

# Sampling Without Compromising Accuracy in Adaptive Data Analysis

Lev Reyzin (UIC)



join work with  
← Benjamin Fish (MSR Montréal),  
Benjamin Rubinstein (Melbourne) →



**talk @ Purdue**

# The problem

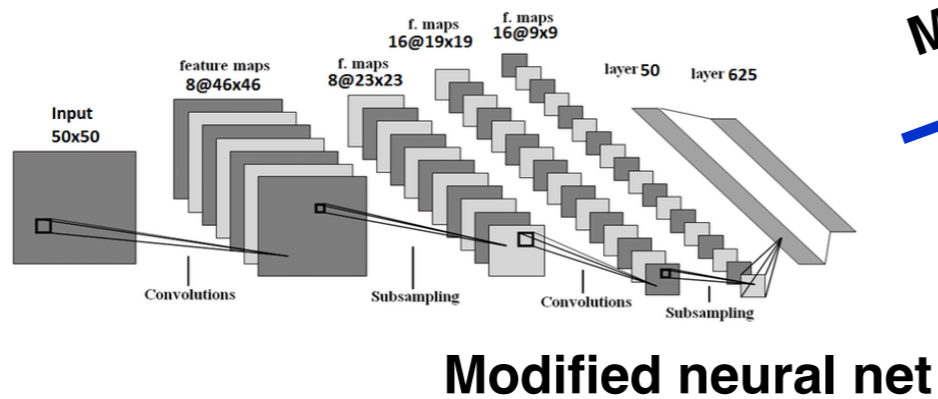
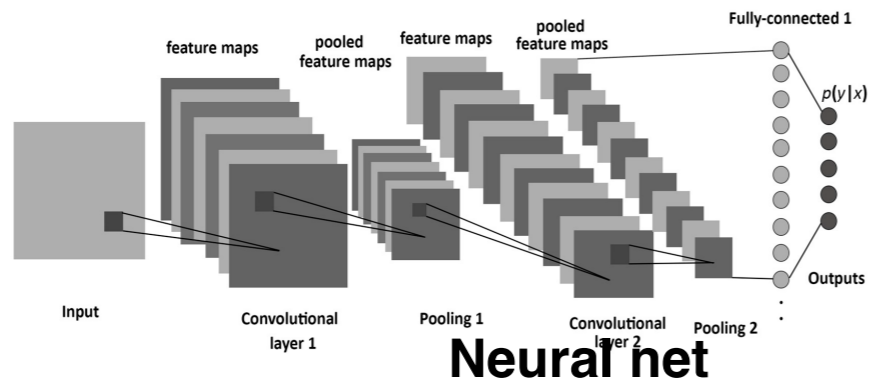
Data sets are often reused, which leads to overfitting.

# The problem

Data sets are often reused, which leads to overfitting.

*a.k.a. “false discovery,” “the garden of forking paths,” “p-hacking”*

# The problem



Measure loss

Measure loss again  
Now loss may not generalize!

Validation set

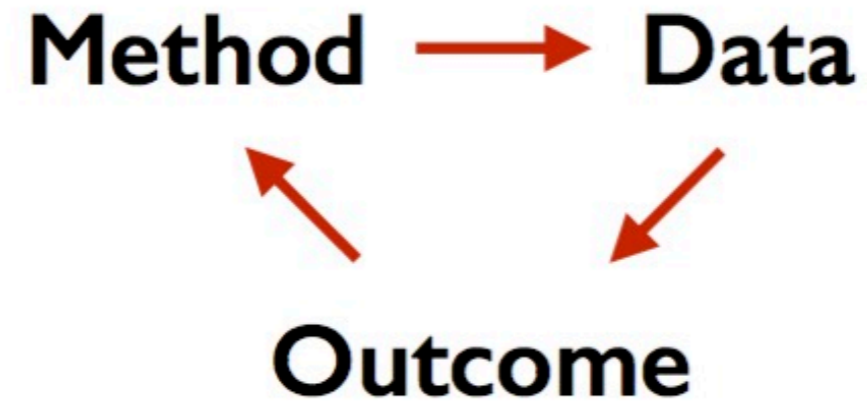
Adaptivity makes it more difficult to generalize to unseen data.

# Adaptivity

**Static**

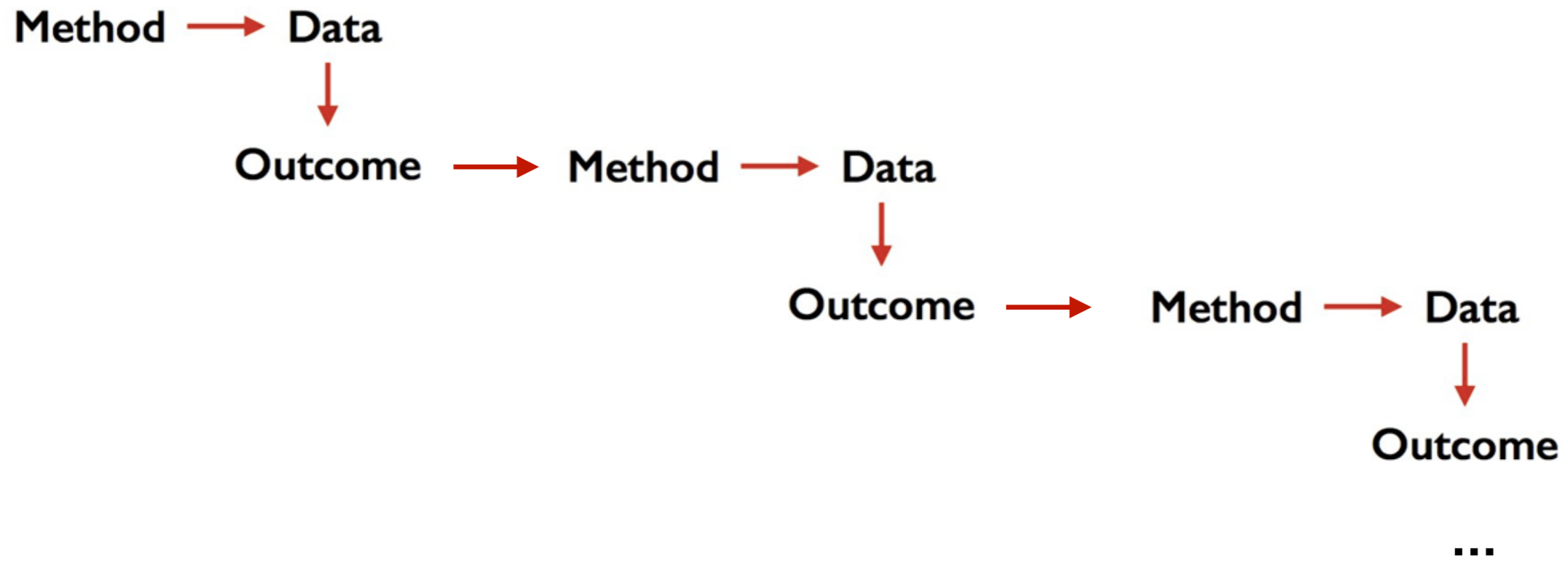


**Adaptive**



(Hardt '15)

