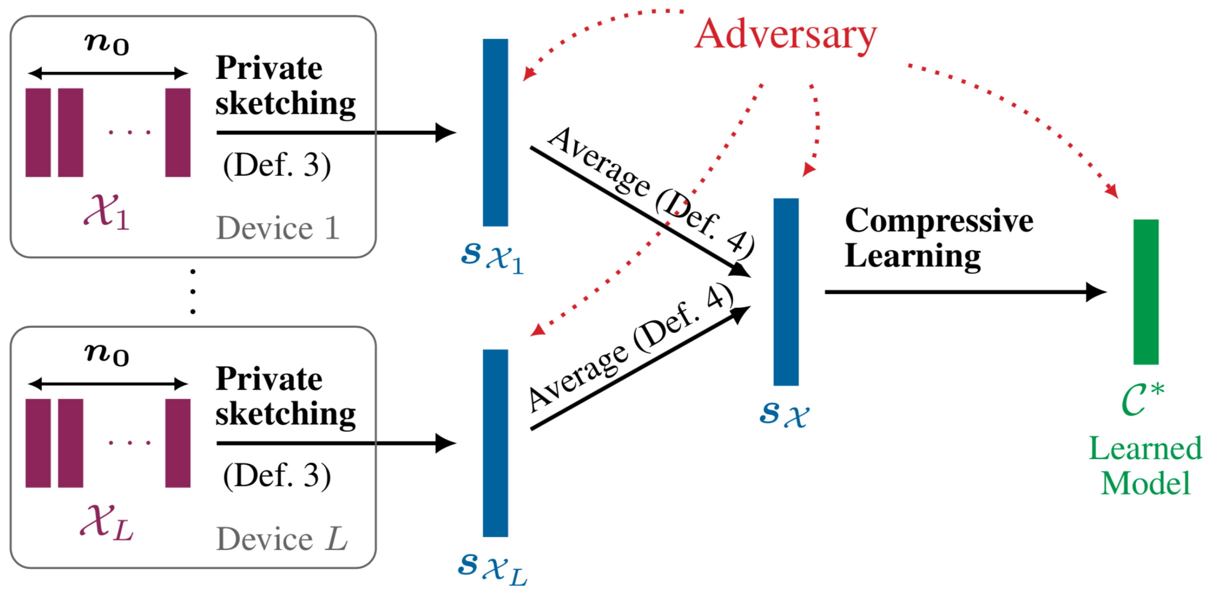
Differentially Private Compressive K-Means



Florimond Houssiau
Yves-Alexandre de Montjoye
Imperial College London

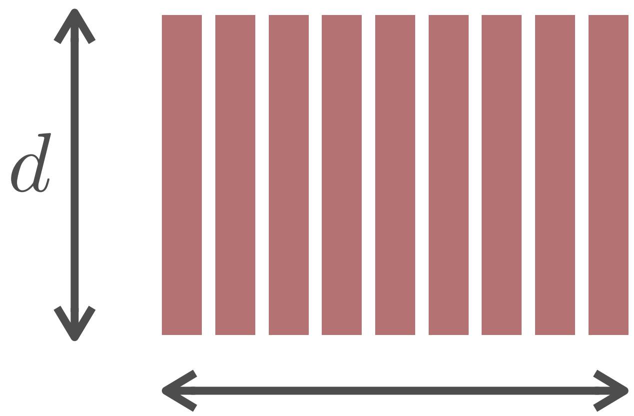
Vincent Schellekens
Laurent Jacques
UCLouvain

Antoine Chatalic
Rémi Gribonval
Inria Rennes

Context: large-scale machine learning

Dataset $\mathcal{X}$

$\mathcal{X} = \{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$

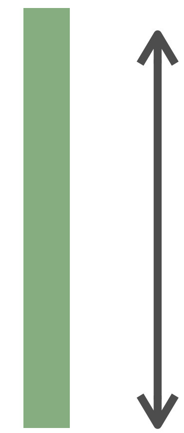


n examples

very large ("large-scale"!)

Machine Learning

Parametric model θ



p parameters

small ("sparsity")

e.g.

- Gaussian Mixture Model
- Principal Component Analysis
- *k*-means clustering

k-means clustering

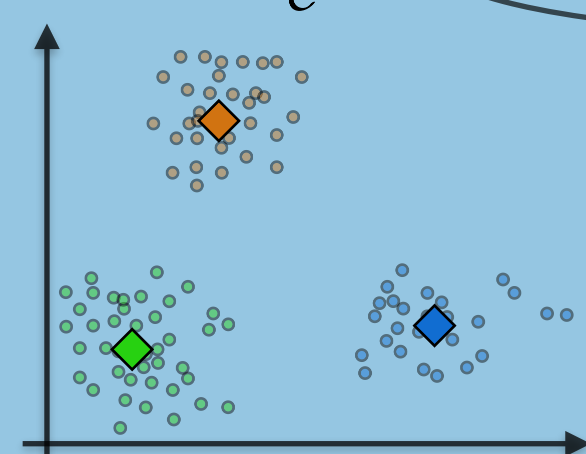
$$\theta = \arg\min_{\mathcal{C}} \mathrm{SSE}_{\mathcal{X}}(\mathcal{C})$$

"centroids" $p = dk$

$\mathcal{C} = \{\boldsymbol{c}_j\}_{j=1}^k$

"Sum of Squared Errors"

$$\mathrm{SSE}_{\mathcal{X}}(\mathcal{C}) := \sum_{\boldsymbol{x}_i \in \mathcal{X}} \min_{\boldsymbol{c}_j \in \mathcal{C}} \|\boldsymbol{x}_i - \boldsymbol{c}_j\|_2^2$$

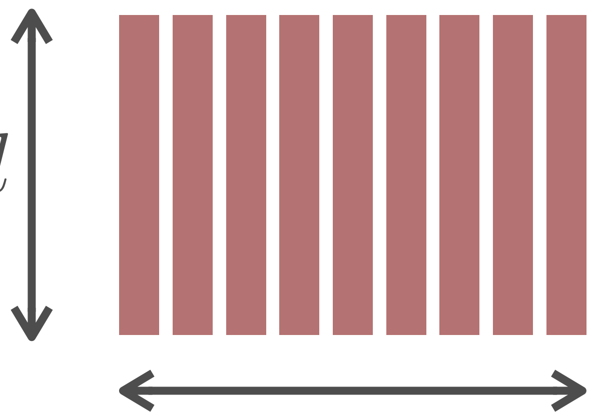


Context: large-scale machine learning

Dataset $\mathcal{X}$

$\mathcal{X} = \{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$

d



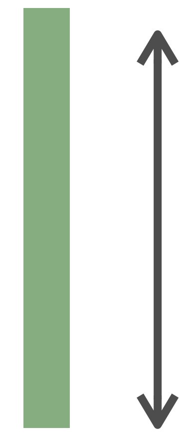
n examples

very large ("large-scale"!)

Machine Learning

Challenges?

Parametric model θ



p parameters

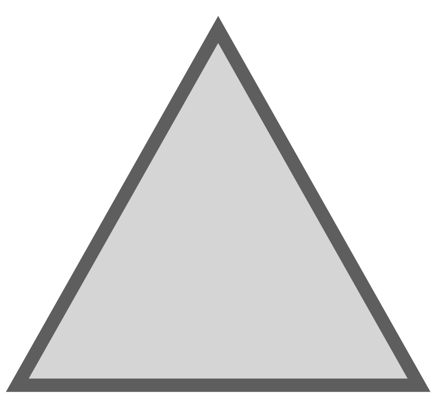
small ("sparsity")

e.g.

- Gaussian Mixture Model
- Principal Component Analysis
- *k*-means clustering

Accuracy

= good precision for θ



Efficiency

= low memory & time reqs.

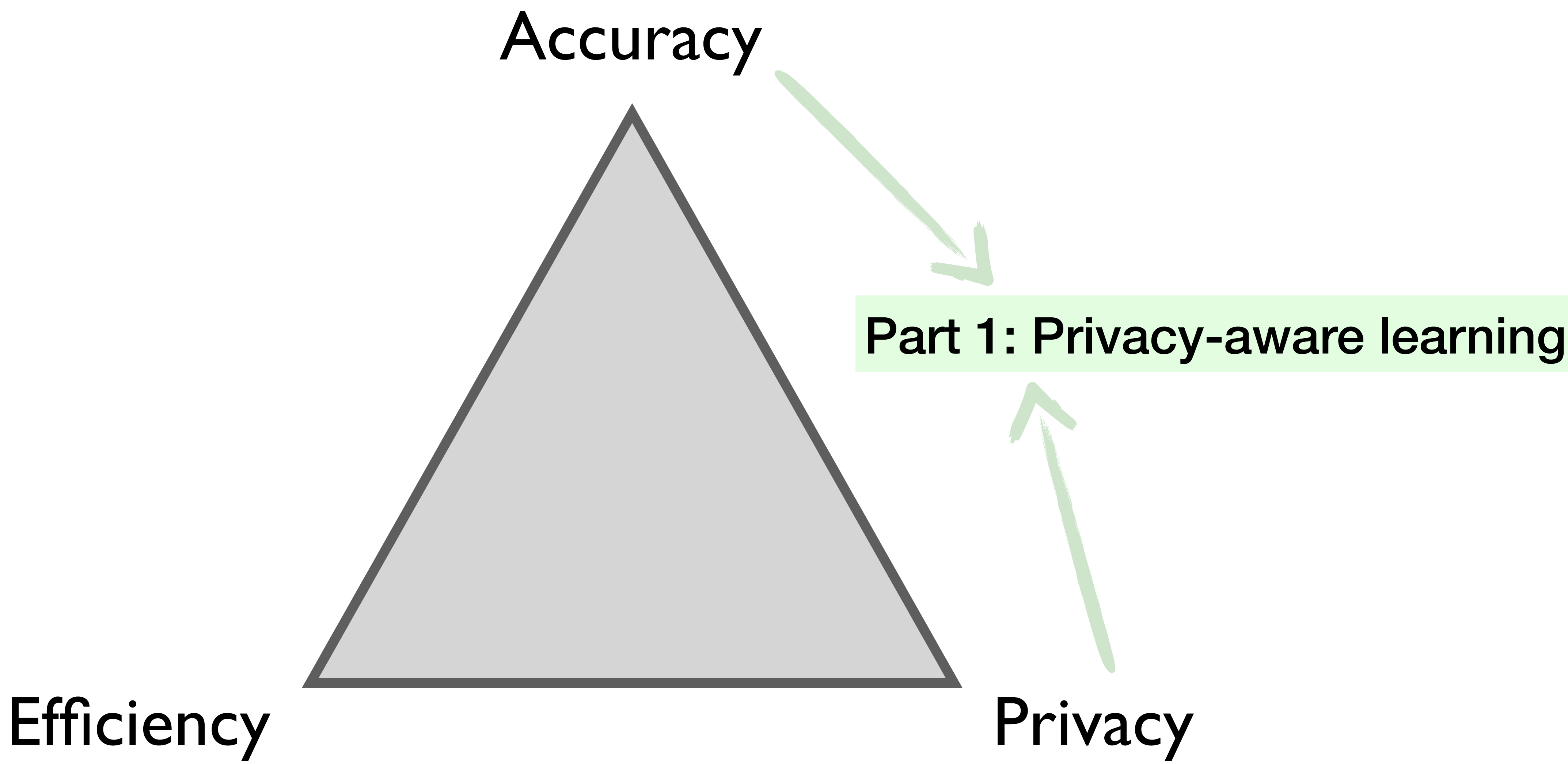
Privacy

= protect sensitive data

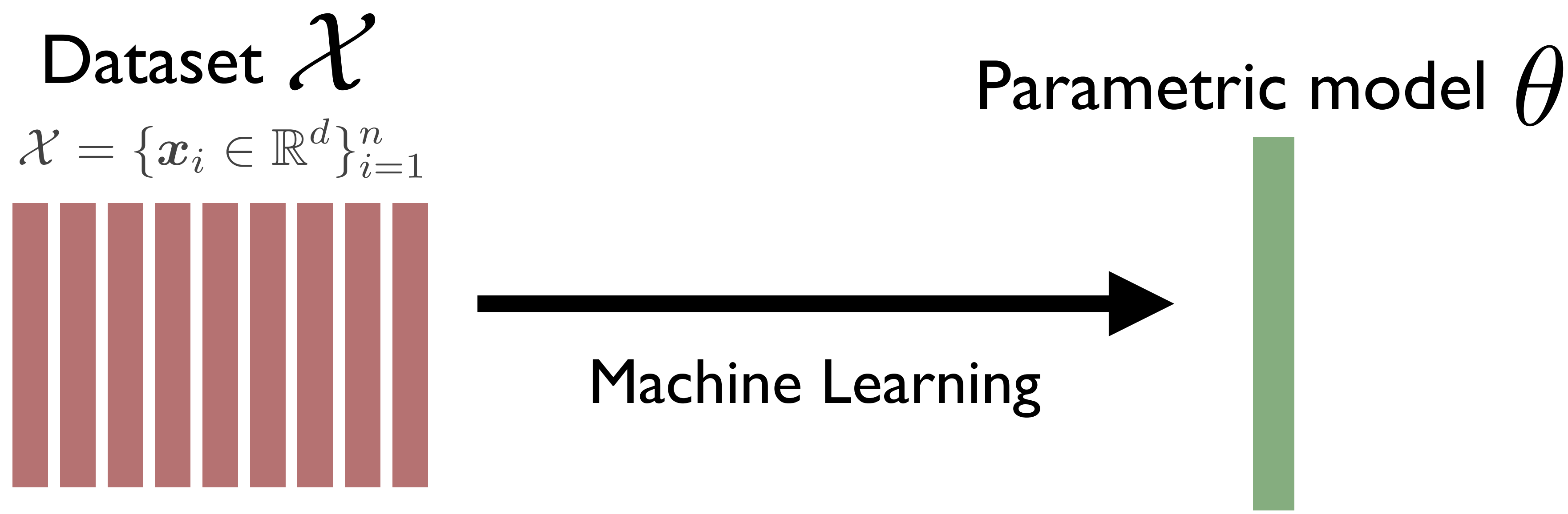
!

