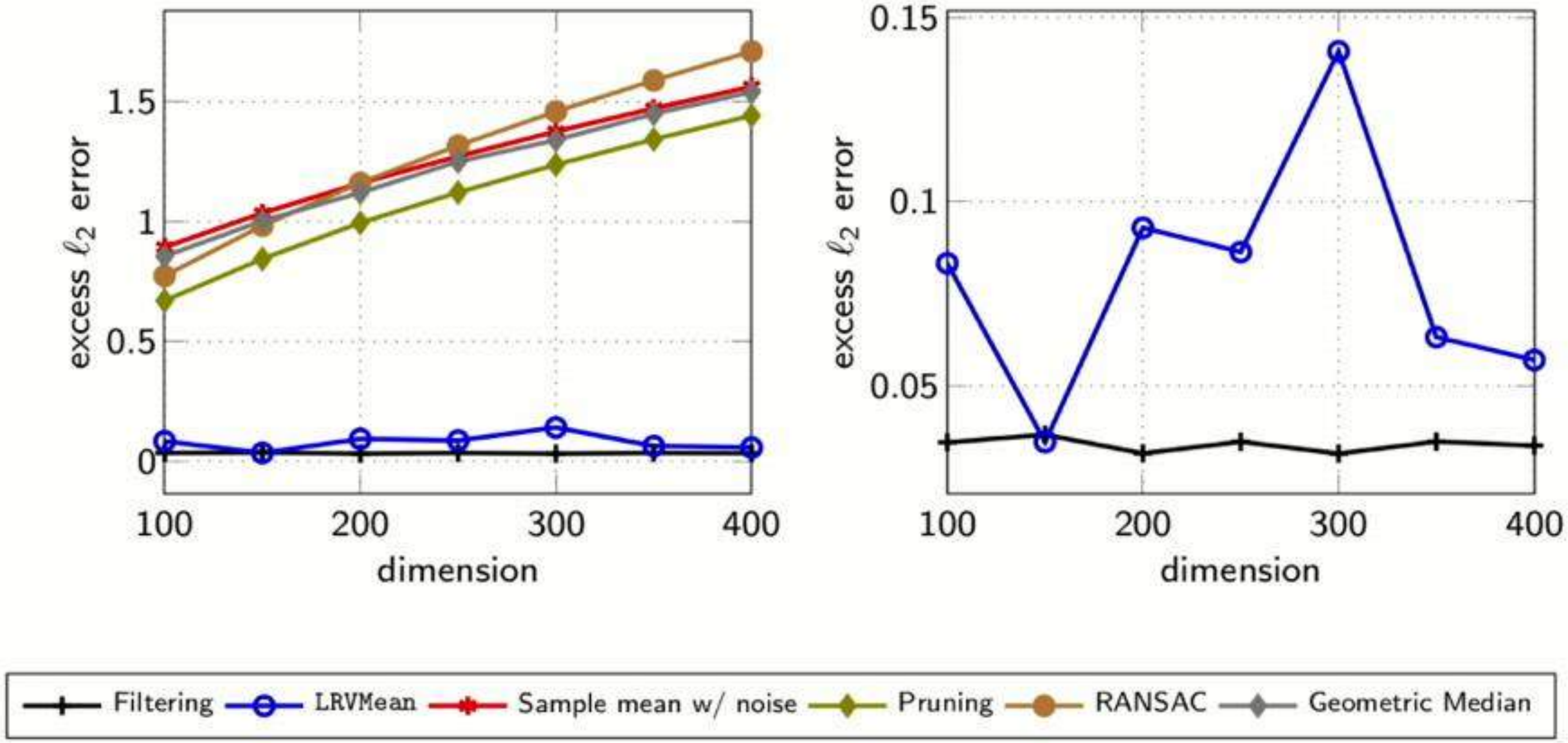
Synthetic Experiments, Unknown Mean



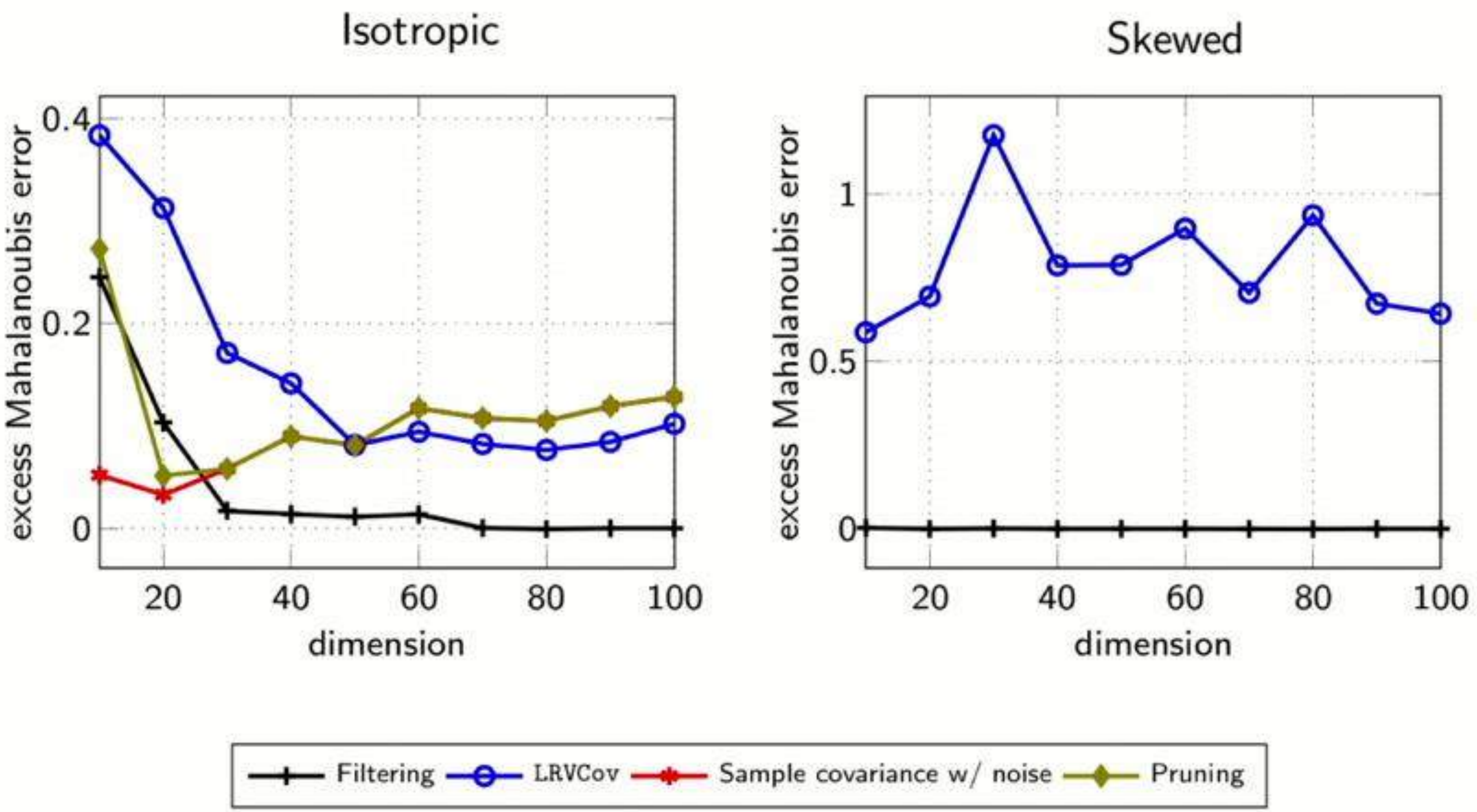
Synthetic Experiments, Unknown Covariance