# Not adaptivity



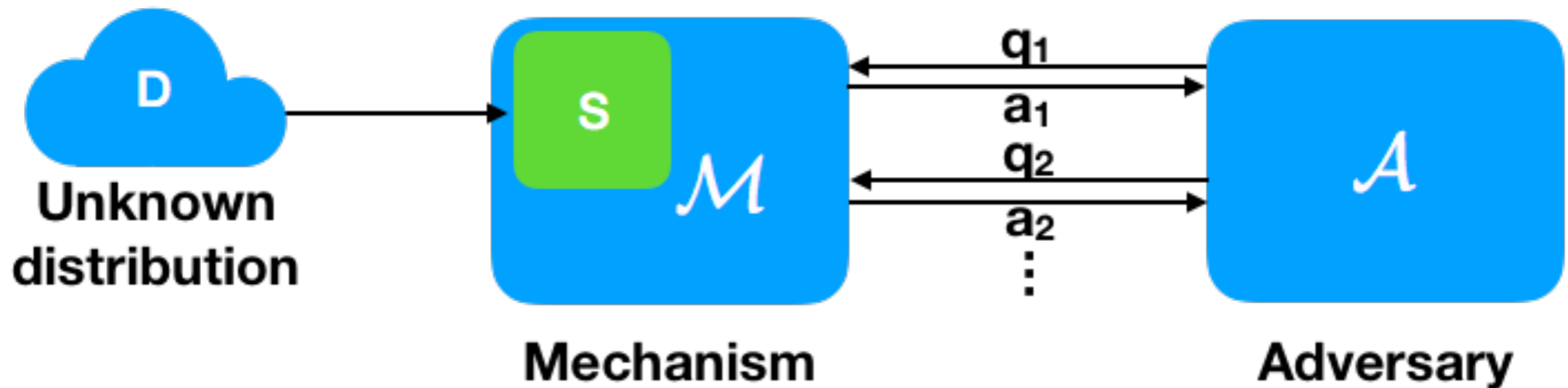
# The goal

We want a way to answer queries adaptively *on a fixed dataset* without having to assume:

1. anything about the nature of the adaptivity
2. that the queries come from a class of bounded complexity (e.g. bounded VC dimension)

# Adaptive data analysis

(Dwork et al. '15):



How many queries can we answer, and how long does it need to take?



# Queries

A *query* is just a function  $q : D \rightarrow \mathbb{R}$ , on which the mechanism wants to return a value close to  $q(D)$ .

# Queries

A *query* is just a function  $q : D \rightarrow \mathbb{R}$ , on which the mechanism wants to return a value close to  $q(D)$ .

A *statistical query* is specified by a function  $q : X \rightarrow [0, 1]$ . Then define:  $q(S) = \frac{1}{|S|} \sum_{x \in S} q(x)$

(Kearns '98)

# Statistical Queries, motivation

most methods can actually be implemented via SQs:

- gradient descent (Robbins, Monro '51)
- EM (Dempster, Laird, Rubin '77)
- SVM (Cortes, Vapnik 1995; Mitra, Murthy, Pal '04)
- boosting (Aslam, Decator '98)
- linear/convex optimization (Dunagan, Vempala '08)
- MCMC (Tanner, Wong '87; Gelfand, Smith '90)
- simulated annealing (Kirkpatrick, Gelatt, Vecchi '83; Cerny '85)

# Queries we consider

A *query* is just a function  $q : D \rightarrow \mathbb{R}$ , on which the mechanism wants to return a value close to  $q(D)$ .

A *counting query* is the special case of a statistical query that asks “What proportion of the data satisfies a given property?”

The property is specified by  $q : X \rightarrow \{0, 1\}$ , where

$$q(S) = \frac{1}{|S|} \sum_{x \in S} q(x)$$

(Dinur, Nissim '03; Yang '05)

# Queries we consider

A *query* is just a function  $q : D \rightarrow \mathbb{R}$ , on which the mechanism wants to return a value close to  $q(D)$ .

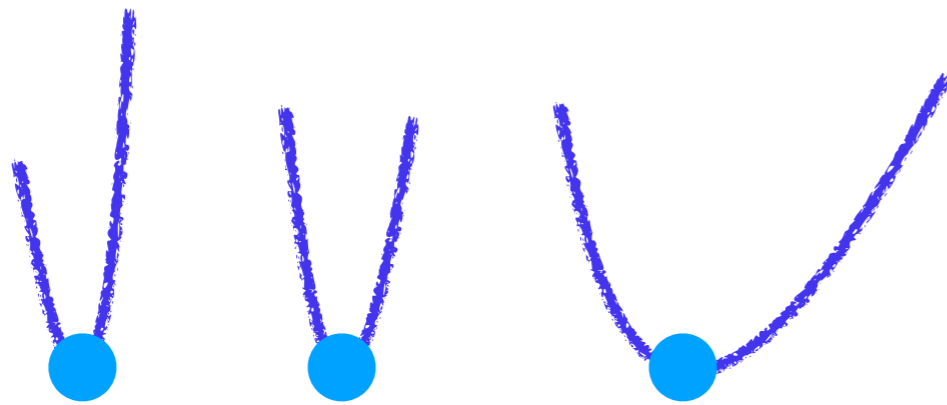
Given loss function  $\mathcal{L} : X^n \times \Theta \rightarrow \mathbb{R}$ , define an *optimization query* as  $q(D) := \arg \min_{\theta \in \Theta} E_{S \sim D^n} [\mathcal{L}(S, \theta)]$ .

