Compressive learning (e.g. clustering) from a (quantized) sketch of the dataset

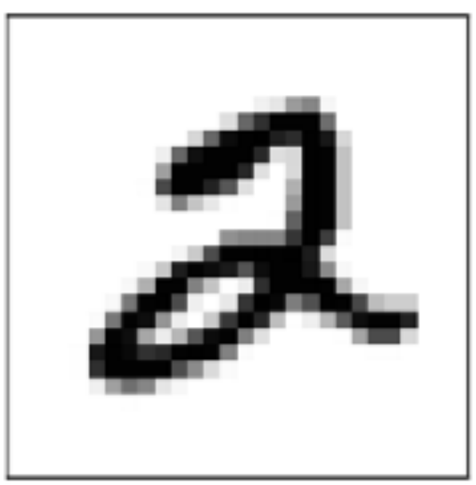
Vincent Schellekens

UCL Université catholique de Louvain

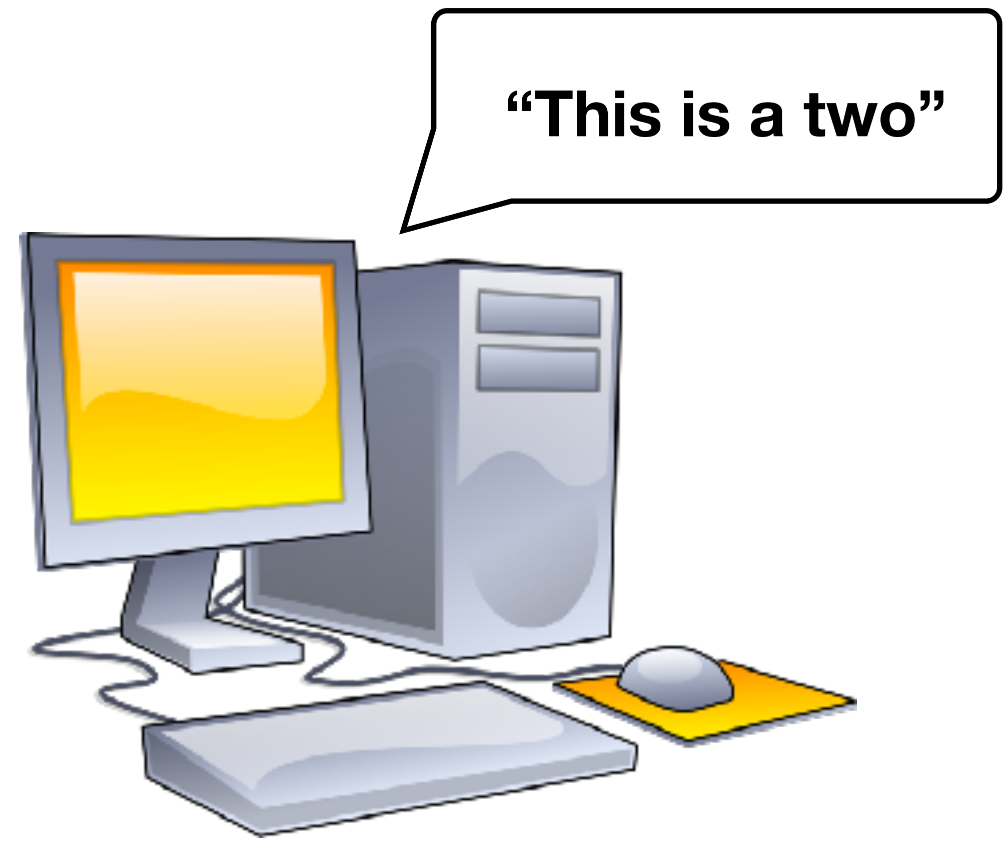