Synthetic Experiments, Unknown Mean

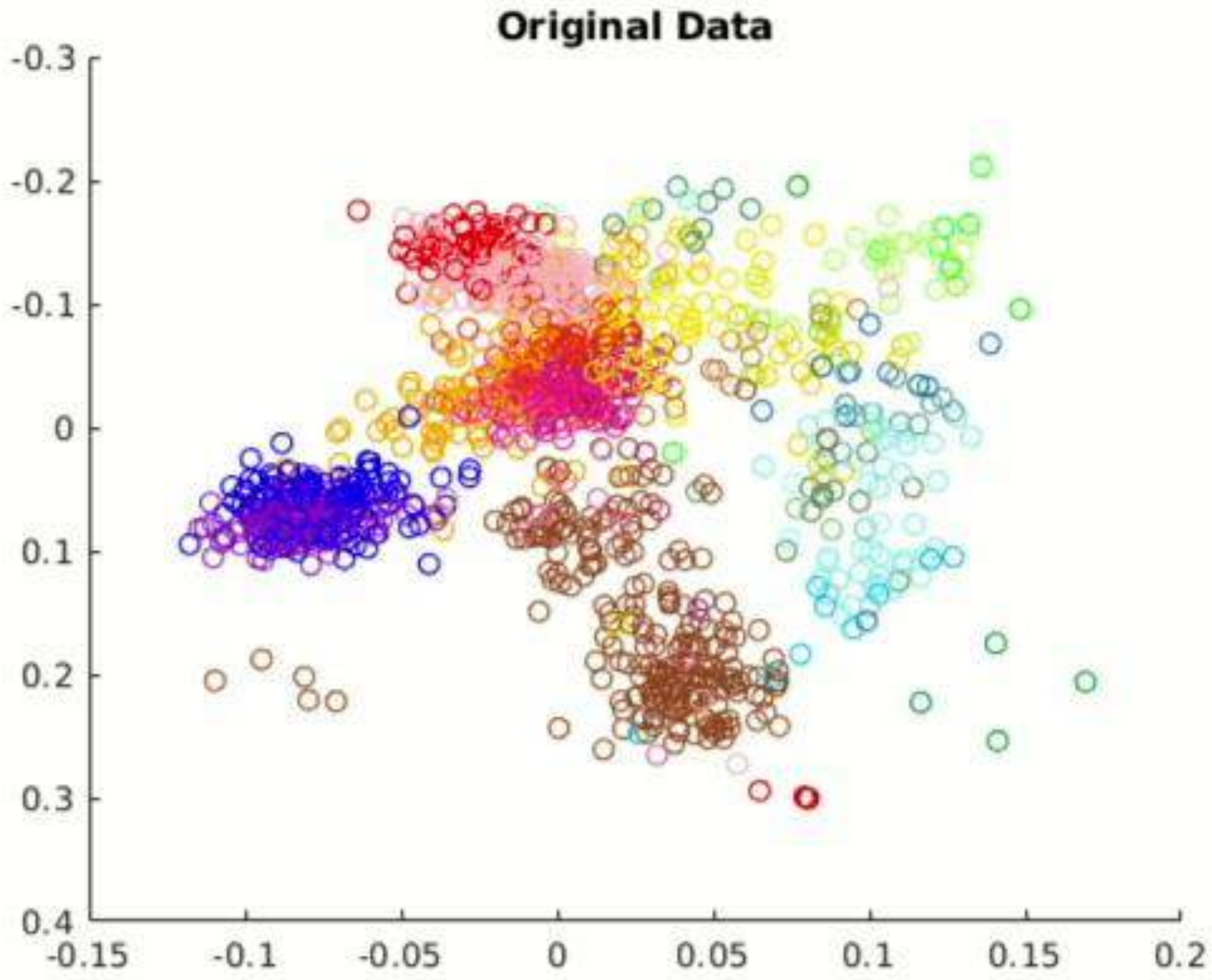


Synthetic Experiments, Unknown Covariance



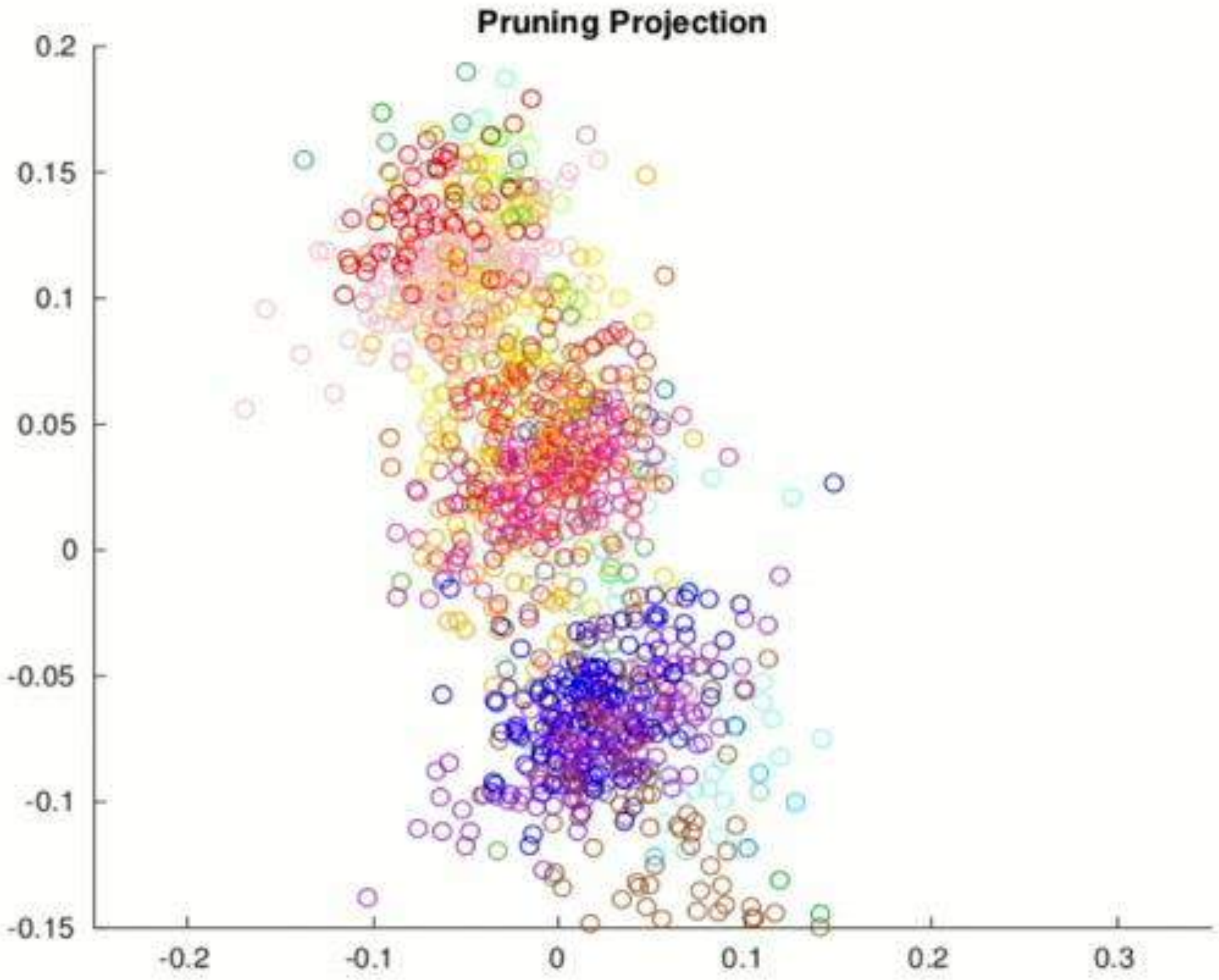
Gene Expression PCA Contains Europe