Several objectives that are **incompatible**!

Some preliminaries (1)



What IS privacy (in ML)



"Sensitive" information!

e.g.,

. . .

Goal: learn (unsupervised) from dataset while protecting its "privacy"!

Ok, but what does it mean?

Privacy is very difficult to define (a research topic on its own)!

Depends on the application (what do we want to protect), and the *attack model* (what do we want to protect against).

Many mathematical privacy definitions, with different pro/cons:

- k-Anonymity
- Information-theoretic privacy definitions
- Differential Privacy ← This work
- …

Differential Privacy: (a possible) definition

Intuitive definition: plausible deniability

> "An algorithm is Differentially Private if its output is not much influenced when one user of the dataset is changed"

"It is not possible to detect with high confidence whether I participated to the dataset or not"

C. Dwork, "*Differential Privacy*," 2006

Differential Privacy: (a possible) definition

Intuitive definition: plausible deniability

> "An algorithm is Differentially Private if its output is not much influenced when one user of the dataset is changed"

"It is not possible to detect with high confidence whether I participated to the dataset or not"

Formal definition: a *randomized* algorithm f is Differentially Private if

$$\epsilon - \mathrm{DP}$$

f satisfies $\epsilon - \mathrm{DP}$ if: $\begin{cases} \forall S \\ \forall X \sim X' \end{cases}$

$$\mathbb{P}[f(X) \in S] \leq e^{\epsilon} \cdot \mathbb{P}[f(X') \in S]$$

C. Dwork, "*Differential Privacy*," 2006

Differential Privacy: (a possible) definition

Intuitive definition: plausible deniability

> "An algorithm is Differentially Private if its output is not much influenced when one user of the dataset is changed"

"It is not possible to detect with high confidence whether I participated to the dataset or not"

Formal definition: a *randomized* algorithm f is Differentially Private if

Taken over the randomness in f

$\epsilon - \mathrm{DP}$

f satisfies $\epsilon - \mathrm{DP}$ if: $\begin{cases} \forall S \\ \forall X \sim X' \end{cases}$

$$\mathbb{P}[f(X) \in S] \leq e^{\epsilon} \cdot \mathbb{P}[f(X') \in S]$$

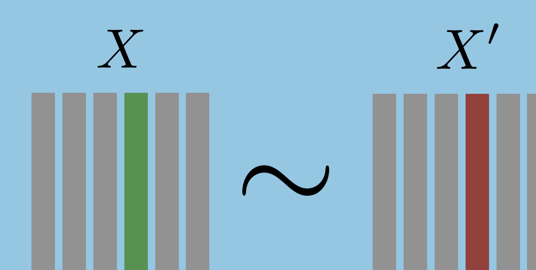
For all subsets of possible outcomes

For all "neighbour" DS

Neighbouring relation $\sim$

$X \sim X'$ if they differ by one entry

i.e. $|X| = |X'|$ and $|(X \cup X') \setminus (X \cap X')| \leq 2$



C. Dwork, "*Differential Privacy*," 2006

Differential Privacy: (a possible) definition

Intuitive definition: plausible deniability

> "An algorithm is Differentially Private if its output is not much influenced when one user of the dataset is changed"

"It is not possible to detect with high confidence whether I participated to the dataset or not"

Formal definition: a *randomized* algorithm f is Differentially Private if

$\epsilon - \mathrm{DP}$

f satisfies $\epsilon - \mathrm{DP}$ if: $\begin{cases} \forall S \\ \forall X \sim X' \end{cases}$

$$\mathbb{P}[f(X) \in S] \leq e^{\epsilon} \cdot \mathbb{P}[f(X') \in S]$$

Taken over the randomness in f

For all subsets of possible outcomes

For all "neighbour" DS

<u>Privacy parameter</u>/budget

! should be small

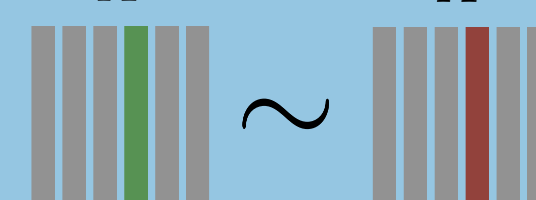
$$\mathbb{P}[f(X) \in S] \simeq \mathbb{P}[f(X') \in S] + \mathcal{O}(\epsilon)$$

Neighbouring relation $\sim$

$X \sim X'$ if they differ by one entry

i.e. $|X| = |X'|$ and $|(X \cup X') \setminus (X \cap X')| \leq 2$

X $\qquad$ X'



C. Dwork, "*Differential Privacy*," 2006

Differential Privacy: pros/cons

- Extensively studied, widely accepted standard (2008-present)
- Very strong (robust to most attacks, side-information, post-processing…)
- Often easy to implement (Laplacian mechanism, see later)

- "Too strong" (restrictive) guarantee?
- How to pick ϵ ?

Differential Privacy: pros/cons

- Extensively studied, widely accepted standard (2008-present)
- Very strong (robust to most attacks, side-information, post-processing…)
- Often easy to implement (Laplacian mechanism, see later)

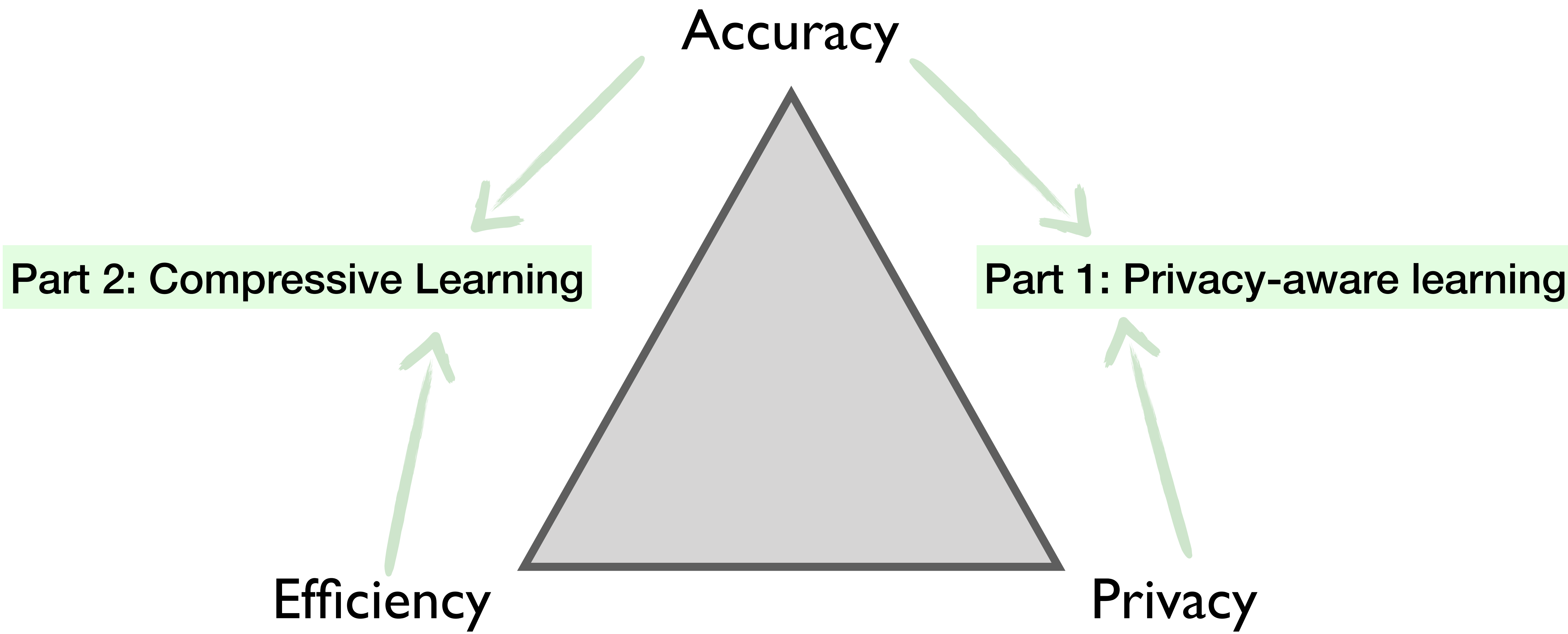
- "Too strong" (restrictive) guarantee?
- How to pick ϵ ?
  - Hard to interpret
  - Should consider "privacy-utility" tradeoff:

"Good" privacy
Bad accuracy

No privacy
Good accuracy

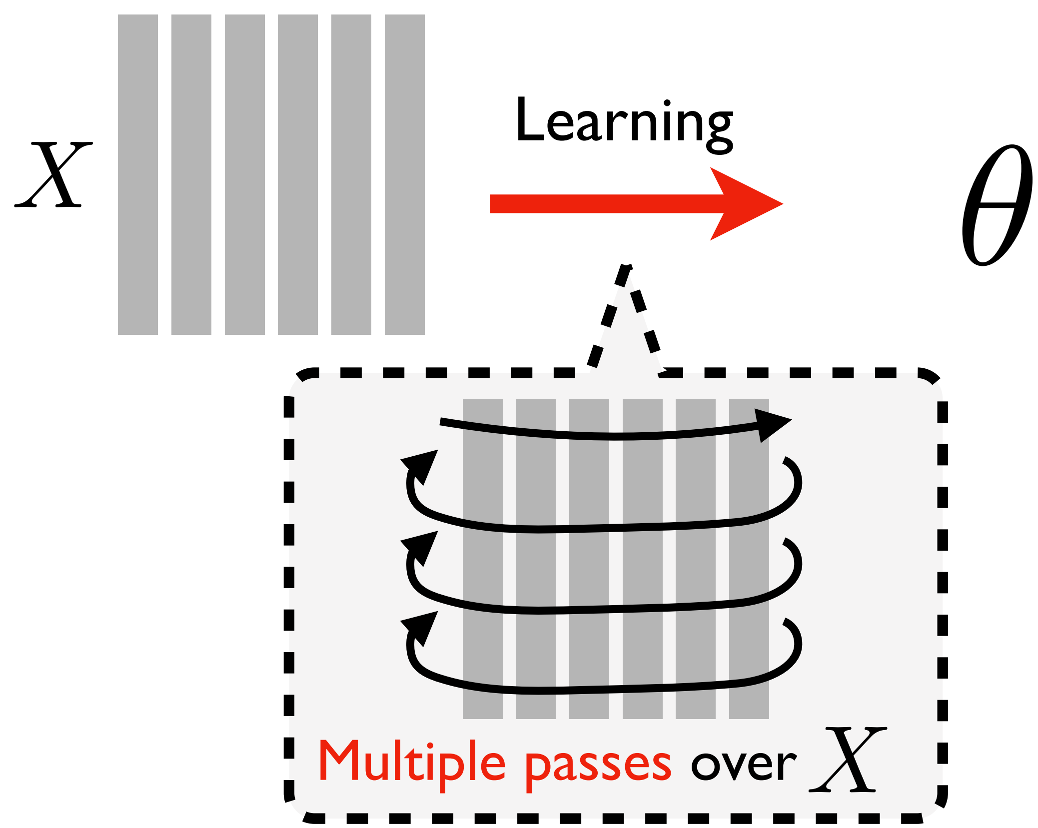
  - Heavily context-dependent, requires "expert knowledge"!
  - Typical values: $\epsilon \simeq 10^{-2} \cdots 10^{-1}$ …to take with a grain of salt!