(Bassily et al. '16)

# Queries we consider

A *query* is just a function  $q : D \rightarrow \mathbb{R}$ , on which the mechanism wants to return a value close to  $q(D)$ .

Given loss function  $\mathcal{L} : X^n \times \Theta \rightarrow \mathbb{R}$ , define an *optimization query* as  $q(D) := \arg \min_{\theta \in \Theta} E_{S \sim D^n} [\mathcal{L}(S, \theta)]$ .



(Bassily et al. '16)

# Other queries

A *query* is just a function  $q : D \rightarrow \mathbb{R}$ , on which the mechanism wants to return a value close to  $q(D)$ .

A *low sensitivity query* is specified by a function  $q : X^n \rightarrow \mathbb{R}$  where for all samples  $S, S' \in X^n$  that differ on only one element,  $|q(S) - q(S')| \leq 1/n$ .

Then define  $q(D) := E_{S \sim D^n} [q(S)]$

(Dwork et al. '06)

# Accuracy

(Dwork et al. '15)

A mechanism  $M$  is  $(\alpha, \beta)$ -accurate on the distribution  $D$  if for all queries  $q_i$ ,

$$P_{\mathcal{M}, \mathcal{A}}[\max_i |q_i(D) - a_i| \leq \alpha] \geq 1 - \beta$$



# Accuracy

(Dwork et al. '15)

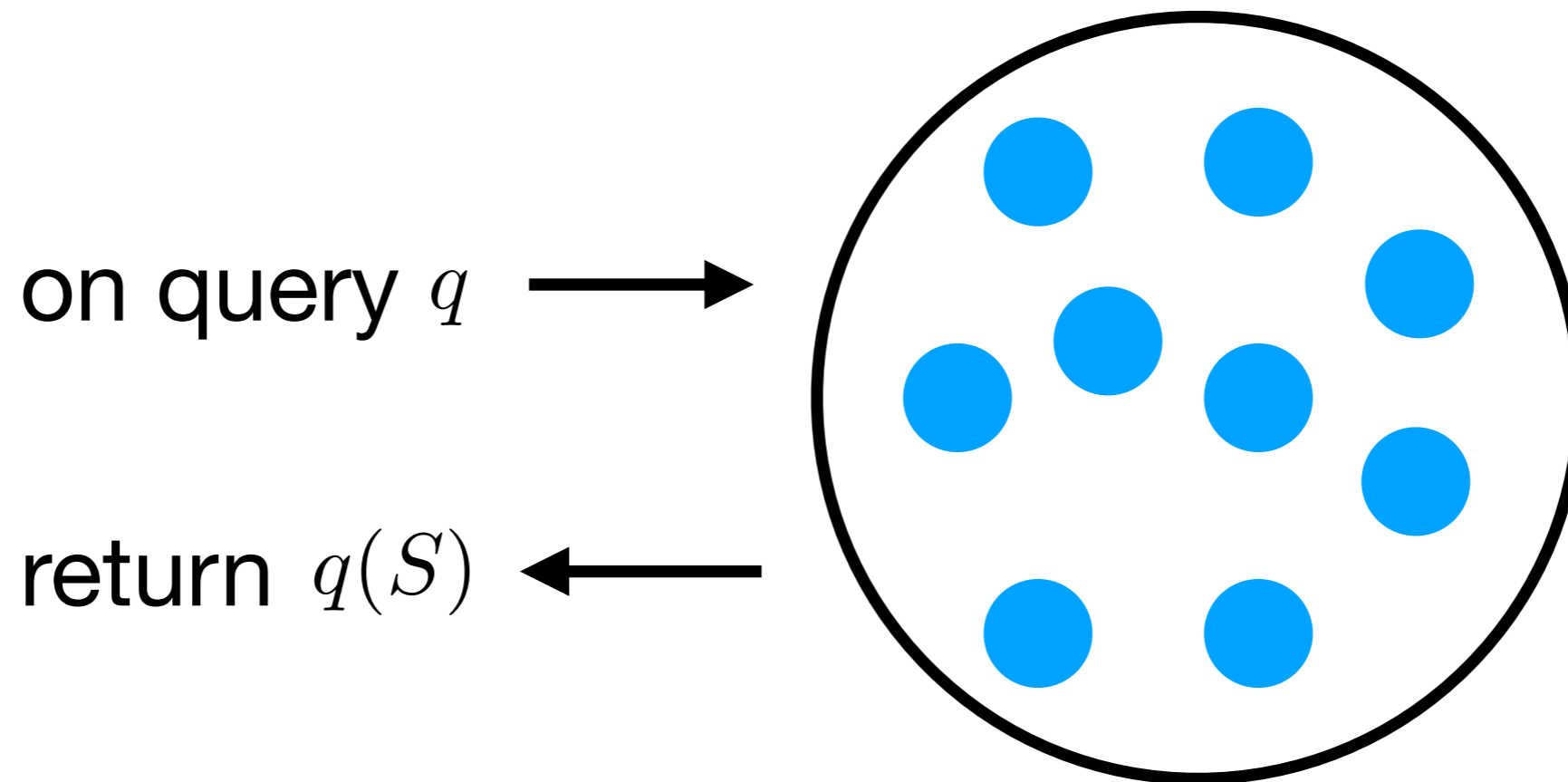
A mechanism  $M$  is  $(\alpha, \beta)$ -accurate on the distribution  $D$  if for all queries  $q_i$ ,

$$P_{\mathcal{M}, \mathcal{A}}[\max_i |q_i(D) - a_i| \leq \alpha] \geq 1 - \beta$$