- Genes Mirror Geography in Europe. [Novembre et al.], *Nature* '08



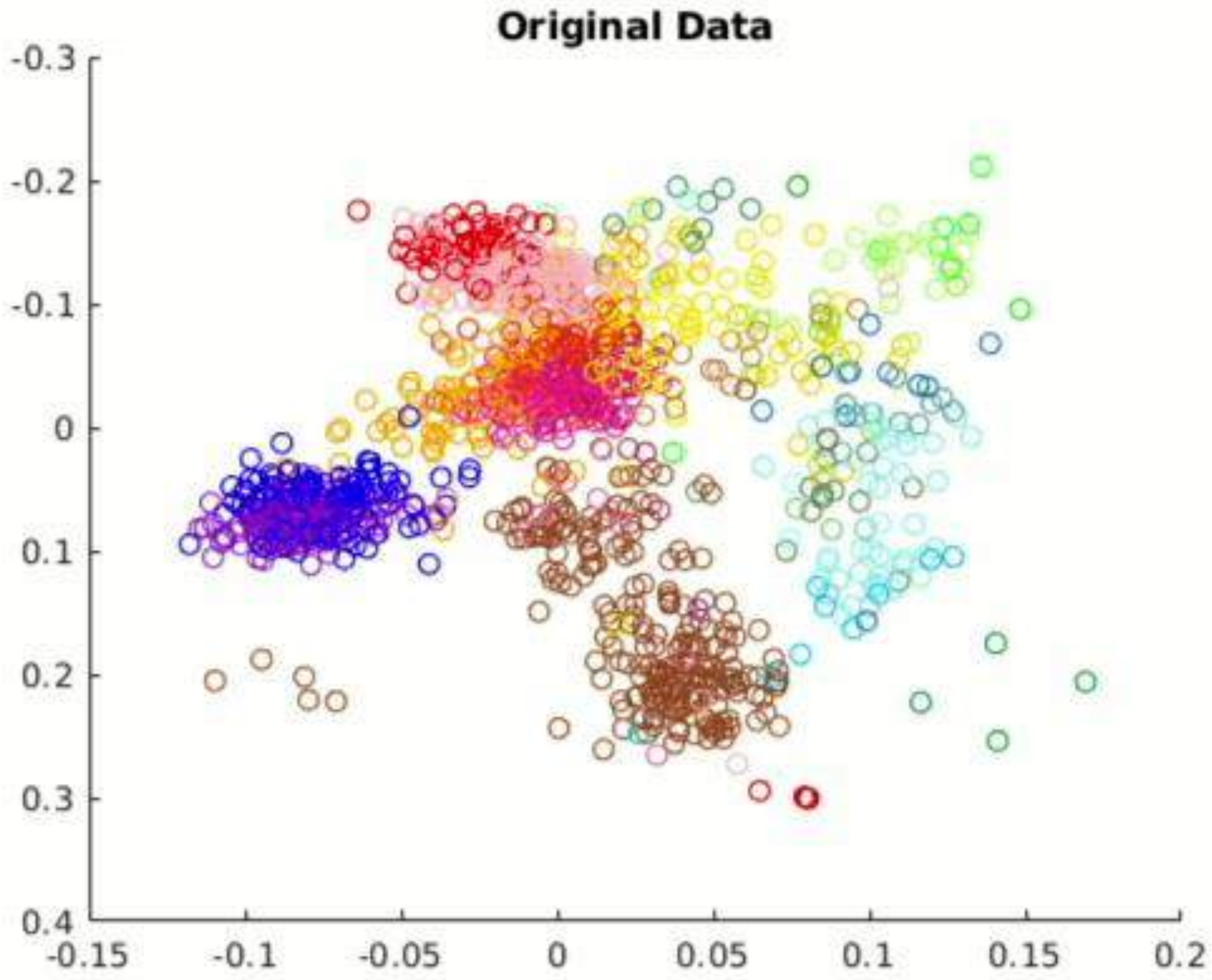
Naively, Corruptions Destroy Europe

- Genes Mirror Geography in Europe. [Novembre et al.'08]