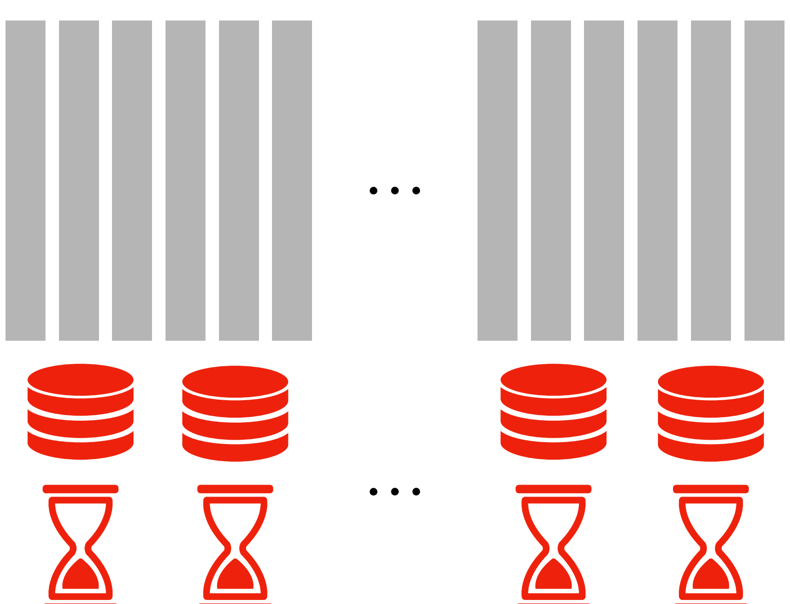
Some preliminaries (2)



Compressive Learning

Usual Learning

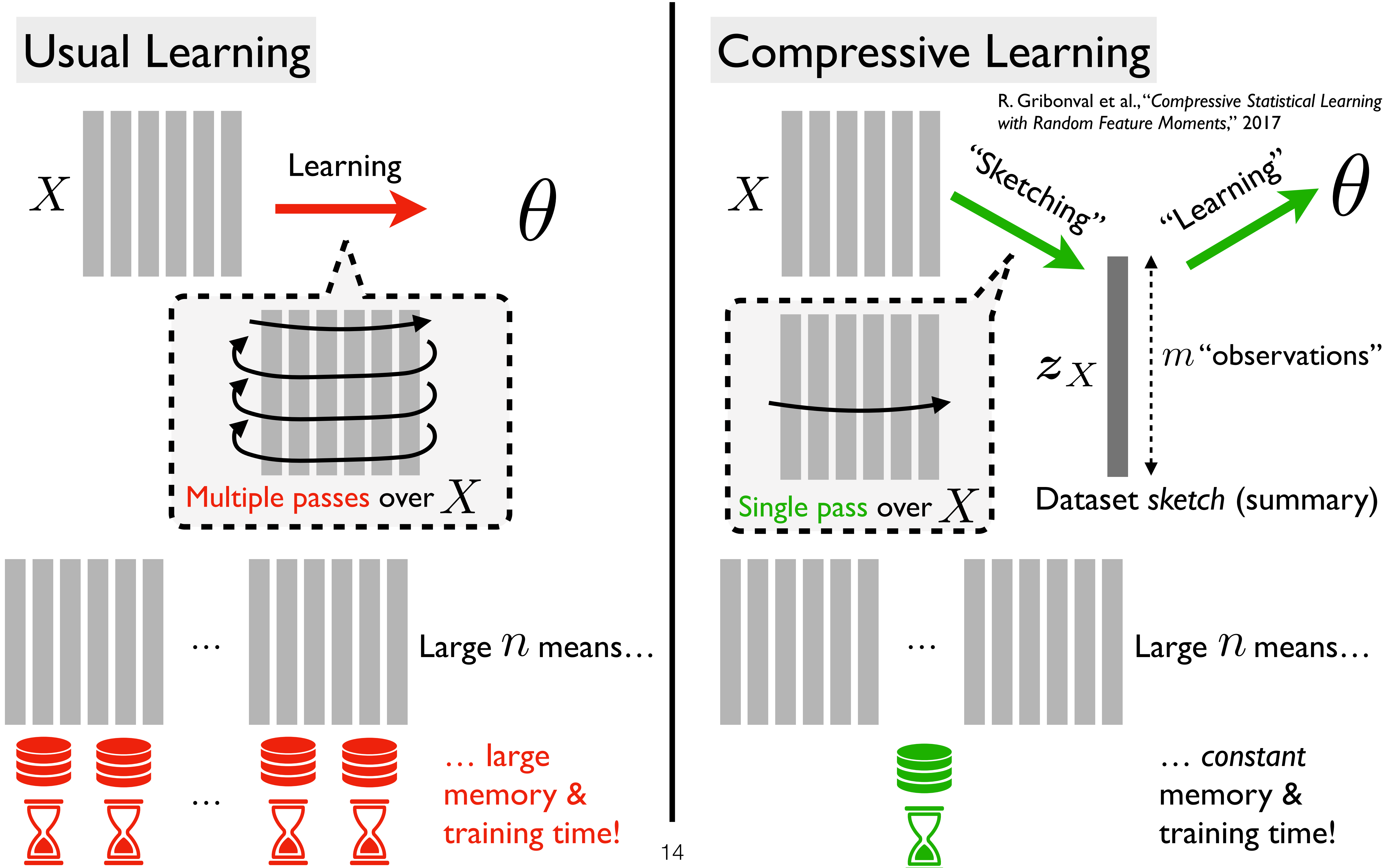




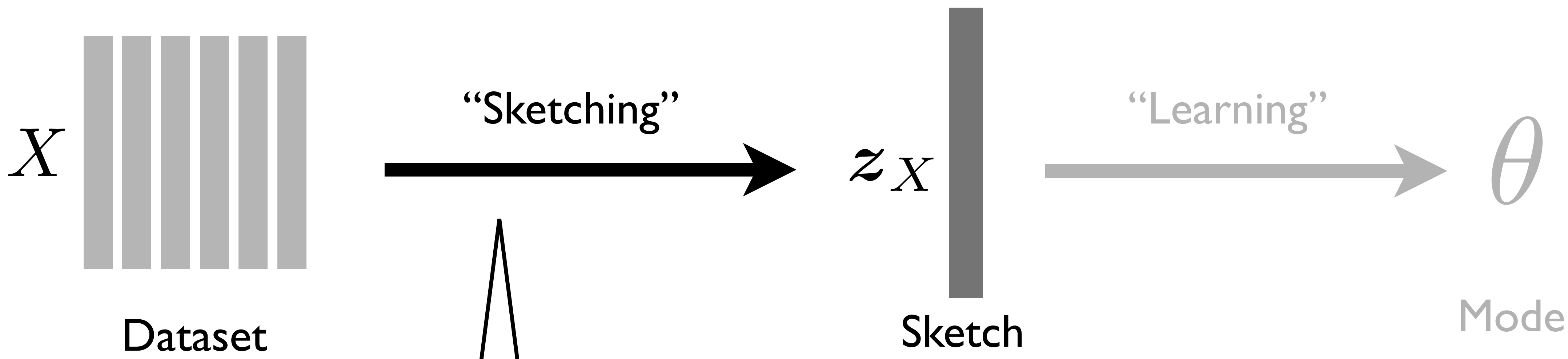
Large n means…

… large memory & training time!

Compressive Learning

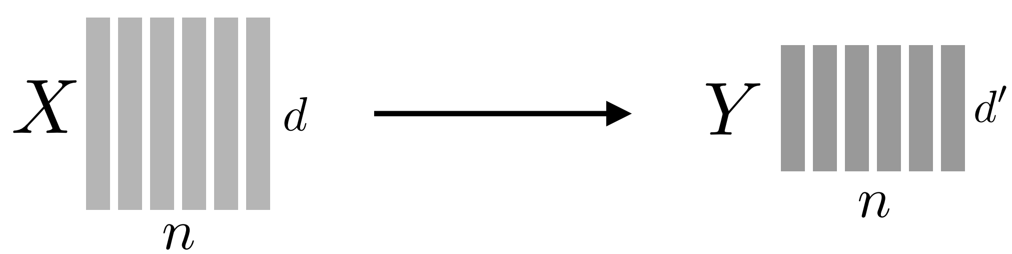


How to compress a dataset?

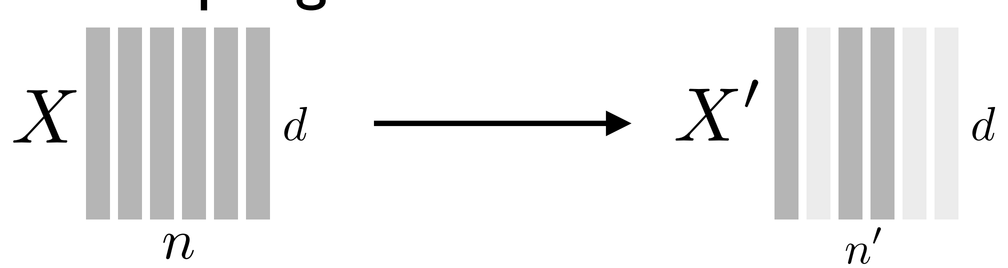


Approaches to "dataset compression":

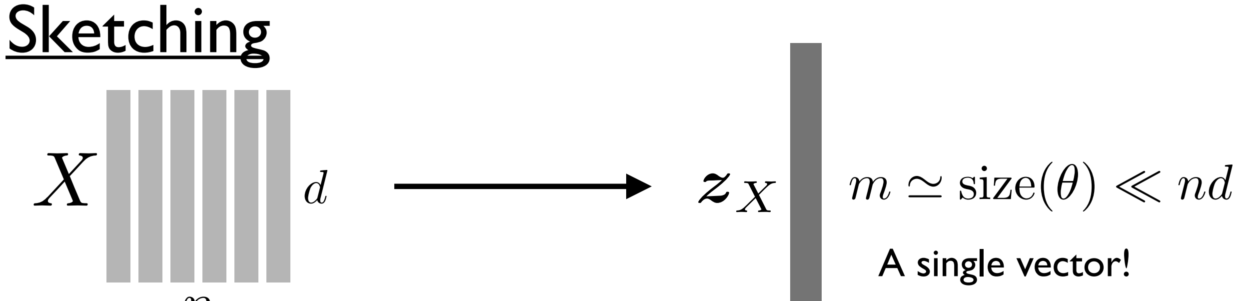
- *Element-wise* dimensionality reduction



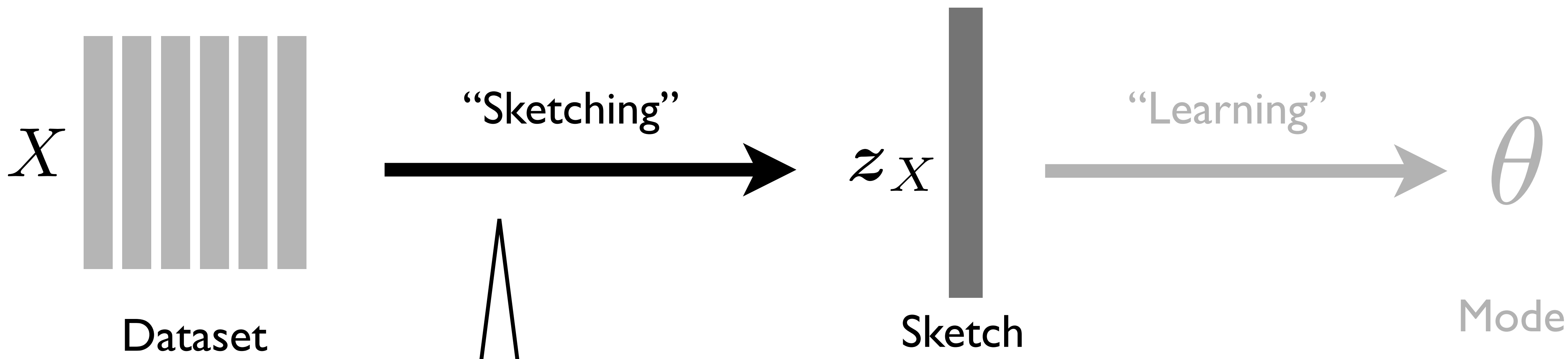
- Subsampling



- <u>Sketching</u>

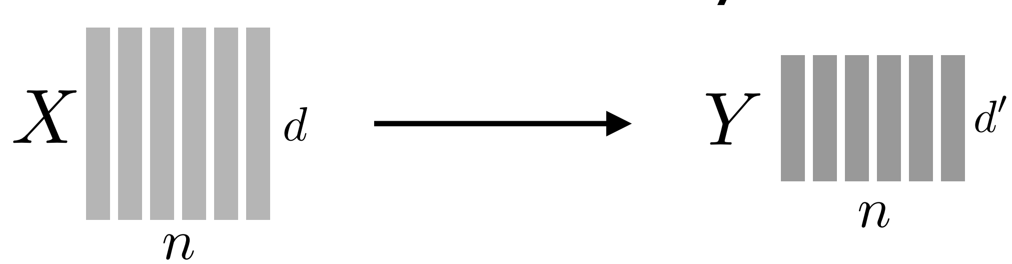


How to sketch a dataset?