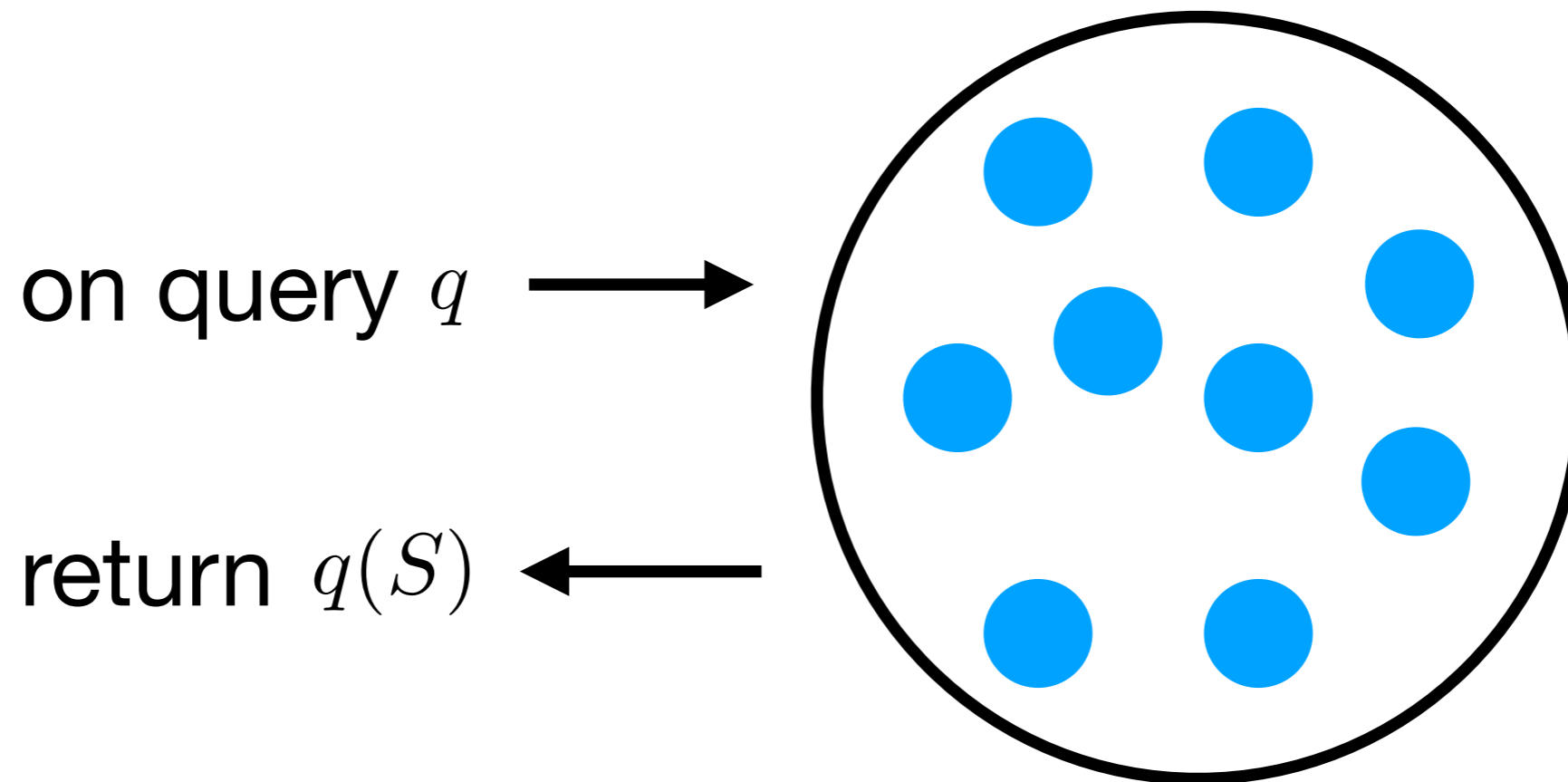
How many samples  $n$  does it take to answer  $k$  adaptive queries efficiently with  $(\alpha, \beta)$ -accuracy?

And how fast can we answer such queries?

# PAC method



# PAC method



*Returning the empirical estimate turns out to be suboptimal!*

# Taking advantage of adaptivity in leaderboards

The *leaderboard*: Takes a hold-out set of  $n$  points with labels  $y \in \{0, 1\}^n$  and for any label prediction  $u \in \{0, 1\}^n$  return the average 0-1 loss  $\mathcal{L}(u)$ .

# Taking advantage of adaptivity in leaderboards

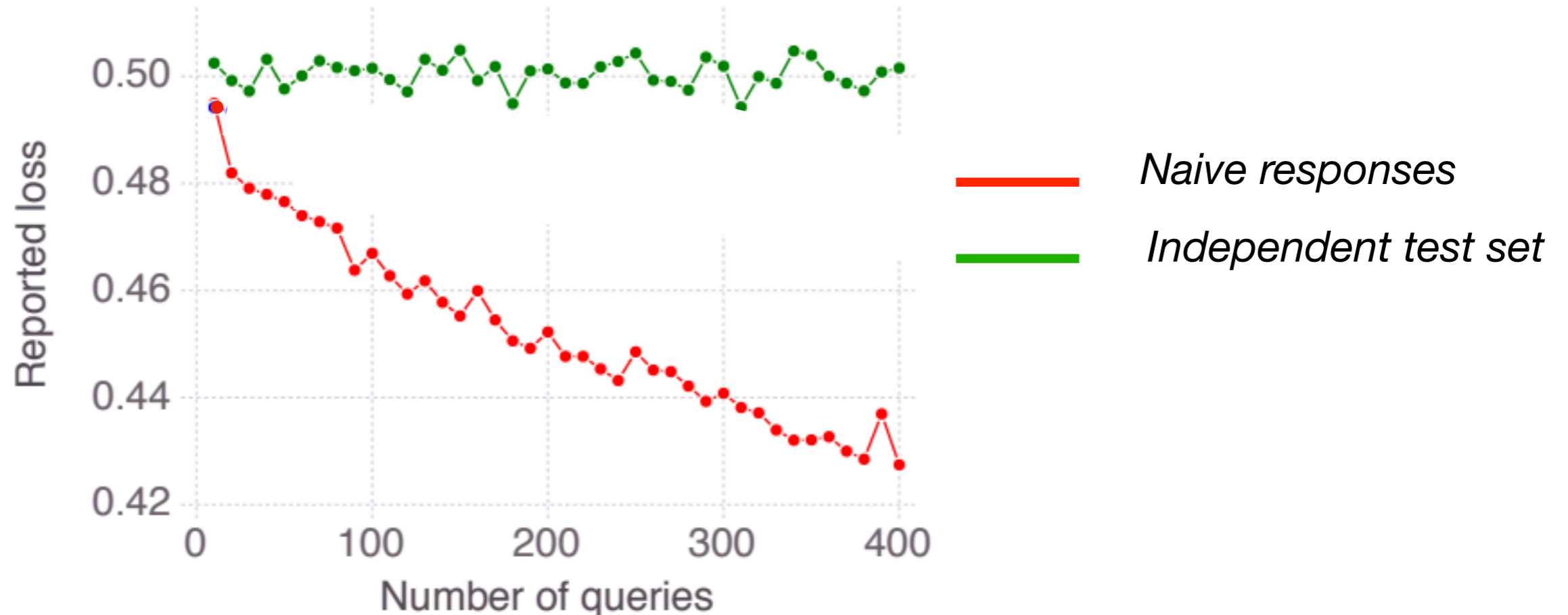
The *leaderboard*: Takes a hold-out set of  $n$  points with labels  $y \in \{0, 1\}^n$  and for any label prediction  $u \in \{0, 1\}^n$  return the average 0-1 loss  $\mathcal{L}(u)$ .