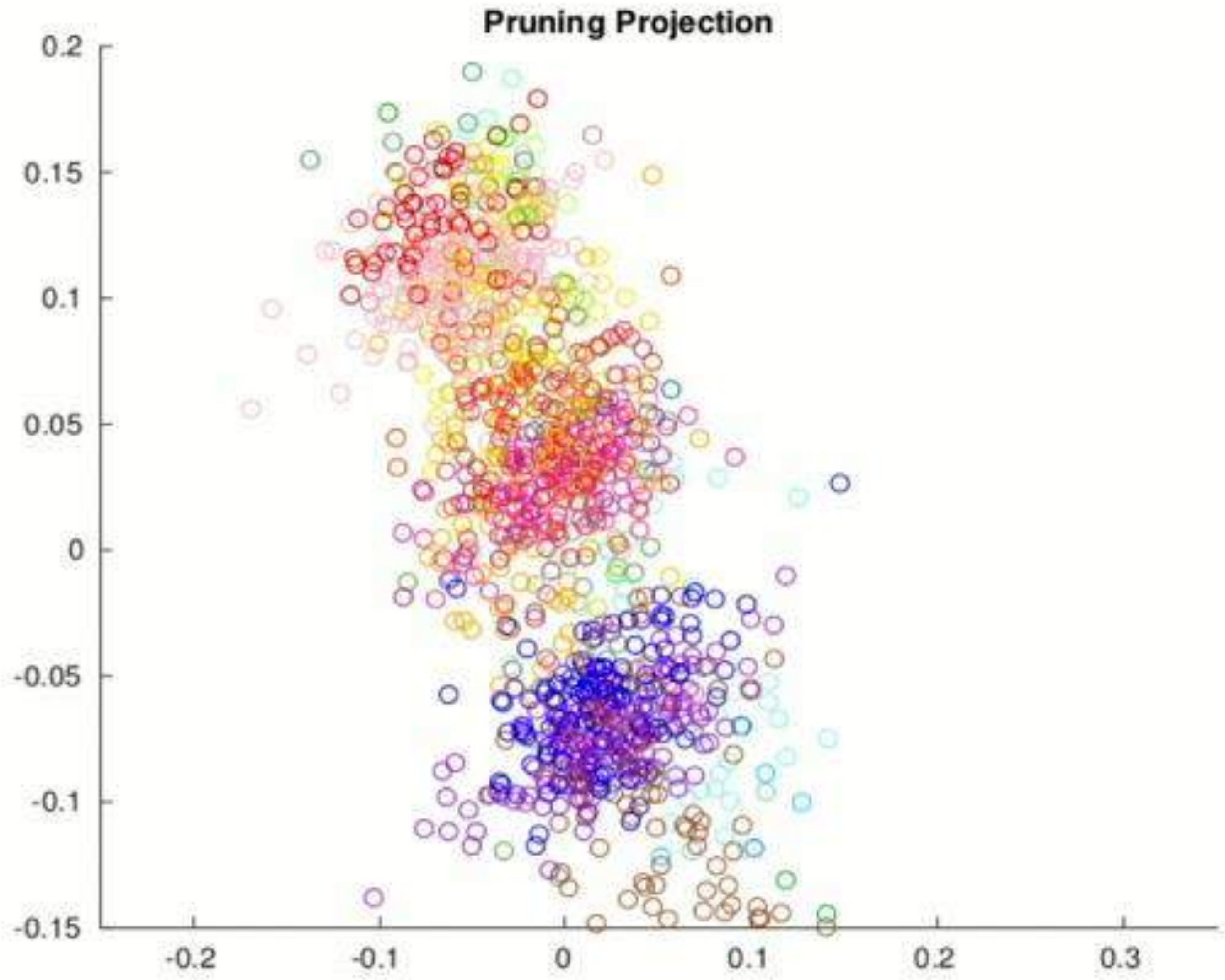
Gene Expression PCA Contains Europe

- Genes Mirror Geography in Europe. [Novembre et al.], *Nature* '08



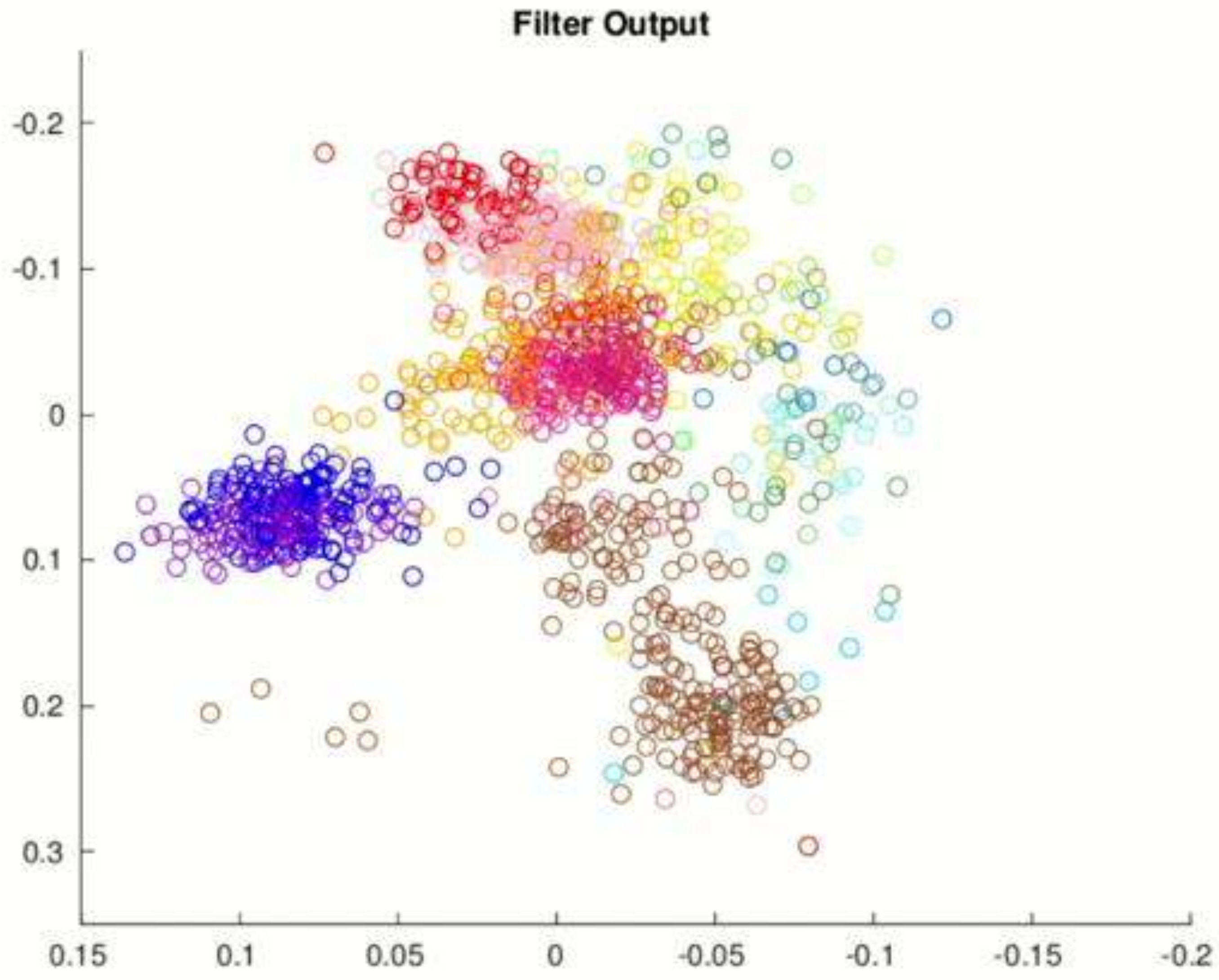
Naively, Corruptions Destroy Europe

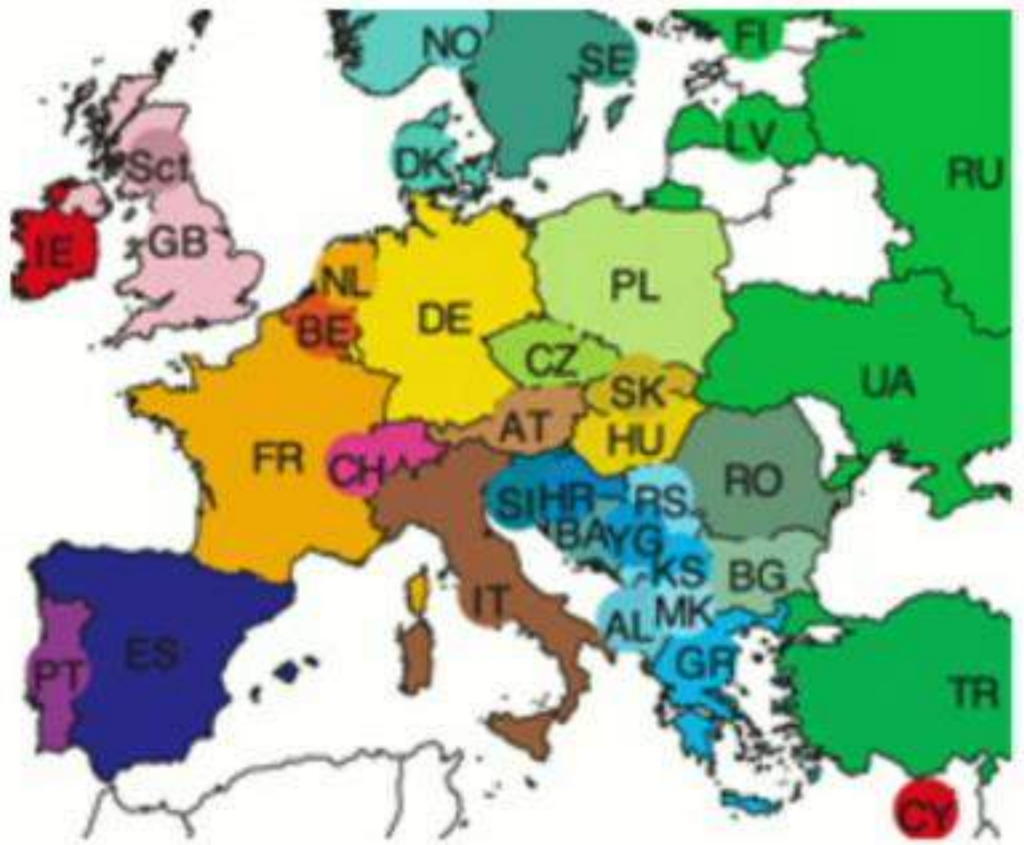
- Genes Mirror Geography in Europe. [Novembre et al.'08]



Our Algorithms Fix Europe!

- Genes Mirror Geography in Europe. [Novembre et al.'08]





Further directions

- Are these error rates tight for efficient algorithms?
  - **Maybe?** [Hopkins-L'19]

- Previous methods require potentially $O(d)$ passes over the data. Is this avoidable?
  - **Yes!** Nearly linear time algorithms [Hopkins-L'19]

- Can we use these methods as new principled outlier detections methods?

Further directions

- Are these error rates tight for efficient algorithms?
  - **Maybe?** [Hopkins-L'19]

- Previous methods require potentially $O(d)$ passes over the data. Is this avoidable?
  - **Yes!** Nearly linear time algorithms [Hopkins-L'19]

- Can we use these methods as new principled outlier detections methods?
  - **Yes…but still much to explore!**

Beyond robust statistics

Can we "robust-ify" more complicated objectives, like supervised learning?

e.g. regression, SVM

These problems can be phrased in the framework of stochastic optimization

Given a loss function $\ell(X, w)$ and a distribution $\mathcal{D}$ over X, minimize

$$f(w) = \mathbb{E}_{X \sim \mathcal{D}}\left[\ell(X, w)\right]$$