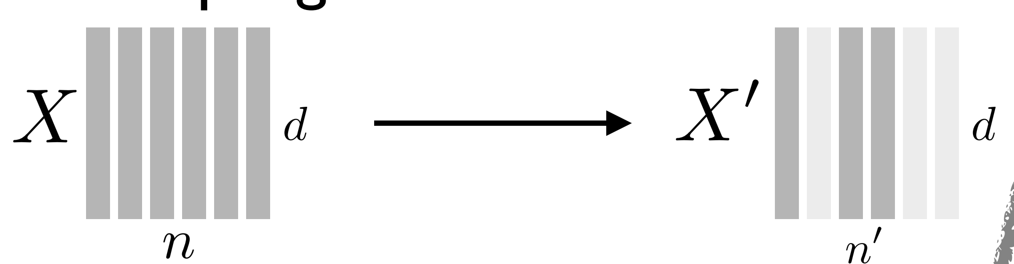


Approaches to “dataset compression”:

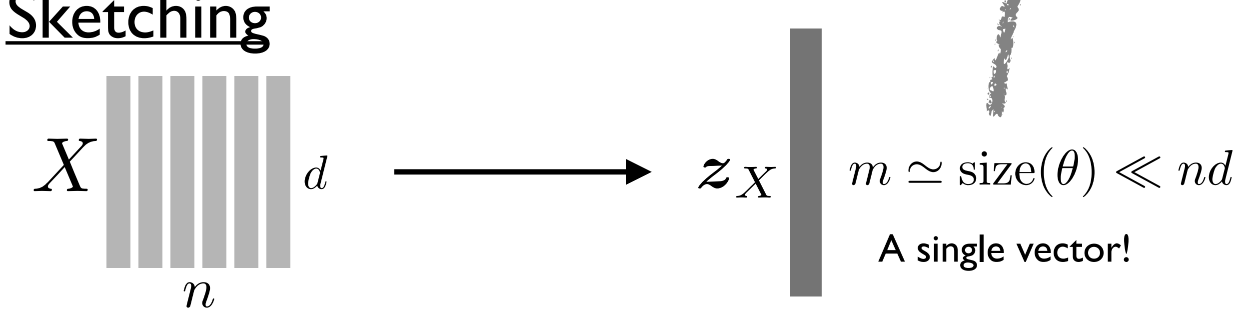
- *Element-wise* dimensionality reduction



- Subsampling



- <u>Sketching</u>



Sketch operator

$\Omega \in \mathbb{R}^{d \times m}$

$$\boldsymbol{z}_X = \frac{1}{n} \sum_{\boldsymbol{x}_i \in X} \boldsymbol{z}_{\boldsymbol{x}_i} \qquad \text{e.g.,} \quad \boldsymbol{z}_{\boldsymbol{x}_i} := \exp(\mathrm{i}\Omega^T \boldsymbol{x}_i)$$

Empirical <u>average</u> of nonlinear features (generalized moments)

Random Fourier Features (sampling the datas *characteristic function*)

Rahimi, Recht, “*Random Features for Large-scale Kernel Machines*”,NIPS 2007]

How to learn from the sketch?

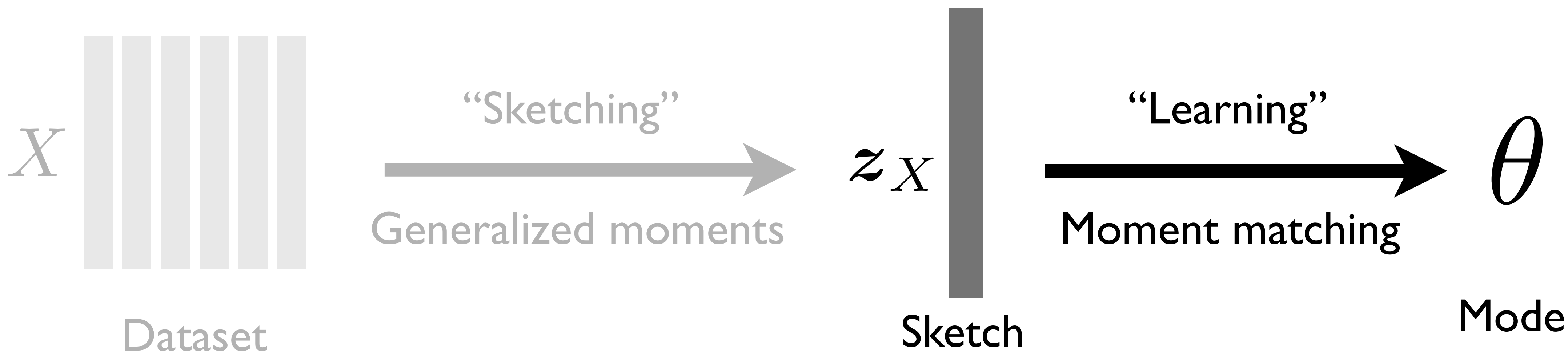
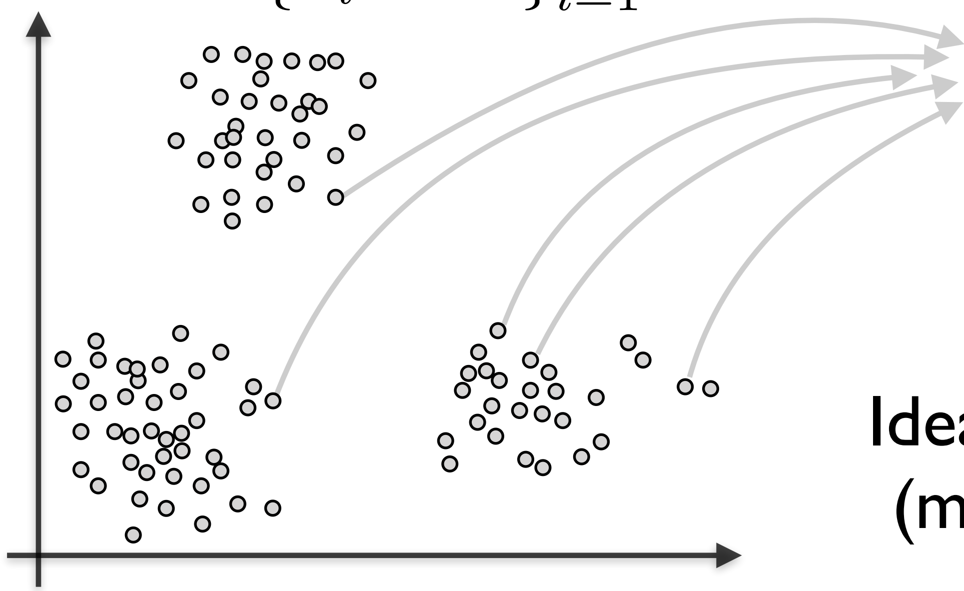


Illustration here: Compressive K-Means

Compressive K-Means

$X = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^n$



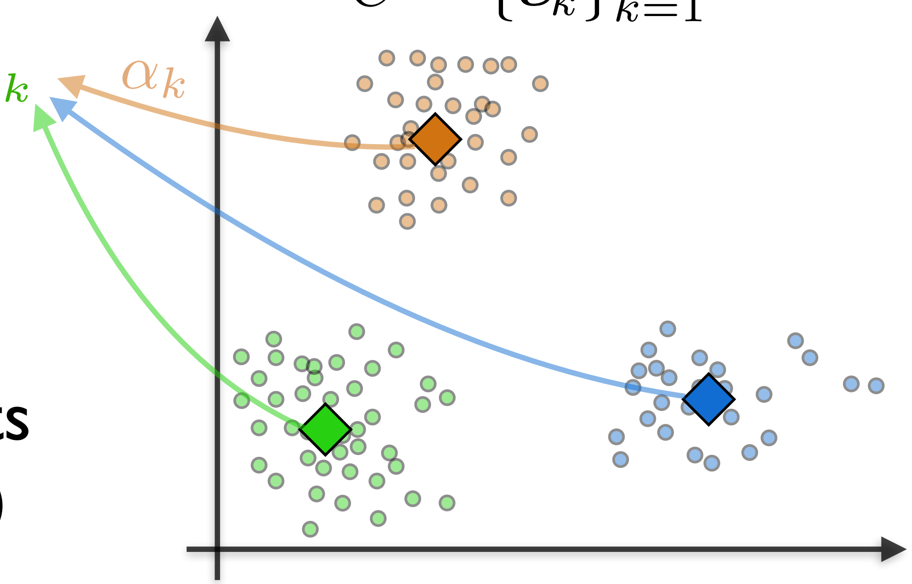
$$z_X \simeq \sum_{k=1}^{K} \alpha_k z_{\mathbf{c}_k}$$

Dataset sketch

Centroids sketch

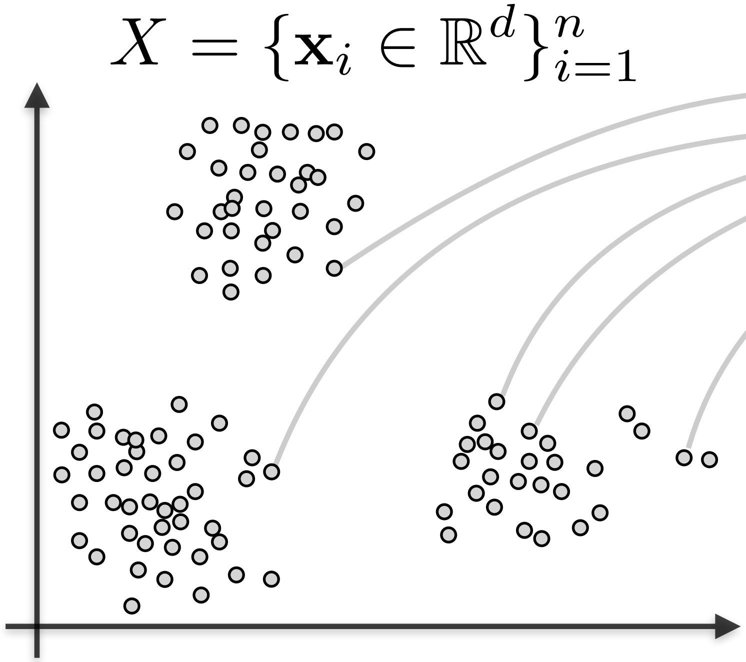
Idea: match the measurements (moment matching problem)

$C = \{\mathbf{c}_k\}_{k=1}^K$



[N. Keriven et al., "*Compressive K-Means*", IEEE ICASSP, 2017]

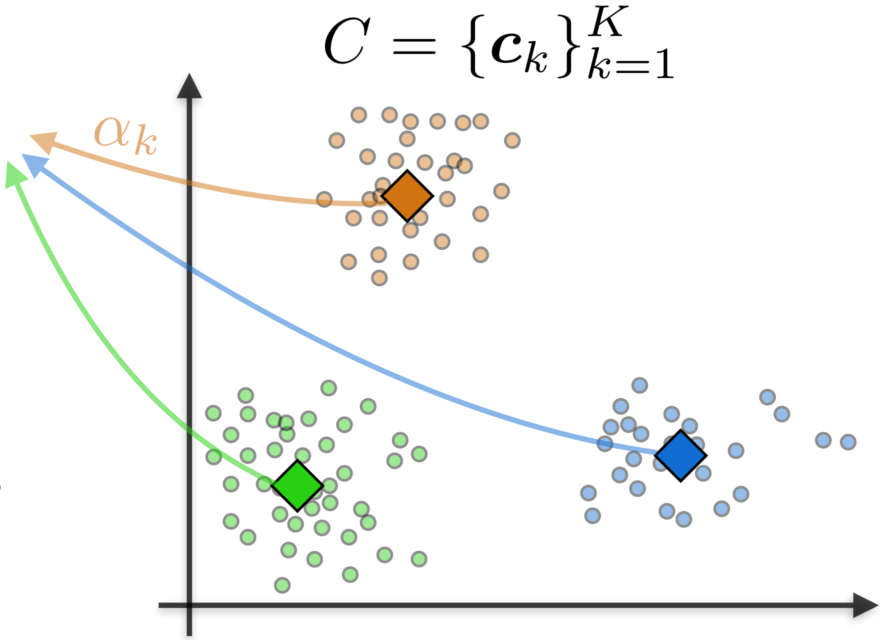
Compressive K-Means



$$\boldsymbol{z}_X \simeq \sum_{k=1}^{K} \alpha_k \boldsymbol{z}_{\boldsymbol{c}_k}$$