The boosting attack (Blum, Hardt '15):

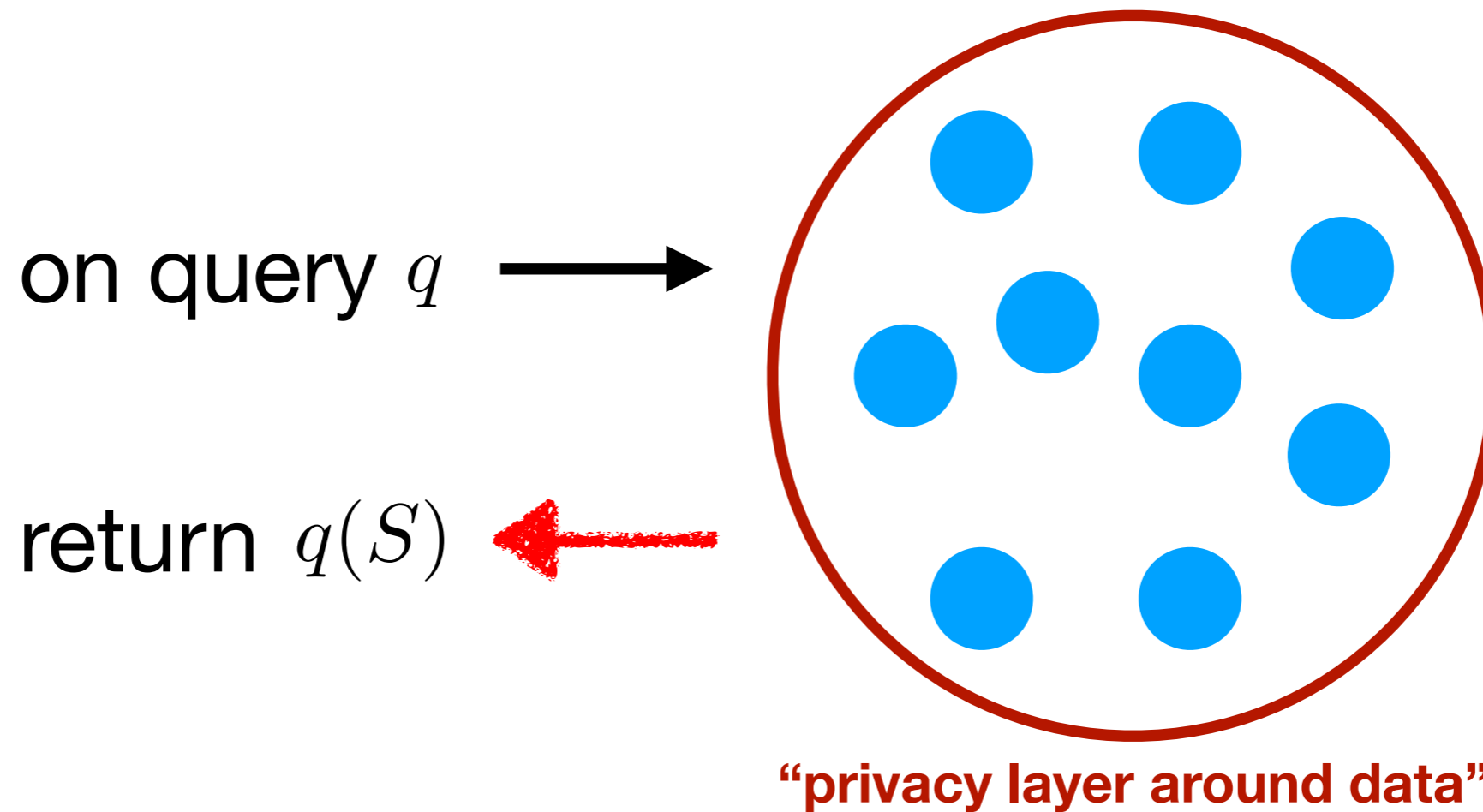
- Pick  $k$  vectors  $u_1, \dots, u_k$  uniformly at random, and receive losses  $\mathcal{L}_1, \dots, \mathcal{L}_k$  in response. Call  $I = \{i : \mathcal{L}_i \leq 1/2\}$ .
- Output  $u^* = \text{maj}(\{u_i : i \in I\})$ , applied coordinate-wise.

# Taking advantage of adaptivity in leaderboards

**Theorem** (Blum and Hardt 2015). *With at least constant probability, the loss is  $\mathcal{L}(u) \leq 1/2 - \Omega\left(\sqrt{k/n}\right)$ .*



# An idea



***“An algorithm that cannot learn the data, will not overfit!”***

# Transfer

Better yet, *stable algorithms*, algorithms whose output only change by a little when the inputs change by a little, can be made to generalize

I'll refer to such a result as a *transfer theorem*.