Beyond robust statistics

Can we "robust-ify" more complicated objectives, like supervised learning?

e.g. regression, SVM

These problems can be phrased in the framework of stochastic optimization

Given a loss function $\ell(X, w)$ and a distribution $\mathcal{D}$ over X, minimize

$$f(w) = \mathbb{E}_{X \sim \mathcal{D}}\left[\ell(X, w)\right]$$

Challenge: Given ε-corrupted samples from $\mathcal{D}$, minimize f

SEVER: Robust stochastic optimization

[Diakonikolas, Kamath, Kane, L, Steinhardt, Stewart], manuscript

First try: just run stochastic gradient descent using robust estimates

Recall:

$$w_{t+1} \leftarrow w_t - \eta_t \cdot \nabla \ell(X_t, w_t),$$

This works because $\mathbb{E}[\nabla \ell(X_t, w_t)] = \nabla f(w_t)$ when data is uncorrupted

SEVER: Robust stochastic optimization

[Diakonikolas, Kamath, Kane, L, Steinhardt, Stewart], manuscript

First try: just run stochastic gradient descent using robust estimates

Recall:

$$w_{t+1} \leftarrow w_t - \eta_t \cdot g_t,$$

where g_t is a robust estimate of $\nabla f(w_t)$

How to do this in the presence of noise?

SEVER: Robust stochastic optimization

[Diakonikolas, Kamath, Kane, L, Steinhardt, Stewart], manuscript

First try: just run stochastic gradient descent using robust estimates

Recall:

$$w_{t+1} \leftarrow w_t - \eta_t \cdot g_t,$$

where g_t is a robust estimate of $\nabla f(w_t)$

How to do this in the presence of noise?