Dataset sketch

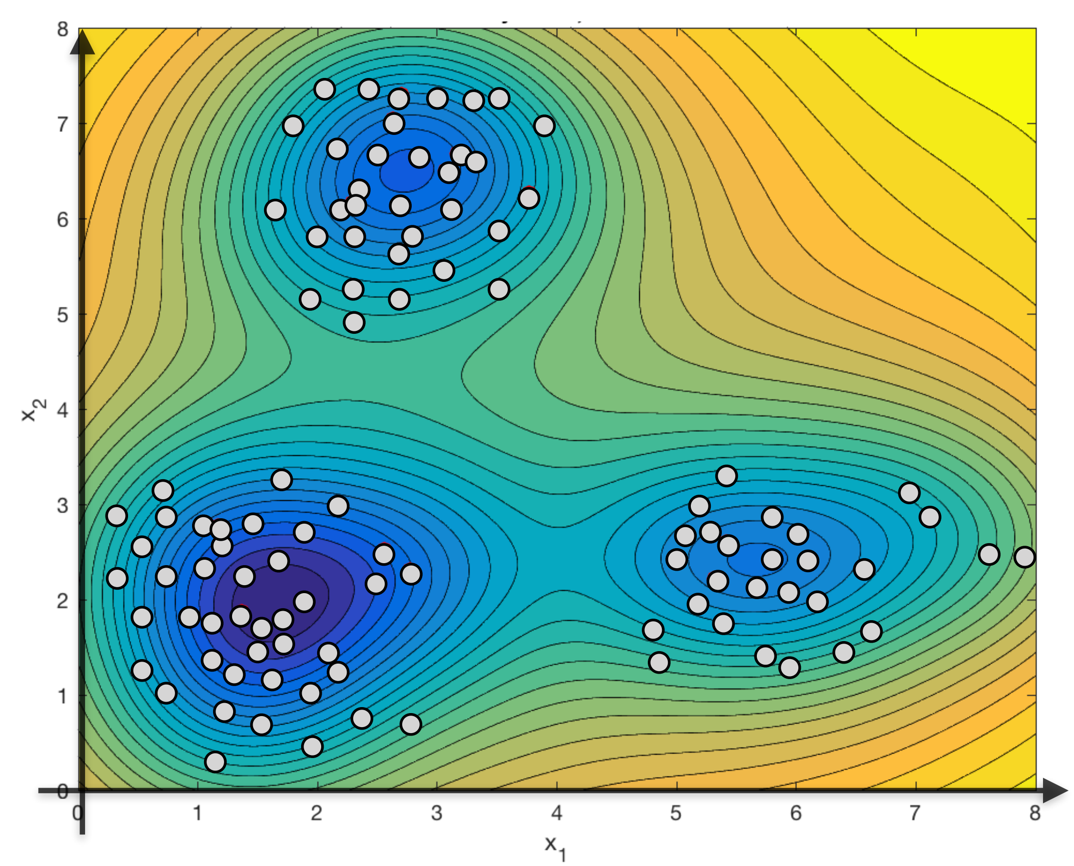
Centroids sketch



Idea: match the measurements (moment matching problem)

$$\min_{C,\boldsymbol{\alpha}} \|\boldsymbol{z}_X - \sum_{k=1}^{K} \alpha_k \boldsymbol{z}_{\boldsymbol{c}_k}\|_2^2$$

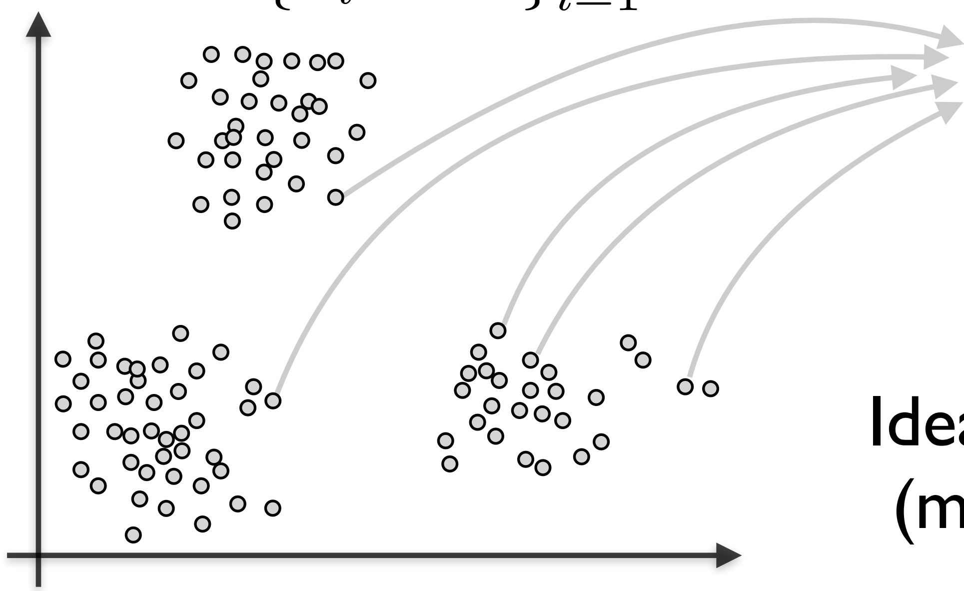
Nonconvex optimization!



[N. Keriven et al., "*Compressive K-Means*", IEEE ICASSP, 2017]

Compressive K-Means

$X = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^n$

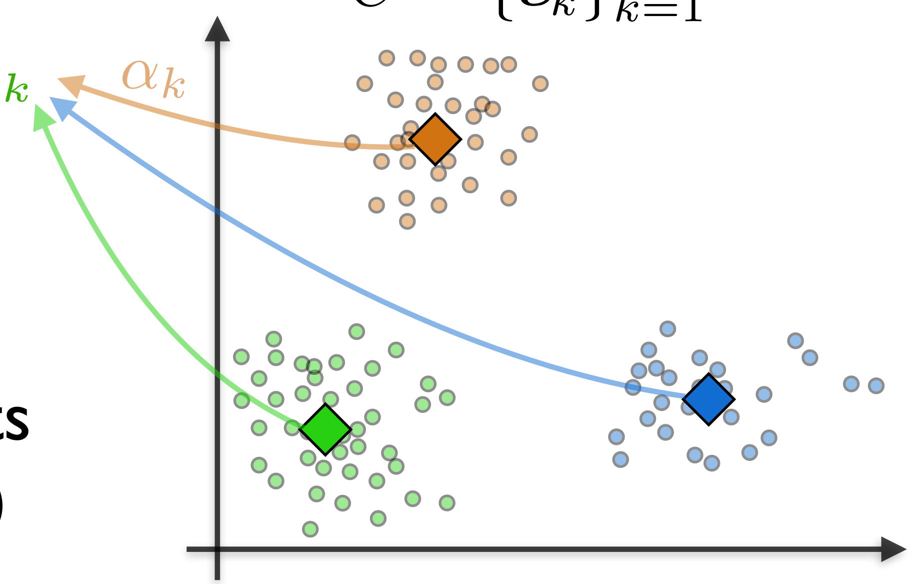


$$\boldsymbol{z}_X \simeq \sum_{k=1}^{K} \alpha_k \boldsymbol{z}_{\boldsymbol{c}_k}$$

Dataset sketch

Centroids sketch

$C = \{\boldsymbol{c}_k\}_{k=1}^K$



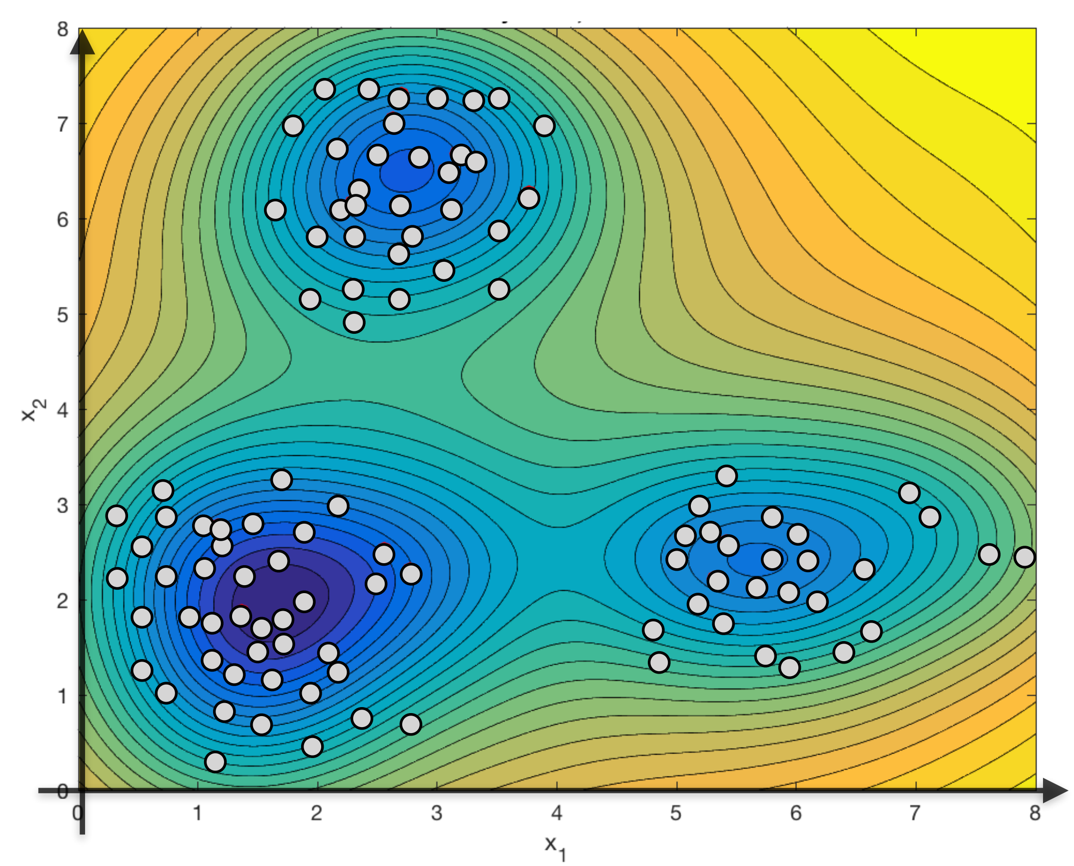
Idea: match the measurements (moment matching problem)

$$\min_{C,\boldsymbol{\alpha}} \|\boldsymbol{z}_X - \sum_{k=1}^{K} \alpha_k \boldsymbol{z}_{\boldsymbol{c}_k}\|_2^2$$

Nonconvex optimization!