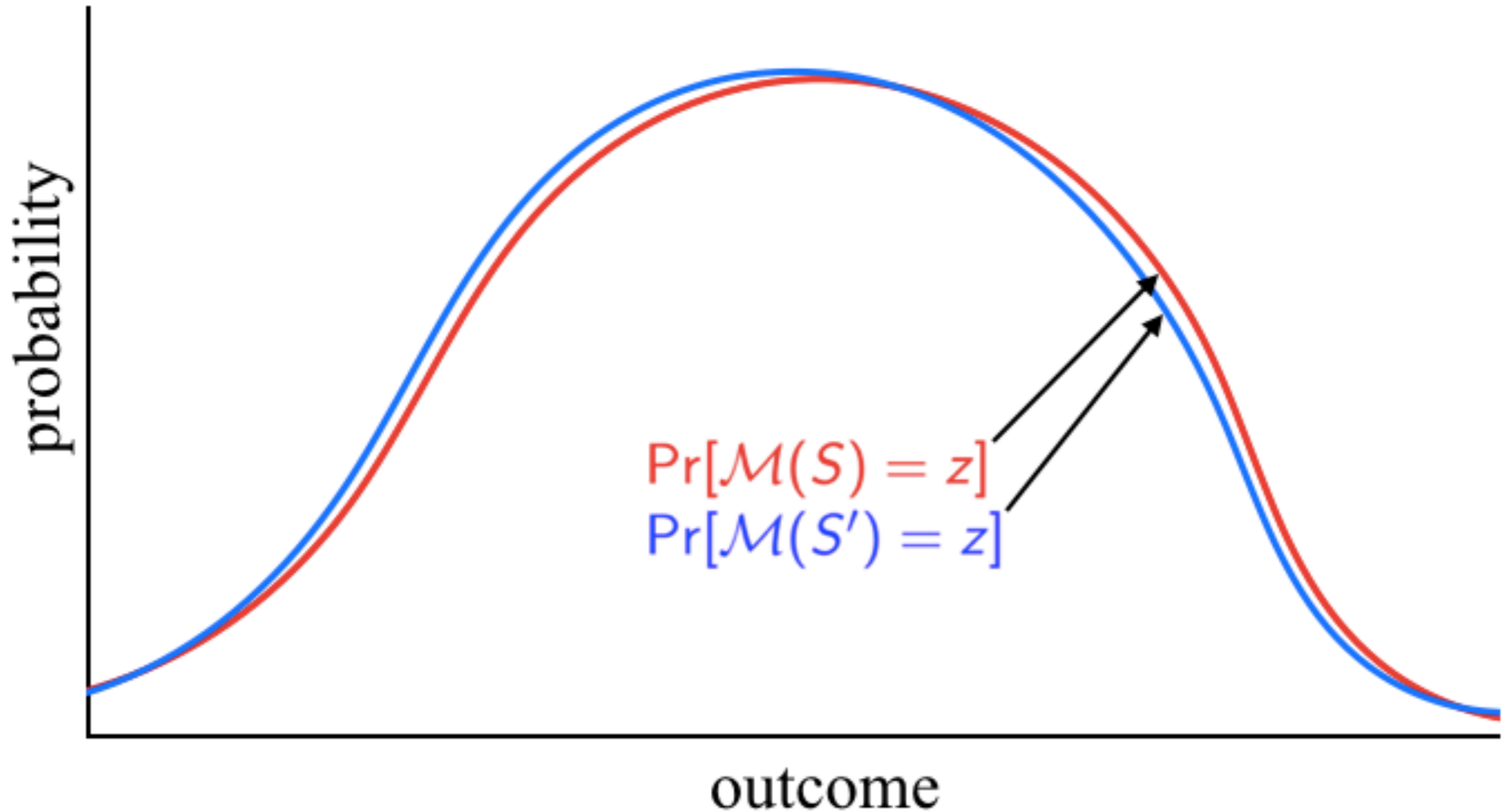
# Differential privacy

The most popular/important definition of stability is differential privacy:

**Definition** (Dwork et al. 2006). *A mechanism  $\mathcal{M}$  is  $(\epsilon, \delta)$ -private if for every two samples  $S, S' \in X^n$  differing by at most one element and every outcome  $z$ ,*

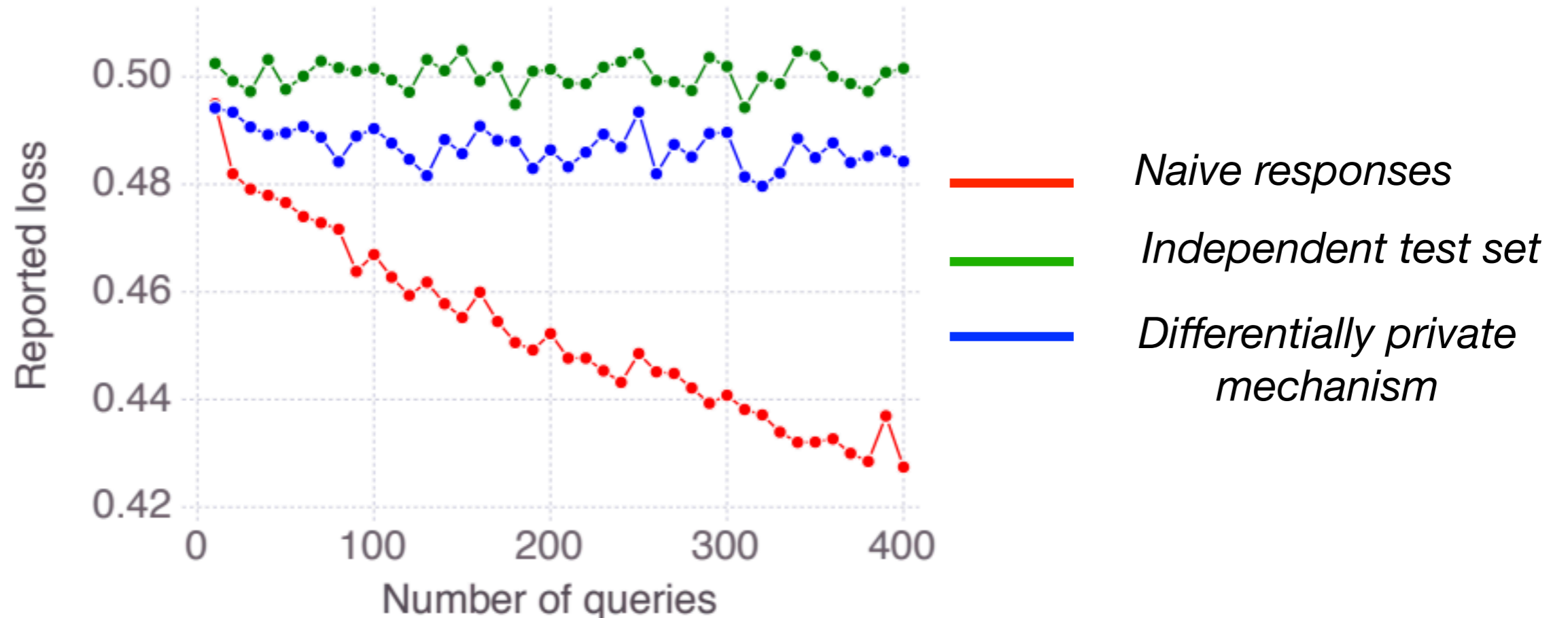
$$P[\mathcal{M}(S) = z] \leq e^\epsilon \cdot P[\mathcal{M}(S') = z] + \delta$$