This works great in theory....but slow in practice

SEVER: Robust stochastic optimization

[Diakonikolas, Kamath, Kane, L, Steinhardt, Stewart], manuscript

First try: just run stochastic gradient descent using robust estimates

Recall:

$$w_{t+1} \leftarrow w_t - \eta_t \cdot g_t,$$

where g_t is a robust estimate of $\nabla f(w_t)$

How to do this in the presence of noise?

This works great in theory....but slow in practice

Better: only filter at minimizer of the empirical risk!

SEVER: Robust stochastic optimization

[Diakonikolas, Kamath, Kane, L, Steinhardt, Stewart], manuscript

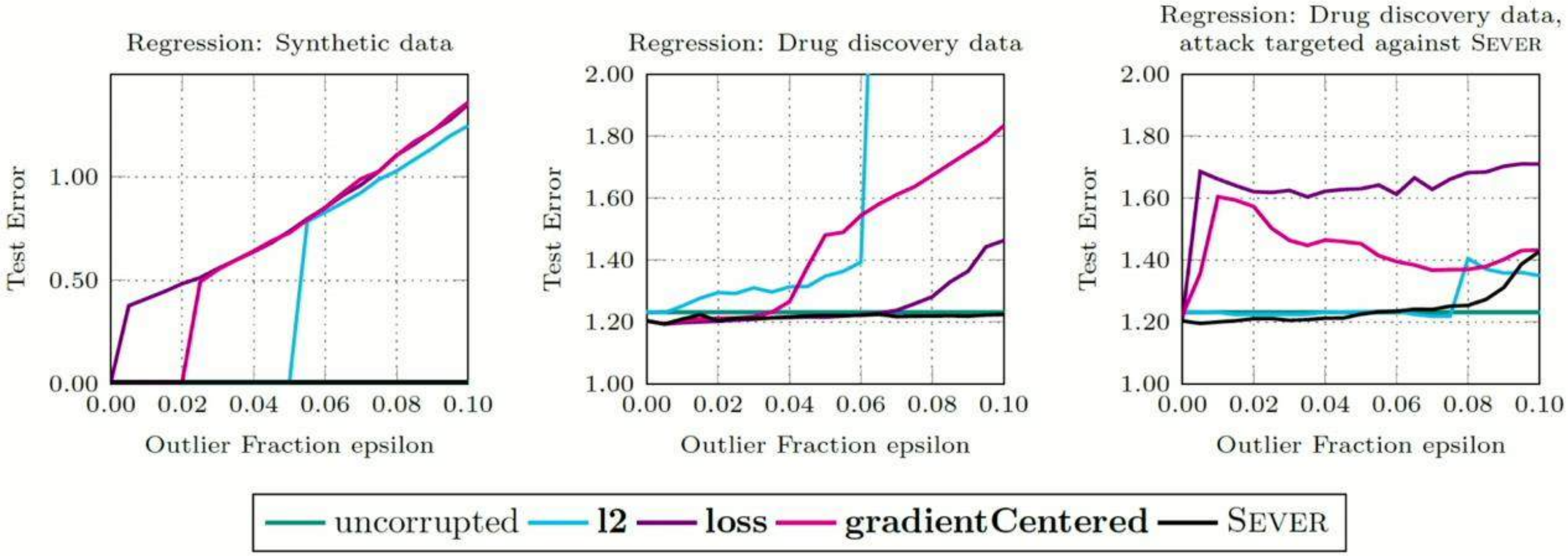
Theorem: Suppose ℓ is convex, and $\mathsf{Cov}\left[\nabla \ell(X, w)\right] \preccurlyeq \sigma^2 I$. Under mild assumptions on $\mathcal{D}$, then SEVER outputs a $\widehat{w}$ so that w.h.p.

$$f(\widehat{w}) - \min_{w} f(w) < O\left(\sqrt{\sigma^2 \varepsilon}\right).$$

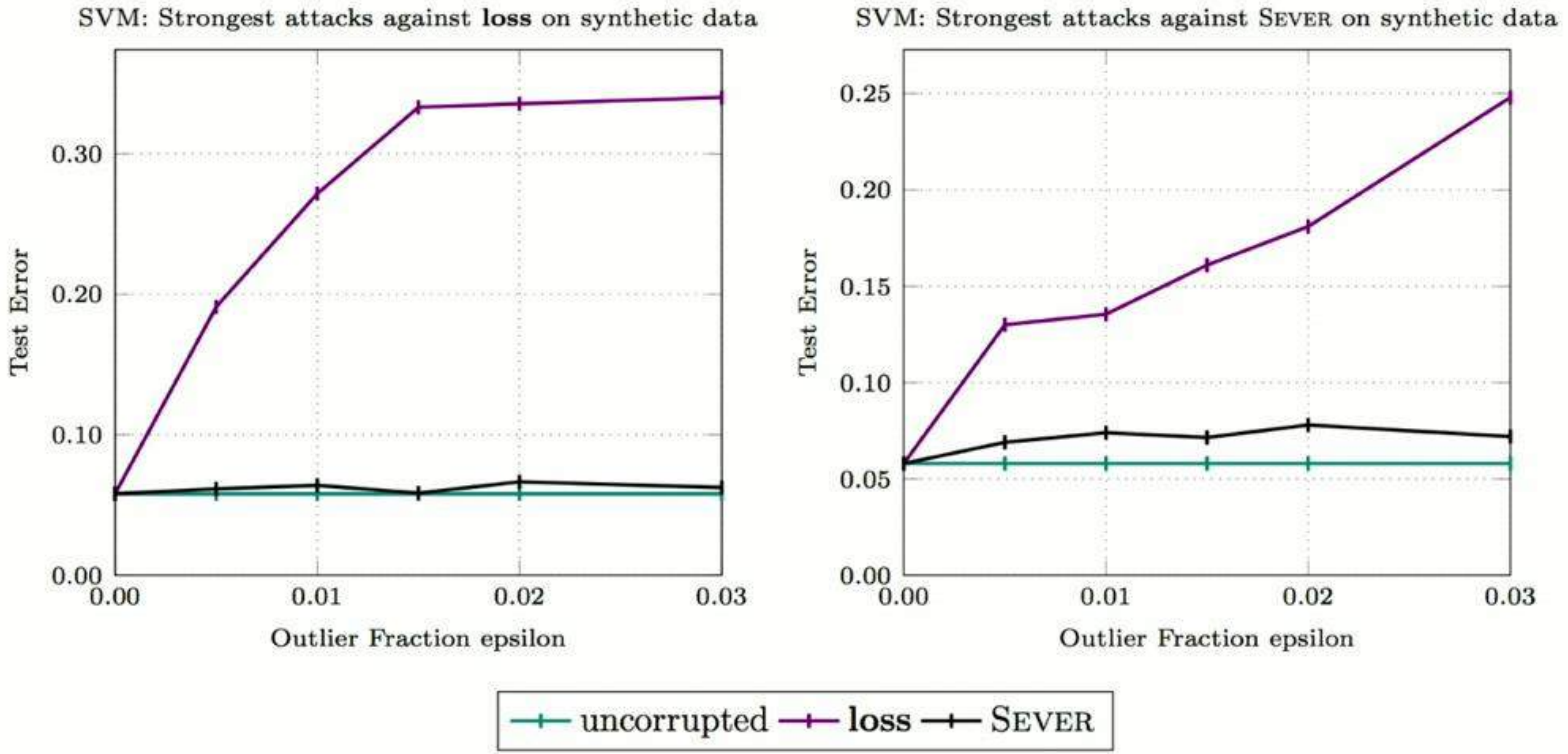
Sample complexity / runtime bounds are polynomial but not super tight