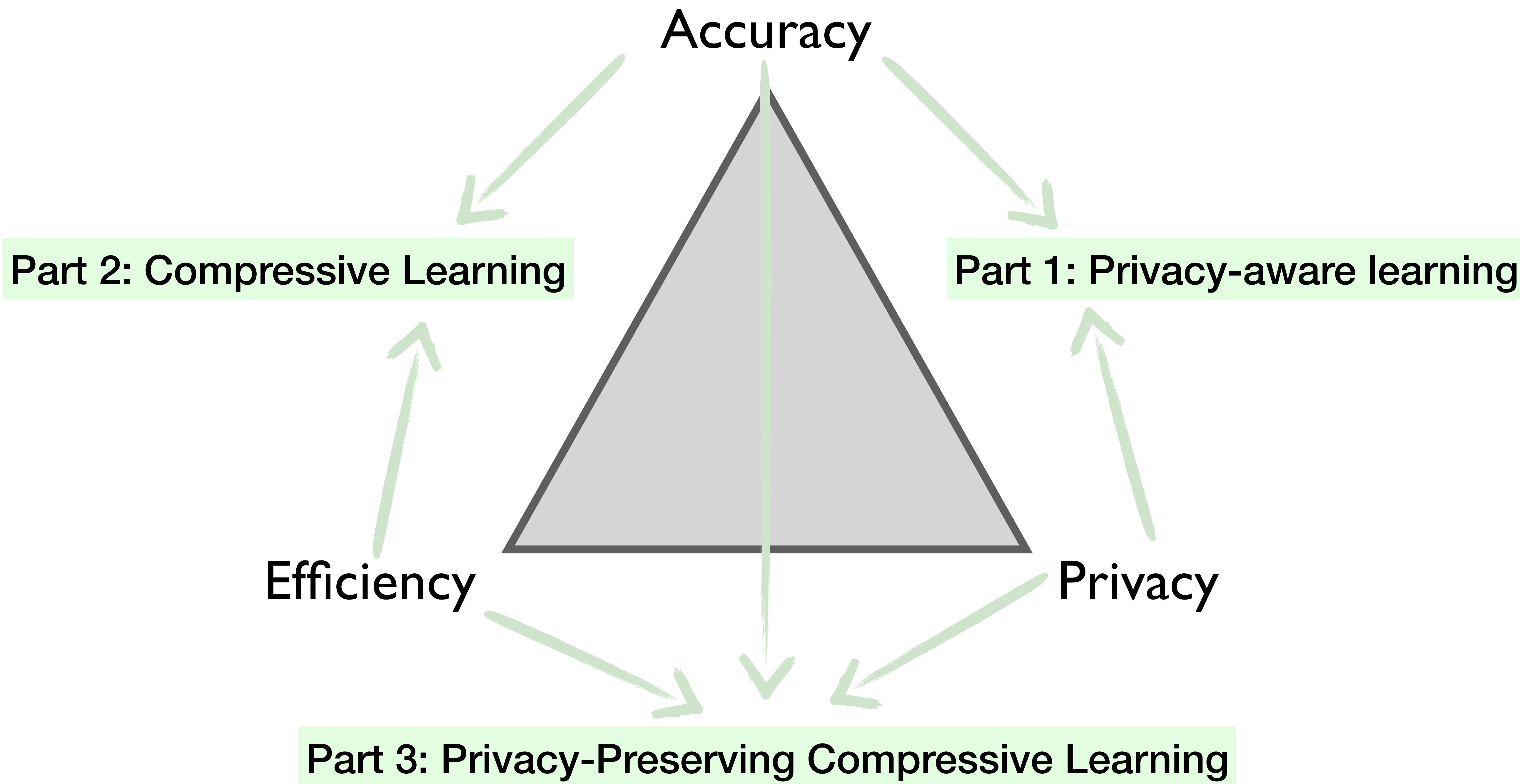
Empirically, works when

$$m = \mathcal{O}(\underbrace{Kd}_{\text{Model size}})$$



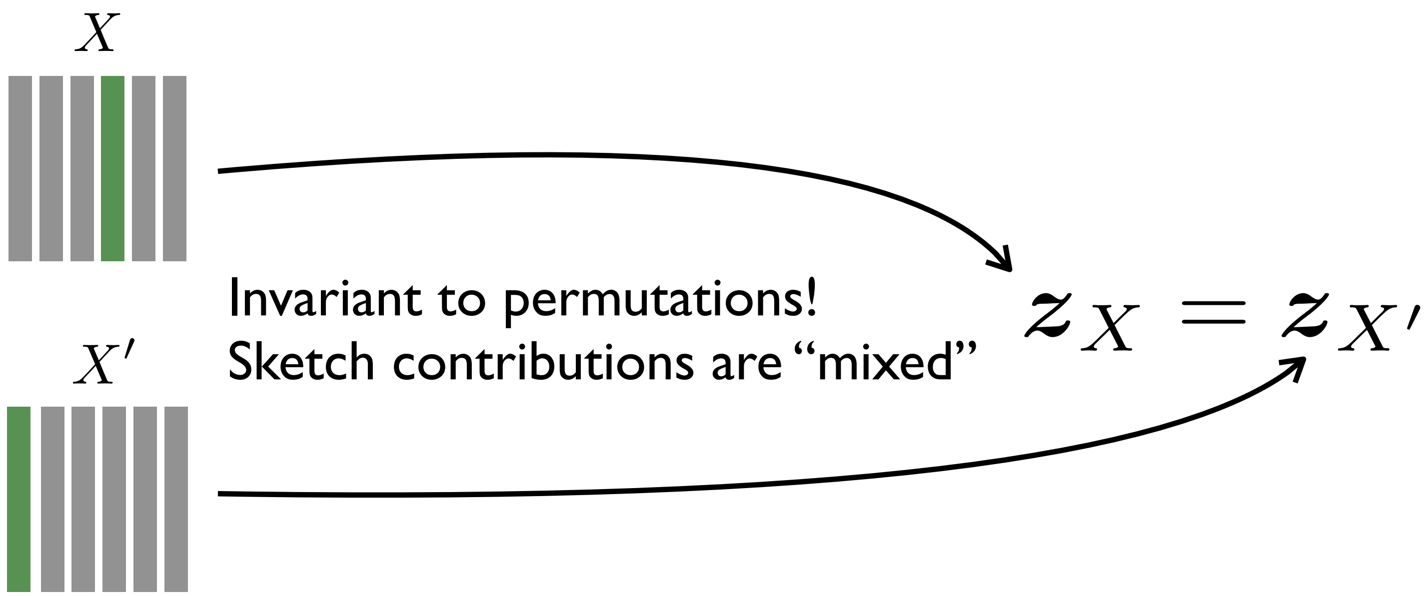
[N. Keriven et al., "*Compressive K-Means*", IEEE ICASSP, 2017]

This work


Accuracy
Part 2: Compressive Learning
Part 1: Privacy-aware learning
Efficiency
Privacy
Part 3: Privacy-Preserving Compressive Learning

Compressive Learning and Privacy

Intuitively, releasing only the sketch provides some form of *anonymity*…

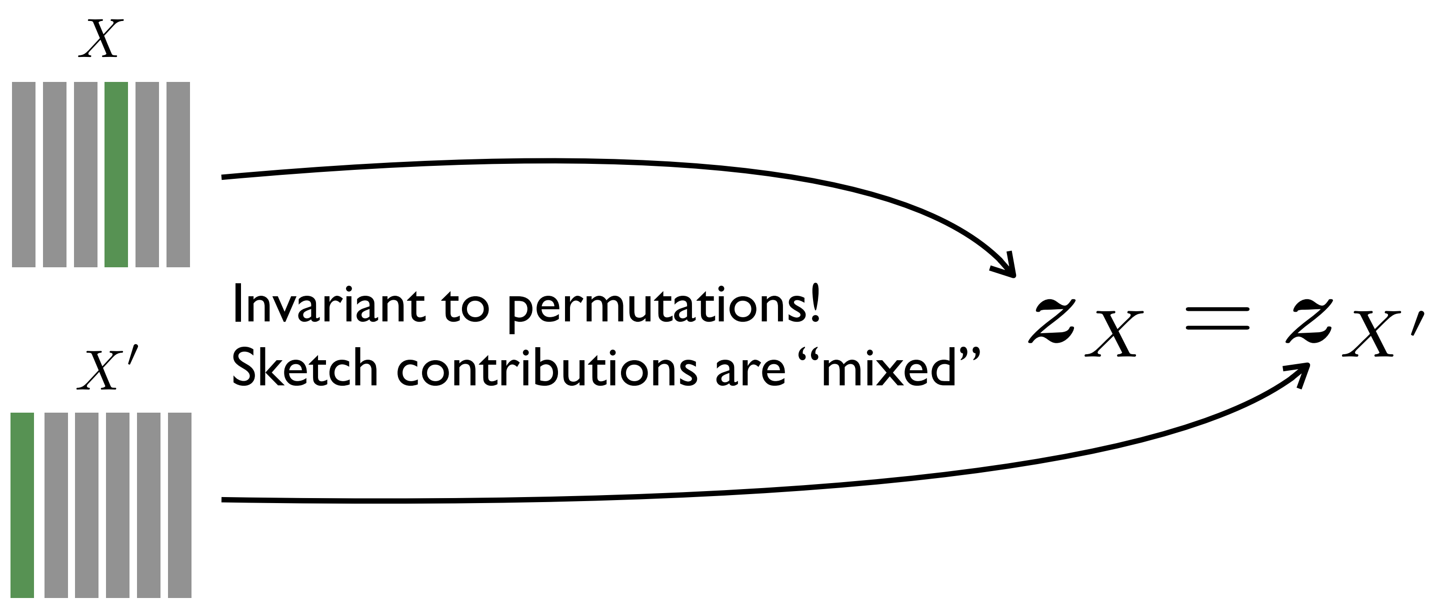


Sketch operator

$$\boldsymbol{z}_X = \frac{1}{n} \sum_{\boldsymbol{x}_i \in X} \exp(\mathrm{i}\Omega^T \boldsymbol{x}_i)$$

Compressive Learning and Privacy

Intuitively, releasing only the sketch provides some form of *anonymity*…



Sketch operator

$$\boldsymbol{z}_X = \frac{1}{n} \sum_{\boldsymbol{x}_i \in X} \exp(\mathrm{i}\Omega^T \boldsymbol{x}_i)$$

Our aim: stronger & formal privacy guarantee: Differential Privacy!

A good match!

CL: "forgets the individual signals
and stores only statistics of the dataset"

DP: "good when output not much influenced by one signal"

$\epsilon - \mathrm{DP}$

f satisfies $\epsilon - \mathrm{DP}$ if: $\forall S$ $\forall X \sim X'$

$$\mathbb{P}[f(X) \in S] \leq e^{\epsilon} \cdot \mathbb{P}[f(X') \in S]$$

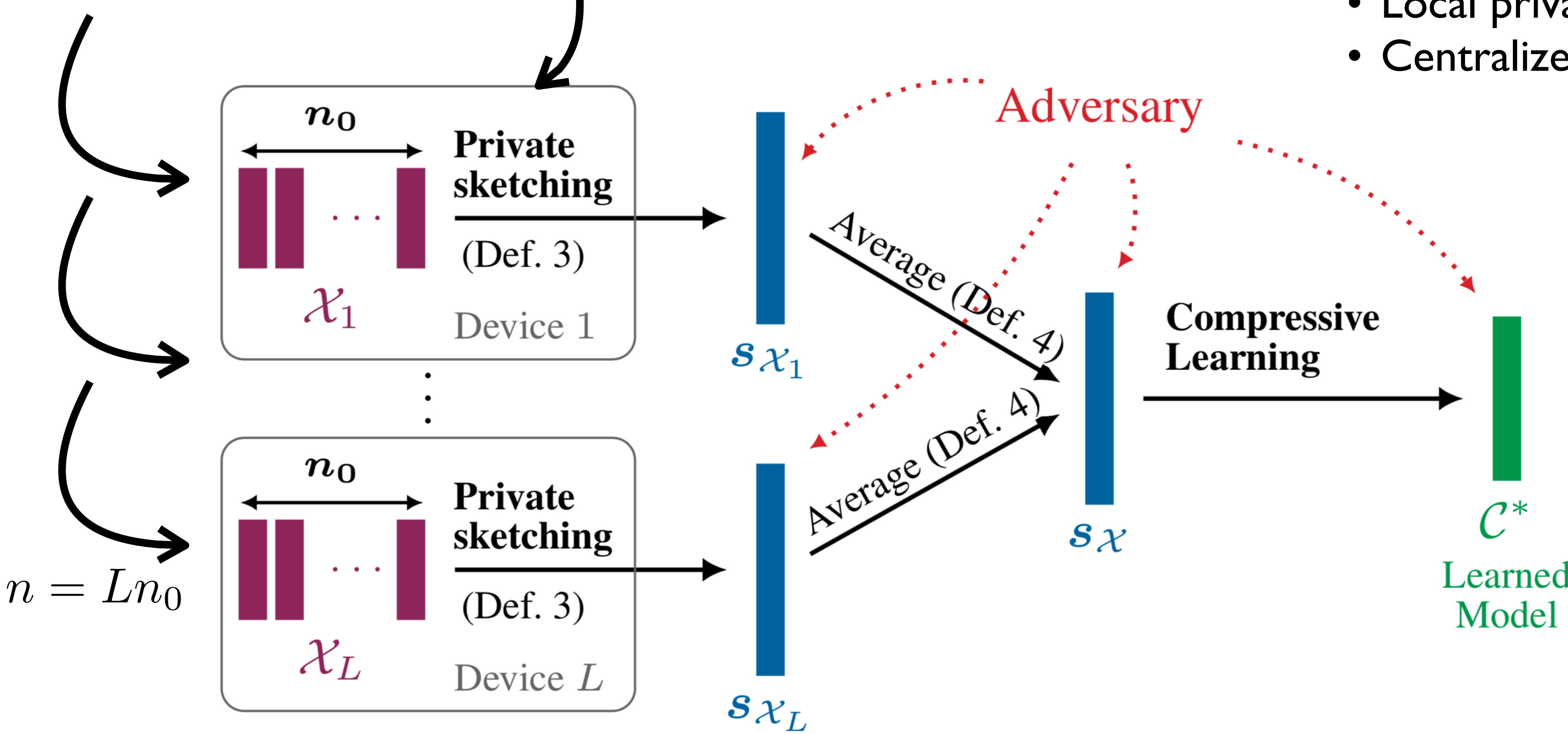
Philosophy: "learn from the data, not about the data"

Private CL: attack model

Dataset is *shared* across L devices, each holding n_0 distinct samples…

Extreme cases:

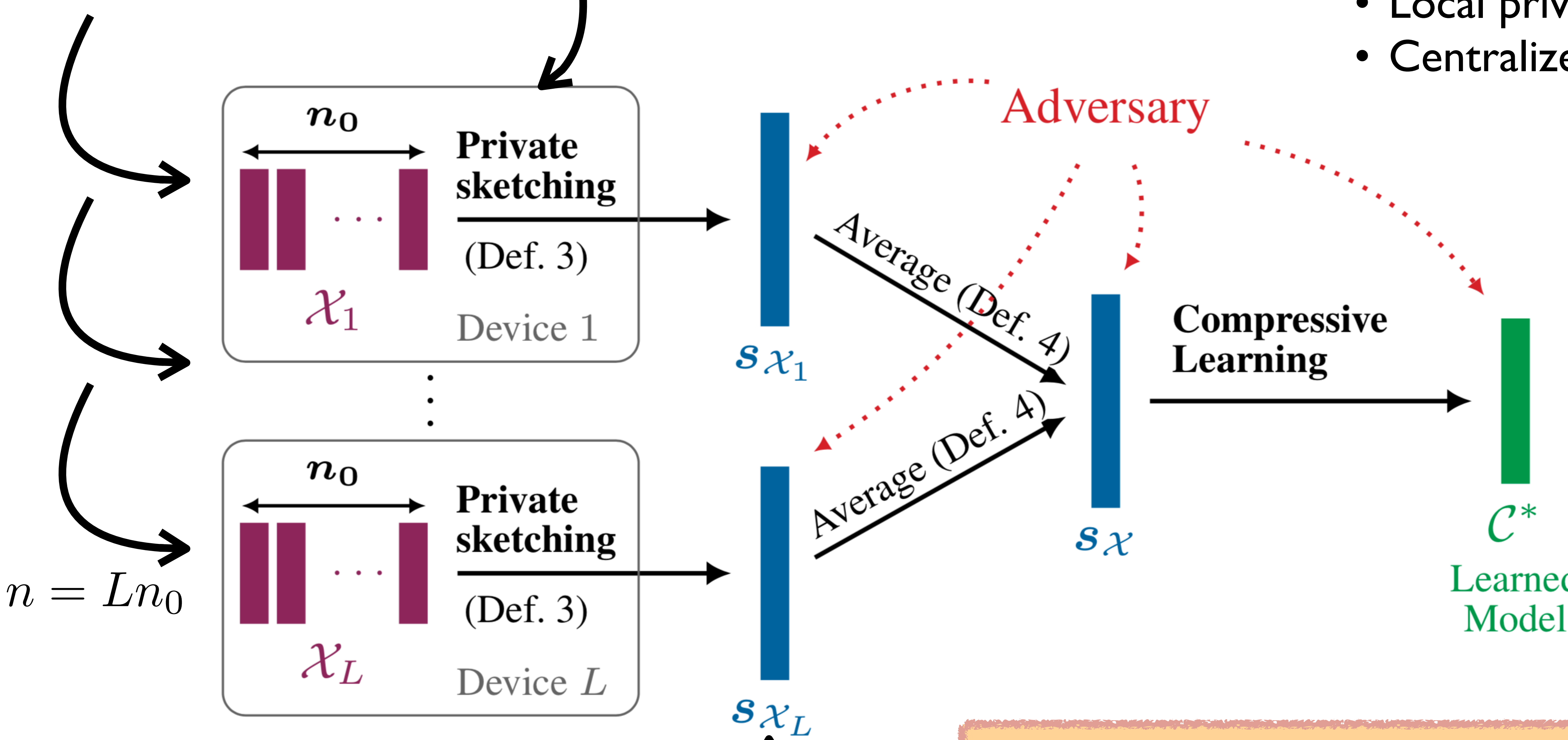
- Local privacy, $n_0 = 1$
- Centralized privacy, $L = 1$



Private CL: attack model

Dataset is *shared* across L devices, each holding n_0 distinct samples…

Extreme cases:
- Local privacy, $n_0 = 1$
- Centralized privacy, $L = 1$



…and releasing a *privacy-preserving local sketch*!

Important remarks

1) The adversary can know the sketch operator!

2) It is randomly drawn but *fixed*,
i.e., <u>additional noise is necessary</u>!

Differentially Private Sketching

Proposed mechanism: Laplacian mechanism *and subsampling* on sketches

Private sketch mechanism

$$\boldsymbol{s}_X := \frac{1}{n} \sum_{\boldsymbol{x}_i \in X} \left(\exp(\mathrm{i}\Omega^T \boldsymbol{x}_i) \odot \boldsymbol{b}_i\right) + \boldsymbol{\xi}$$

Subsampling: binary mask, keeps r values

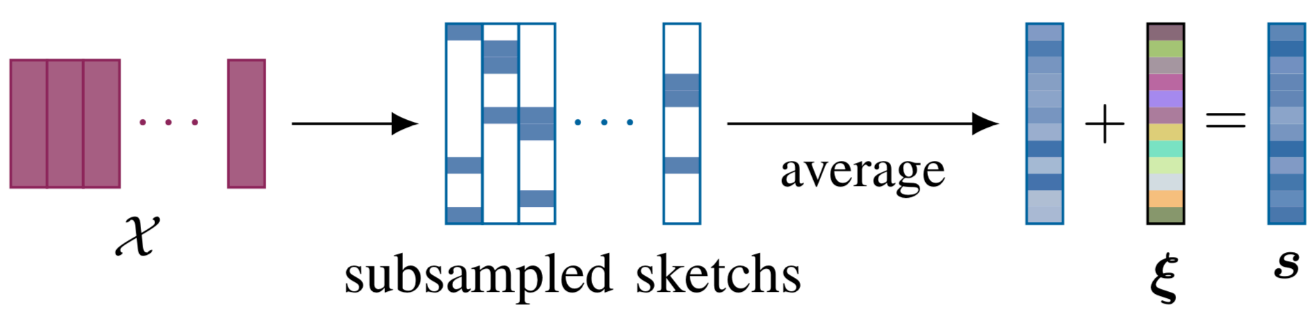
$\boldsymbol{b}_i \in \{0,1\}^m, \ \|\boldsymbol{b}_i\|_1 = r$

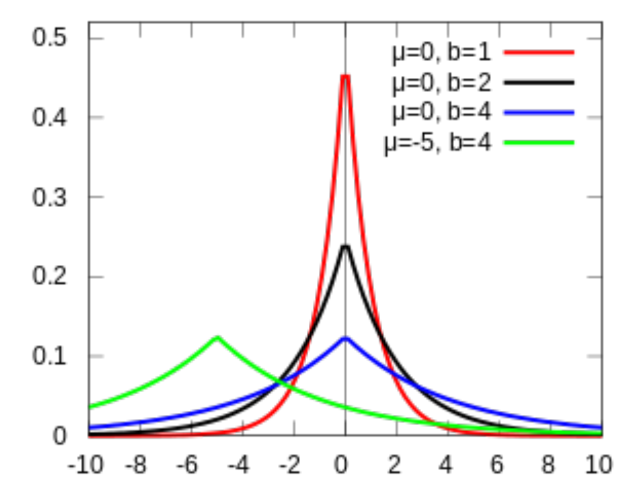
$$\xi_j \sim \mathrm{Lap}(\sigma_\xi/\sqrt{2})$$

Laplace random variable

$n \sim \mathrm{Lap}(b)$ has density $p_n(n) = \frac{1}{2b} e^{-\frac{|n|}{b}}$

Variance: $\sigma_n^2 = 2b^2$





Differentially Private Sketching

Proposed mechanism: Laplacian mechanism *and subsampling* on sketches

Level of noise?

Private sketch mechanism

$$\boldsymbol{s}_X := \frac{1}{n} \sum_{\boldsymbol{x}_i \in X} (\exp(\mathrm{i}\Omega^T \boldsymbol{x}_i) \odot \boldsymbol{b}_i) + \boldsymbol{\xi}$$

Subsampling: binary mask, keeps r values

$\boldsymbol{b}_i \in \{0,1\}^m, \ \|\boldsymbol{b}_i\|_1 = r$

$\xi_j \sim \mathrm{Lap}(\sigma_\xi / \sqrt{2})$

Theorem: the proposed mechanism is private:

If $\sigma_\xi \propto \dfrac{\sqrt{m}}{n_0 \epsilon}$, then $\boldsymbol{s}_X$ provides $\epsilon - \mathrm{DP}$ to the contributors of X

Differentially Private Sketching: proof

Theorem: the proposed mechanism

Keeps r values

$$\boldsymbol{s}_X := \frac{1}{n} \sum_{\boldsymbol{x}_i \in X} (\exp(\mathrm{i}\Omega^T \boldsymbol{x}_i) \odot \boldsymbol{b}_i) + \boldsymbol{\xi}$$

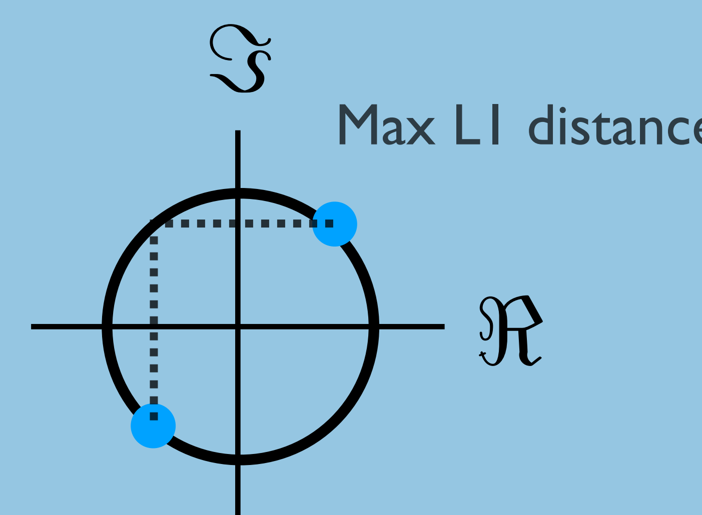
$\xi_j \sim \mathrm{Lap}(\sigma_\xi / \sqrt{2})$

where $\sigma_\xi \propto \frac{\sqrt{m}}{n_0 \epsilon}$

is $\epsilon - \mathrm{DP}$

Proof idea:

$$\frac{p(\boldsymbol{s}_X)}{p(\boldsymbol{s}_{X'})} \leq \exp\left(\frac{1}{\sigma_\xi n} \underbrace{\|\boldsymbol{z}_{\boldsymbol{x}} \odot \boldsymbol{b} - \boldsymbol{z}_{\boldsymbol{x}'} \odot \boldsymbol{b}\|_1}_{r \text{ nonzero entries}} \right)$$



Remark

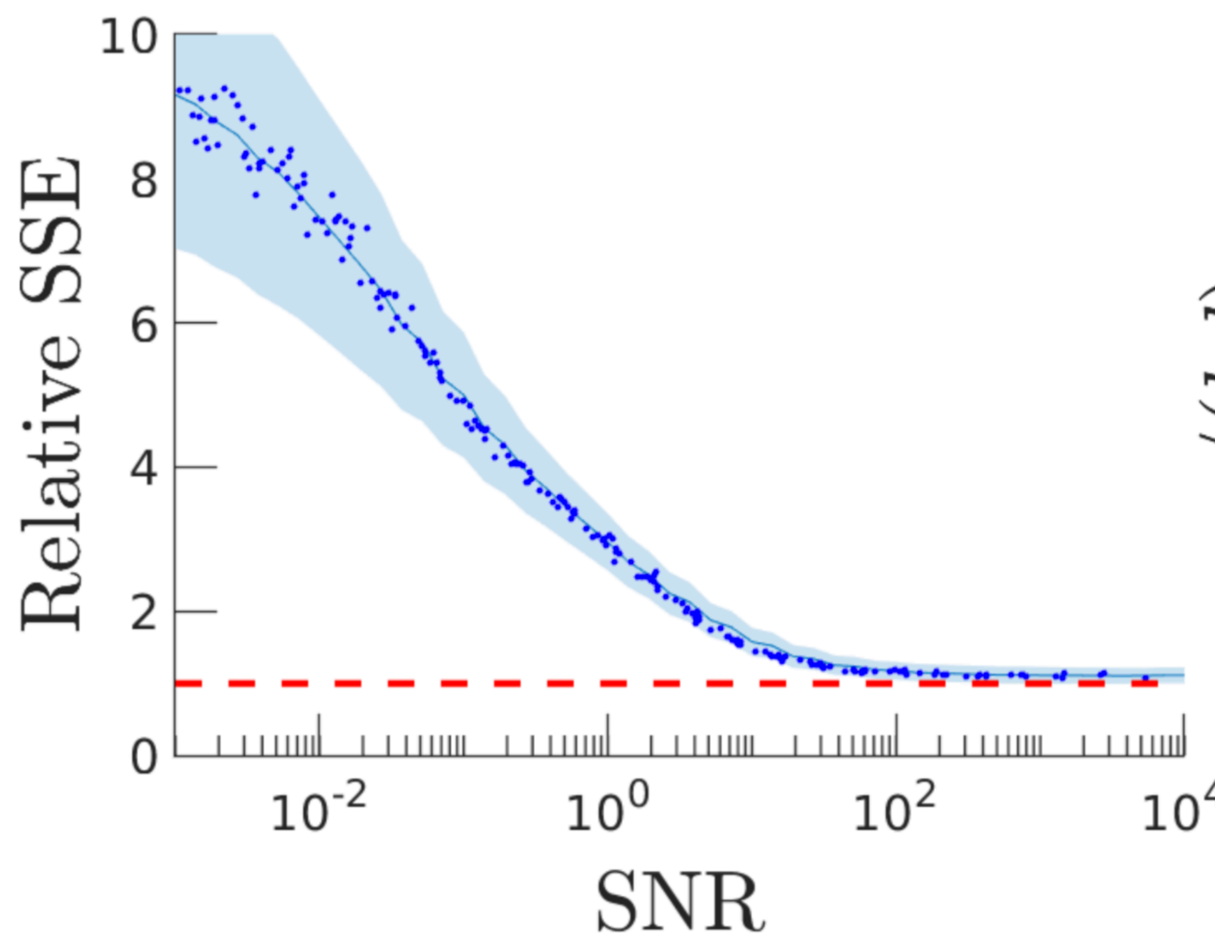
Ongoing work: the bound above is sharp (without additional constraints)