# Differential privacy illustration



# Differentially private responses in leaderboards

**Theorem** (Blum and Hardt 2015). *With at least constant probability, the loss is  $\mathcal{L}(u) \leq 1/2 - \Omega\left(\sqrt{k/n}\right)$ .*



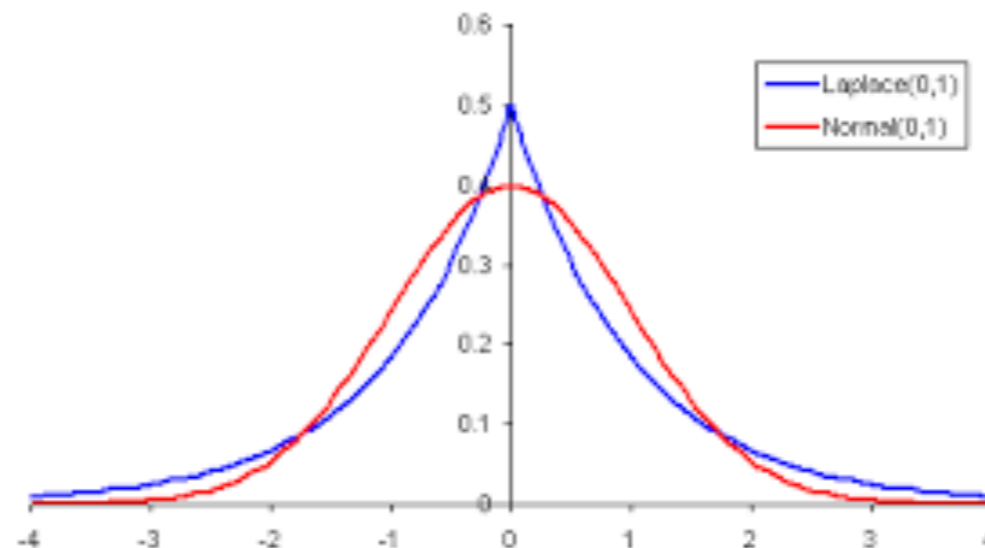
# A mechanism for SQs

(Dwork et al. '15, Bassily et al. '16)

Given a data set  $S$  of size  $n$  and a query  $q$ ,  $\mathcal{M}$  will

1. Return  $q(S) + \text{Lap}\left(\frac{1}{n\epsilon}\right)$  (Laplacian noise)

$$\text{Lap}(z|b) = \frac{1}{2b} \exp\left(-\frac{|z|}{b}\right)$$



# A mechanism for SQs

(Dwork et al. '15, Bassily et al. '16)

Given a data set  $S$  of size  $n$  and a query  $q$ ,  $\mathcal{M}$  will

1. Return  $q(S) + \text{Lap}\left(\frac{1}{n\epsilon}\right)$  (Laplacian noise)
- Then this mechanism is: on the one hand, accurate on the sample and, on the other hand private.
- This is sufficient to guarantee accuracy on the distribution (*transfer*)

# Known results (sample complexity)

- Low sensitivity + statistical queries:  $n = \tilde{O}\left(\sqrt{k}/\alpha^2\right)$   
(Bassily et al. '16) and this bound is tight in  $k$   
(Steinke, Ullman '15)
- Convex optimization in  $d$  dimensions:  $\tilde{O}\left(\sqrt{kd}/\alpha^2\right)$   
(Bassily et al. '16)
- Bounded max-information is sufficient, which enables hypothesis testing, etc. (Dwork et al. '15, Rogers et al. '16)

# Computational complexity

- All of these mechanisms take  $\Omega(n)$  time per query, not realistic for large-scale analysis
- Simultaneously, there is a large gap between theory and practice, e.g. sampling practices like bootstrapping

# Summary of our results

		<i>This work</i>	<i>Bassily et al. 2016</i>
<i>Samples per query</i>	statistical queries	$\tilde{O}(\log(k)/\alpha^2)$	$\tilde{O}(\sqrt{k}/\alpha^2)$
	sampling counting queries	$\tilde{O}(\log(k)/\alpha^2)$	N/A
<i>Iterations per query</i>	convex optimization	$\tilde{O}(\log(k)/\alpha^2)$	$\tilde{O}(dk/\alpha^4)$
	strongly convex optimization	$\tilde{O}(\log(k)/\alpha)$	$\tilde{O}(dk/\alpha^3)$

dependence on  $\beta$  suppressed for convenience.

recall:  $k$  is the number of queries,  $\alpha$  the accuracy rate



# SQs

**Theorem:** There is a mechanism for answering statistical queries with

- *sample complexity*  $n = \tilde{O}\left(\frac{\sqrt{k}}{\alpha^2}\right)$
- $\ell = \tilde{O}\left(\frac{\log(k)}{\alpha^2}\right)$  *samples per query*
- $\tilde{O}\left(\frac{\log^2(k)}{\alpha^2}\right)$  *time per query.*

# Fast mechanism for SQs

Given a data set  $S$  of size  $n$  and a query  $q$ ,  $\mathcal{M}$  will

1. Sample  $\ell$  points uniformly at random (with or without replacement) and call this sample  $S_\ell$
  2. Return  $q(S_\ell) + \text{Lap}\left(\frac{1}{\ell\epsilon}\right)$  (Laplacian noise)
- This corresponds to bootstrapping: every new query we re-subsample
  - $\mathcal{M}(S, q) \approx q(S_\ell) \approx q(S) \approx q(D)$

# Why does this work?

Proof idea: While subsampling means we have a worse estimator, sampling also boosts the amount of privacy we have.

We can prove a transfer theorem where the two exactly cancel out!