For specific instances (e.g. SVM, regression), we obtain tighter bounds

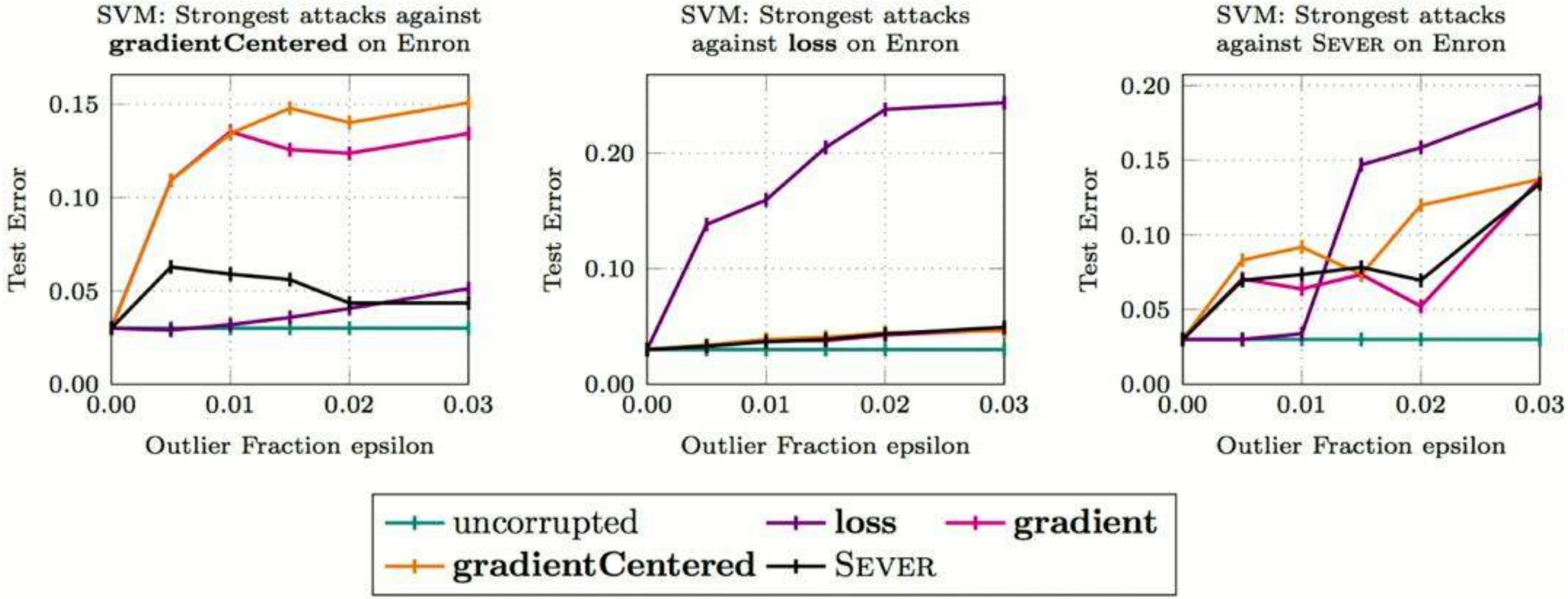
Performance for ridge regression



Performance for SVMs, synthetic



Performance for SVMs, real data



Beyond(er) robust statistics: backdoor attacks

[Tran, L, Madry], NeurIPS'18

Attacks against ResNet on CIFAR10:

Beyond(er) robust statistics: backdoor attacks

[Tran, L, Madry], NeurIPS'18

Attacks against ResNet on CIFAR10:

Natural | Poisoned | Natural | Poisoned

"airplane" | "bird" | "automobile" | "cat"

Beyond(er) robust statistics: backdoor attacks

[Tran, L, Madry], NeurIPS'18

Attacks against ResNet on CIFAR10:

Natural | Poisoned | Natural | Poisoned

"airplane" | "bird" | "automobile" | "cat"

These attacks convince the network that the implanted watermark is a strong signal for classification

As a result, the learned representation amplifies the signal of the watermark, creating a backdoor

Beyond robust statistics: backdoor attacks

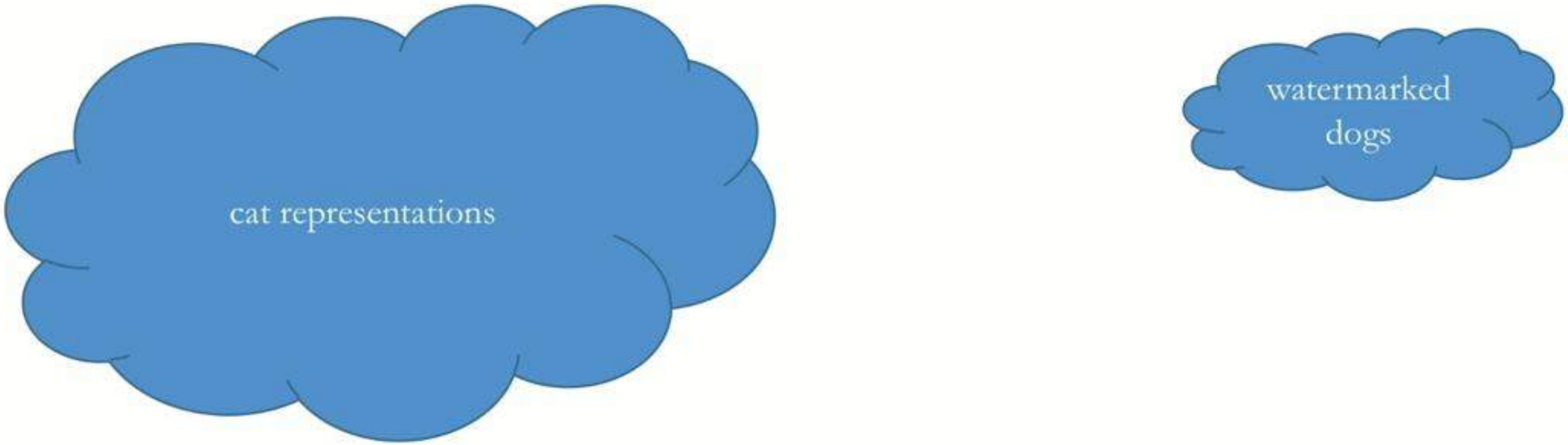
[Tran, L, Madry], NeurIPS'18

So what happens to the training set at the learned representation level?