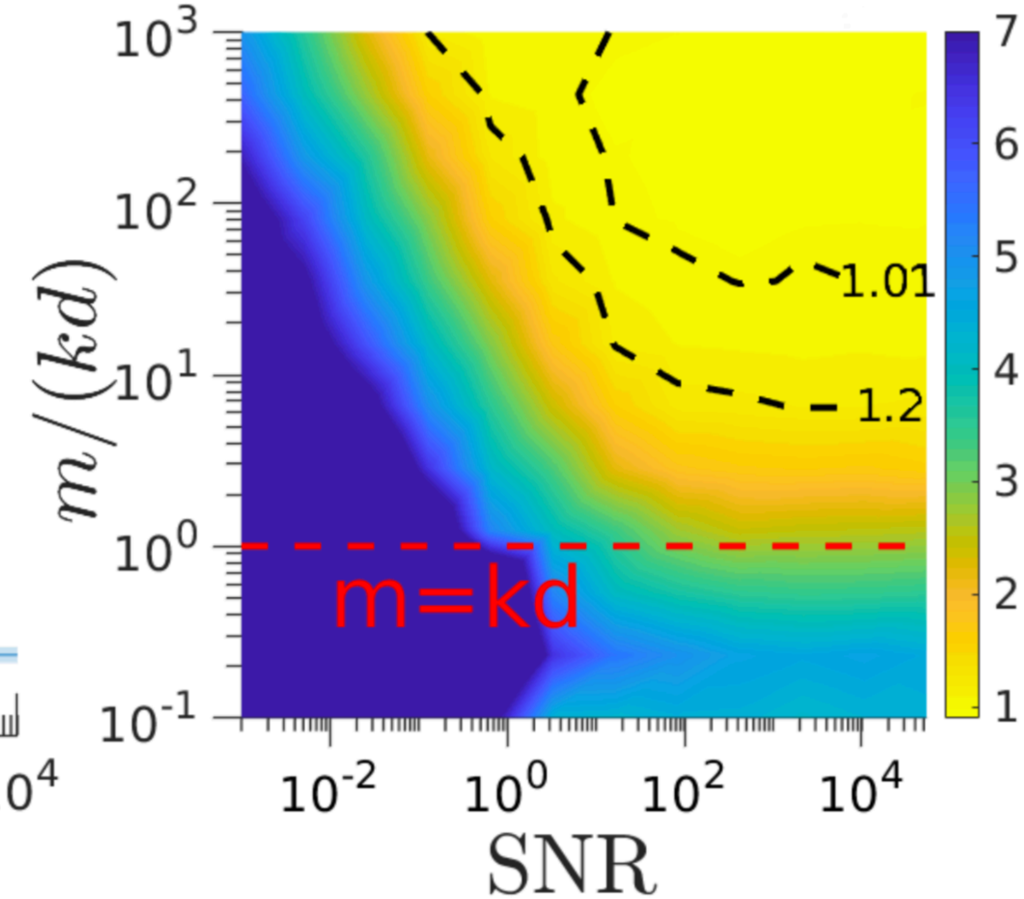
But… can we still learn?

I.e., what about "utility"??

How does the addition of noise and subsampling affect learning?

$$\text{SNR} \triangleq \frac{\|\boldsymbol{z}\|^2}{\sum_{j=1}^{m} \text{Var}((\boldsymbol{s}_{\mathcal{X}})_j)} = \frac{\alpha_r n_0 L \|\boldsymbol{z}\|^2}{1 - \alpha_r \|\boldsymbol{z}\|^2 + \sigma_\xi^2}.$$



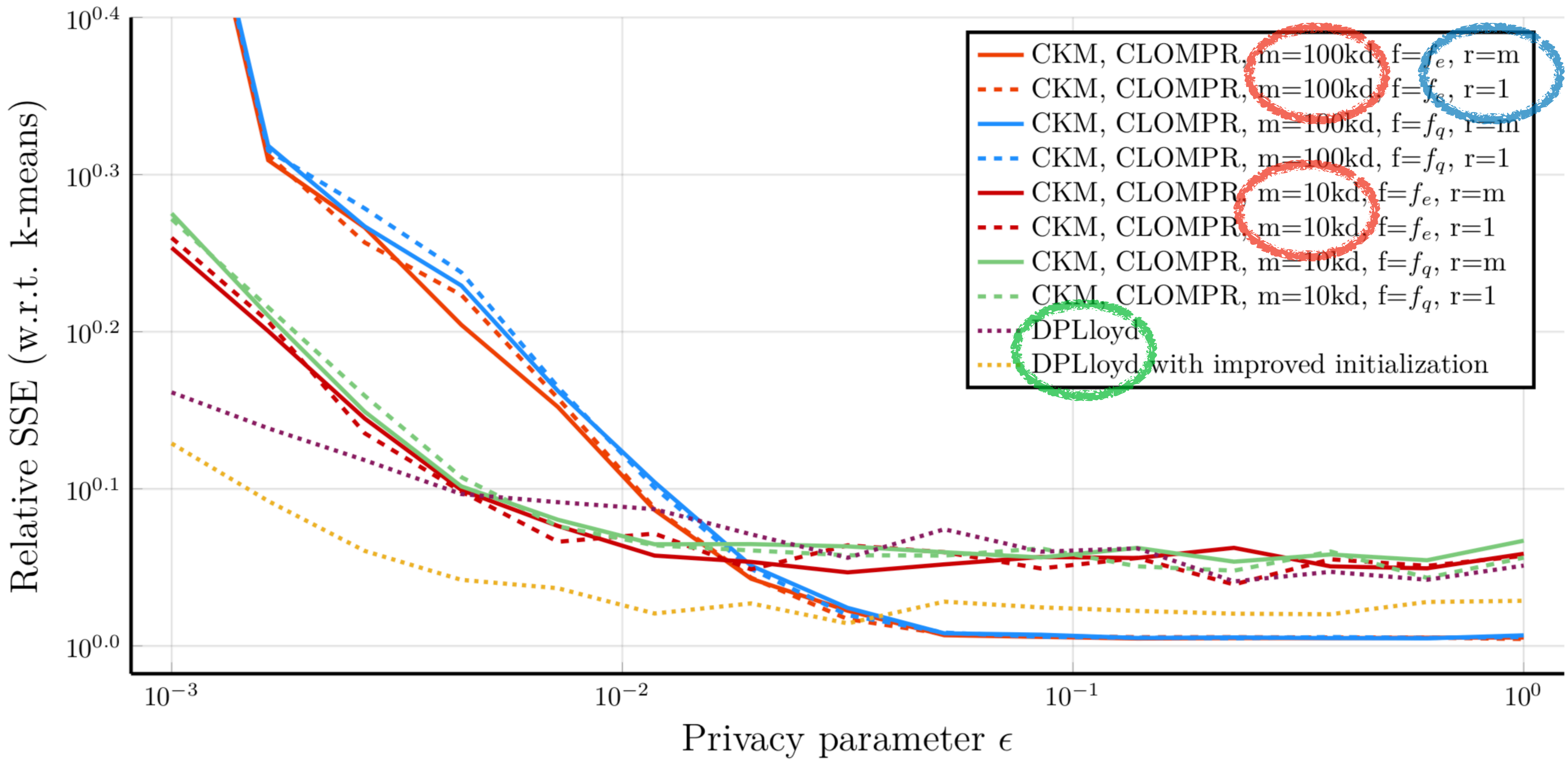


$$\text{SNR}(\epsilon; n_0, L, \alpha_r, m) = \frac{\alpha_r\, n_0\, L\, \delta}{1 - \alpha_r \delta + \frac{32 \alpha_r m^2}{n_0 \epsilon^2}}$$

The SNR helps to understand the effect of the parameters

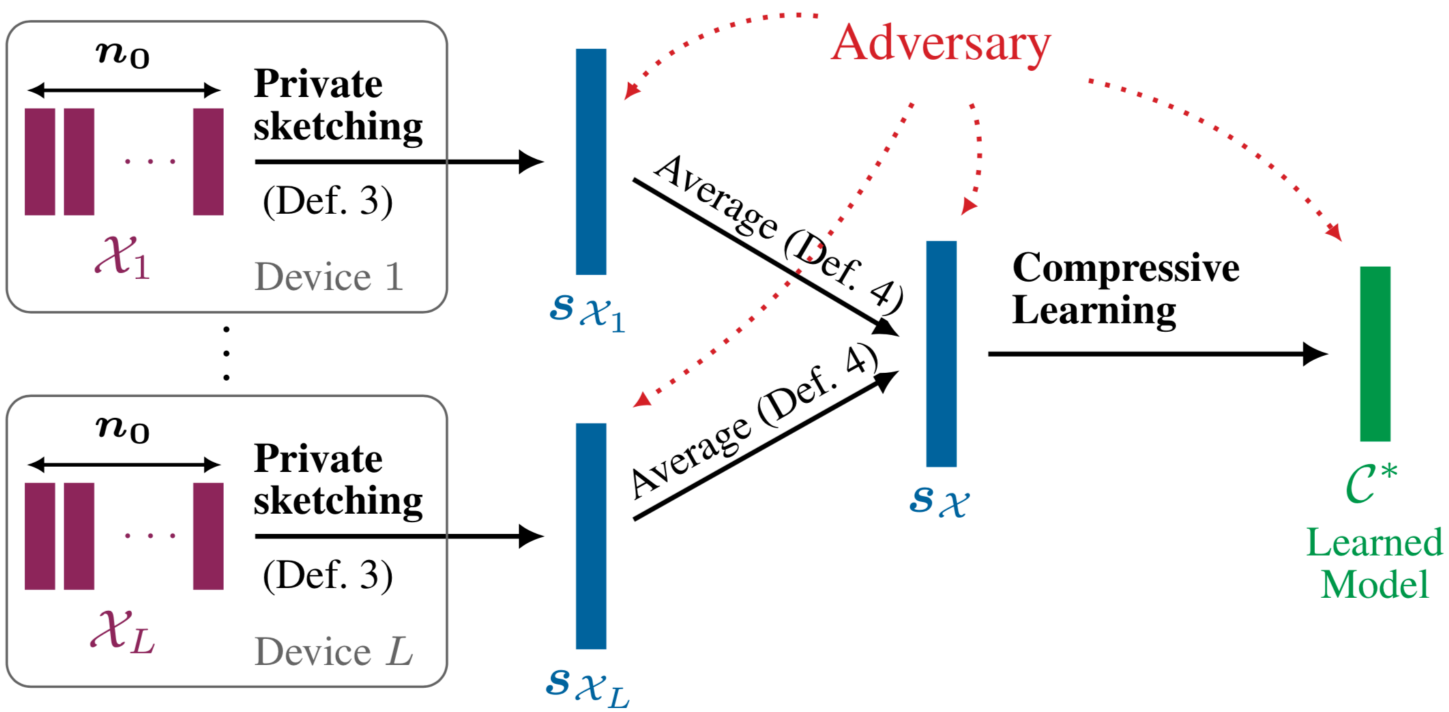
Privacy-utility tradeoff (case study)

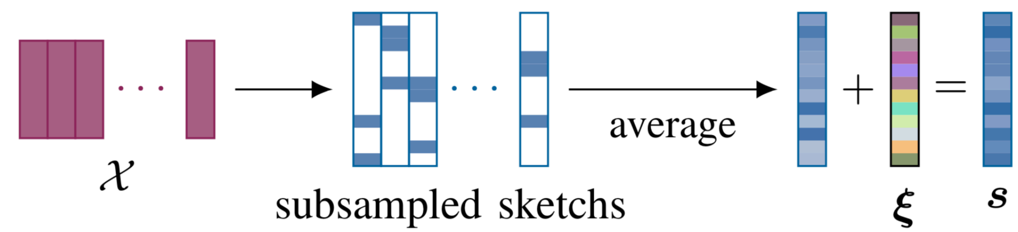
Some experimental privacy-utility curves (in a well-controlled environment)



… competitive with state-of-the-art Differentially Private K-Means :-)

Thank you!





$$s_X := \frac{1}{n} \sum_{x_i \in X} (\exp(\mathrm{i}\Omega^T x_i) \odot b_i) + \xi$$

$$\sigma_\xi \propto \frac{\sqrt{m}}{n_0 \epsilon}$$