Useful lemma:

**Proposition (Bun et al., 2015; Balle et al., 2018)** *Given mechanism  $\mathcal{P} : X^\ell \rightarrow Y$ , let  $\mathcal{M}$  do the following: Sample uniformly at random with replacement  $\ell$  points from an input sample  $S \in X^n$ , and call this set  $S_\ell$ . Output  $\mathcal{P}(S_\ell)$ . Then if  $\mathcal{P}$  is  $\epsilon$ -private, then  $\mathcal{M}$  is  $\log(1 + (1 - (1 - \frac{1}{n})^\ell)(e^\epsilon - 1))$ -private for  $\ell \geq 1$ .*

# Applications to convex optimization

Recall: given loss function  $\mathcal{L} : X^n \times \Theta \rightarrow \mathbb{R}$ , define optimization query  $q(D) := \arg \min_{\theta \in \Theta} E_{S \sim D^n} [\mathcal{L}(S, \theta)]$

We need a few conditions:

- $\mathcal{L}$  differentiable and convex (or strongly convex)
- $\Theta$  convex
- $\nabla \mathcal{L}$  is statistical
- for any  $S' \subset S$  and  $x \in \Theta$ ,  $E[\|\nabla \mathcal{L}(S', x)\|^2] \leq G^2$
- $\Theta$  has bounded diameter (needed for non-strongly convex optimization)

# Applications to convex optimization

**Theorem** (Convex optimization). *There is such a mechanism with*

- $n = \tilde{O}\left(\frac{d^{3/2} \sqrt{k} \log(k/\beta)}{\alpha^5}\right)$  *sample complexity*
- $\tilde{O}\left(\frac{d^2 \log(k/\beta)}{\alpha^5}\right)$  *samples per query*
- $\tilde{O}\left(\frac{\log(k/\beta)}{\alpha^2}\right)$  *iterations of gradient descent per query.*

# Applications to convex optimization

**Theorem** (Strongly convex optimization). *There is such a mechanism with*

- $n = \tilde{O}\left(\frac{d^{3/2}\sqrt{k}\log(k/\beta)}{\alpha^{5/2}}\right)$  *sample complexity*
- $\tilde{O}\left(\frac{d^2\log(k/\beta)}{\alpha^3}\right)$  *samples per query*
- $\tilde{O}\left(\frac{\log(k/\beta)}{\alpha}\right)$  *iterations of gradient descent per query.*

# (Stochastic Block) Gradient descent with a black box

1. Pick arbitrary  $x_0 \in \Theta$
2. Repeat  $x_t := x_{t-1} - \eta \tilde{\nabla} \mathcal{L}(S, x_{t-1})$   
where each component of  $\tilde{\nabla} \mathcal{L}(S, x_{t-1})$  is given by our mechanism  $\mathcal{M}$  for statistical queries:

$$\tilde{\nabla} \mathcal{L}(S, x_{t-1})^{(i)} := \mathcal{M}(\nabla \mathcal{L}(S, x_{t-1})^{(i)}, S)$$

# (Stochastic Block) Gradient descent with a black box

1. Pick arbitrary  $x_0 \in \Theta$
2. Repeat  $x_t := x_{t-1} - \eta \tilde{\nabla} \mathcal{L}(S, x_{t-1})$   
where each component of  $\tilde{\nabla} \mathcal{L}(S, x_{t-1})$  is given by our mechanism  $\mathcal{M}$  for statistical queries:

$$\tilde{\nabla} \mathcal{L}(S, x_{t-1})^{(i)} := \mathcal{M}(\nabla \mathcal{L}(S, x_{t-1})^{(i)}, S)$$

Proof idea: Make  $\mathcal{M}$  sufficiently accurate so that the cumulative difference in the estimated gradients and the actual gradients is sufficiently small in the worst case.



# Sampling counting queries

For a counting query  $q : X \rightarrow \{0, 1\}$ ,

- Before: want  $\mathcal{M}$  to return a value close to  $E_{x \sim D}[q(x)]$

# Sampling counting queries

For a counting query  $q : X \rightarrow \{0, 1\}$ ,

- Before: want  $\mathcal{M}$  to return a value close to  $E_{x \sim D}[q(x)]$
- Now:  $\mathcal{M}$  should return  $q(x)$  itself, for  $x \sim D$
- A *sampling counting query* is again specified by a property  $q : X \rightarrow \{0, 1\}$  but now  $\mathcal{M}$  must return an answer  $a \in \{0, 1\}$

# Sampling counting queries

**Definition.** A mechanism  $\mathcal{M}$  is  $(\alpha, \beta)$ -accurate on distribution  $D$  for  $k$  sampling counting queries  $q_i$  if for all states of  $\mathcal{M}$  and  $\mathcal{A}$ , when  $\mathcal{M}$  is given an i.i.d. sample  $S$  from  $D$ ,

$$P_{S, \mathcal{M}, \mathcal{A}} \left[ \max_i |E_{\mathcal{M}}[\mathcal{M}(q_i)] - q_i(D)| \leq \alpha \right] \geq 1 - \beta.$$

# SCQ results

**Theorem** (Transfer theorem for SCQs). *Let  $\mathcal{M}$  be a mechanism that on input sample  $S \sim D^n$  answers  $k$  adaptively chosen sampling counting queries, is  $(\frac{\alpha}{64}, \frac{\alpha\beta}{16})$ -private for some  $\alpha, \beta > 0$  and  $(\alpha/2, 0)$ -accurate on  $S$ . Suppose further that  $n \geq \frac{1024 \log(k/\beta)}{\alpha^2}$ . Then  $\mathcal{M}$  is  $(\alpha, \beta)$ -accurate on  $D$ .*

# SCQ results

**Theorem.** *There is an  $(\alpha, \beta)$ -accurate mechanism for answering  $k$  sampling counting queries with*

- *sample complexity  $n = O\left(\frac{\sqrt{k} \log(\frac{1}{\alpha\beta})}{\alpha^2}\right)$*
- *$\ell = 1$  sample per query*
- *$\tilde{O}\left(\log\left(\frac{k \log(\frac{1}{\beta})}{\alpha}\right)\right)$  time per query.*

# Conclusions

- Faster data analysis is not only possible, but can be implemented via simple mechanisms, without downsides!
- Applications to adaptive convex optimization.
- Even with a constant number of samples, it is still possible to produced meaningful mechanisms.