Beyond robust statistics: backdoor attacks

[Tran, L, Madry], NeurIPS'18

So what happens to the training set at the learned representation level?



Empirically, results in a noticeable perturbation in the covariance ⇒ our algorithms can detect the corruptions!

Beyond robust statistics: backdoor attacks

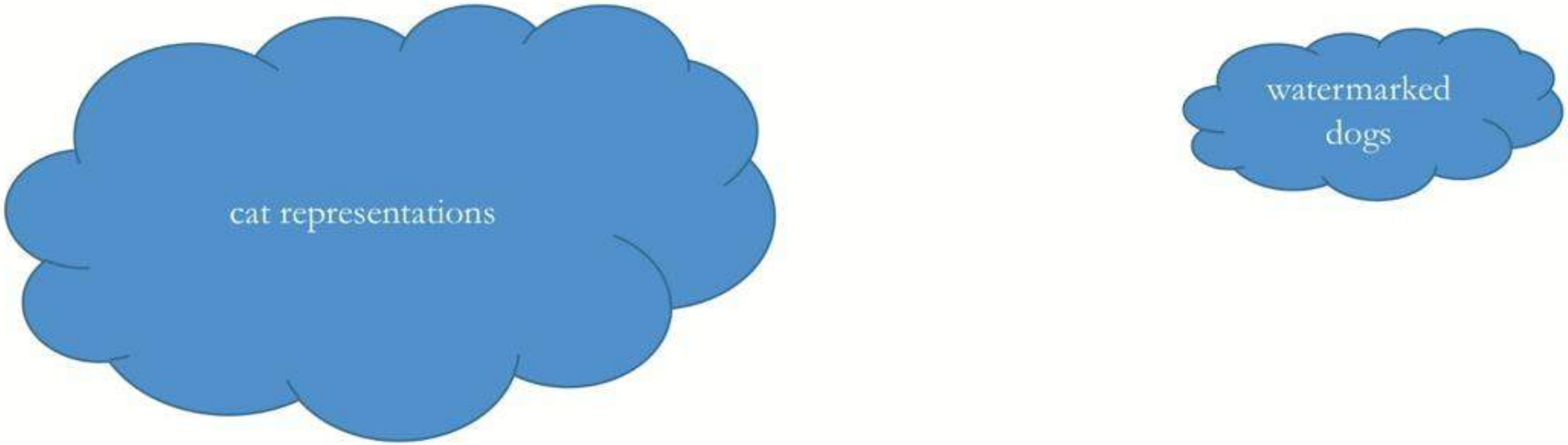
[Tran, L, Madry], NeurIPS'18

Sample	Target	Epsilon	Nat 1	Pois 1	# Pois Left	Nat 2	Pois 2	Std Pois
	bird	5%	92.27%	74.20%	57	92.64%	2.00%	1.20%
		10%	92.32%	89.80%	7	92.68%	1.50%	
	cat	5%	92.45%	83.30%	24	92.24%	0.20%	0.10%
		10%	92.39%	92.00%	0	92.44%	0.00%	
	dog	5%	92.17%	89.80%	7	93.01%	0.00%	0.00%
		10%	92.55%	94.30%	1	92.64%	0.00%	
	horse	5%	92.60%	99.80%	0	92.57%	1.00%	0.80%
		10%	92.26%	99.80%	0	92.63%	1.20%	
	cat	5%	92.86%	98.60%	0	92.79%	8.30%	8.00%
		10%	92.29%	99.10%	0	92.57%	8.20%	
	deer	5%	92.68%	99.30%	0	92.68%	1.10%	1.00%
		10%	92.68%	99.90%	0	92.74%	1.60%	
	frog	5%	92.87%	88.80%	10	92.61%	0.10%	0.30%
		10%	92.82%	93.70%	3	92.74%	0.10%	
	bird	5%	92.52%	97.90%	0	92.69%	0.00%	0.00%
		10%	92.68%	99.30%	0	92.45%	0.50%	

Beyond robust statistics: backdoor attacks

[Tran, L, Madry], NeurIPS'18

So what happens to the training set at the learned representation level?



Empirically, results in a noticeable perturbation in the covariance ⇒ our algorithms can detect the corruptions!

Beyond robust statistics: backdoor attacks

[Tran, L, Madry], NeurIPS'18

Sample	Target	Epsilon	Nat 1	Pois 1	# Pois Left	Nat 2	Pois 2	Std Pois
	bird	5%	92.27%	74.20%	57	92.64%	2.00%	1.20%
		10%	92.32%	89.80%	7	92.68%	1.50%	
	cat	5%	92.45%	83.30%	24	92.24%	0.20%	0.10%
		10%	92.39%	92.00%	0	92.44%	0.00%	
	dog	5%	92.17%	89.80%	7	93.01%	0.00%	0.00%
		10%	92.55%	94.30%	1	92.64%	0.00%	
	horse	5%	92.60%	99.80%	0	92.57%	1.00%	0.80%
		10%	92.26%	99.80%	0	92.63%	1.20%	
	cat	5%	92.86%	98.60%	0	92.79%	8.30%	8.00%
		10%	92.29%	99.10%	0	92.57%	8.20%	
	deer	5%	92.68%	99.30%	0	92.68%	1.10%	1.00%
		10%	92.68%	99.90%	0	92.74%	1.60%	
	frog	5%	92.87%	88.80%	10	92.61%	0.10%	0.30%
		10%	92.82%	93.70%	3	92.74%	0.10%	
	bird	5%	92.52%	97.90%	0	92.69%	0.00%	0.00%
		10%	92.68%	99.30%	0	92.45%	0.50%	

Towards a theory of robust machine learning

Goal: Develop a principled theory of robust machine learning

- Understanding the computational limits of robust estimation?
- What's going on with deep networks?
- Connections with other models of robustness?

Thanks!