ISP GROUP Image & Signal Processing

Context: machine learning

A machine learning classic: hand-written digit recognition



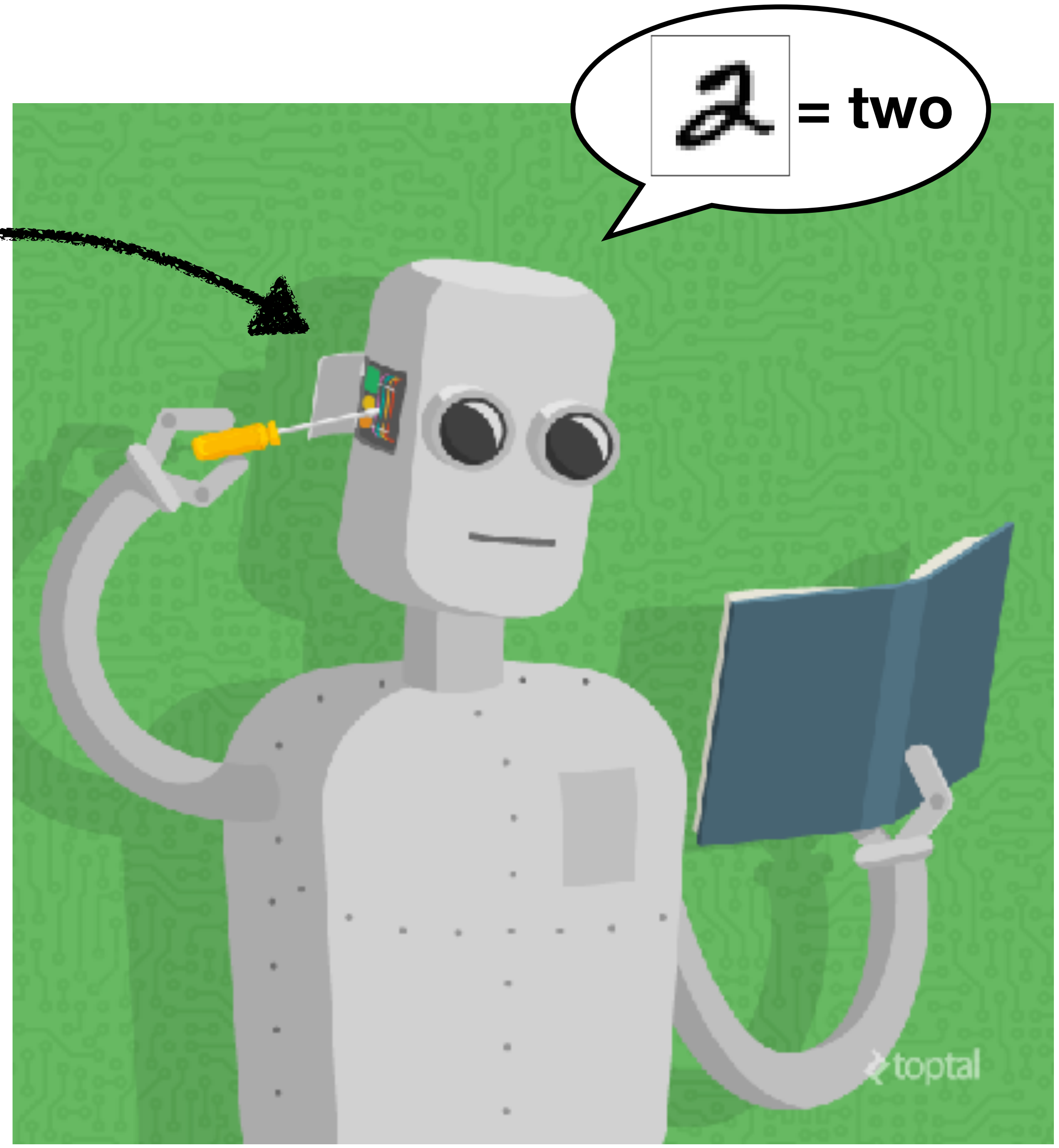
MNIST dataset



Verrrry difficult to program explicitly!

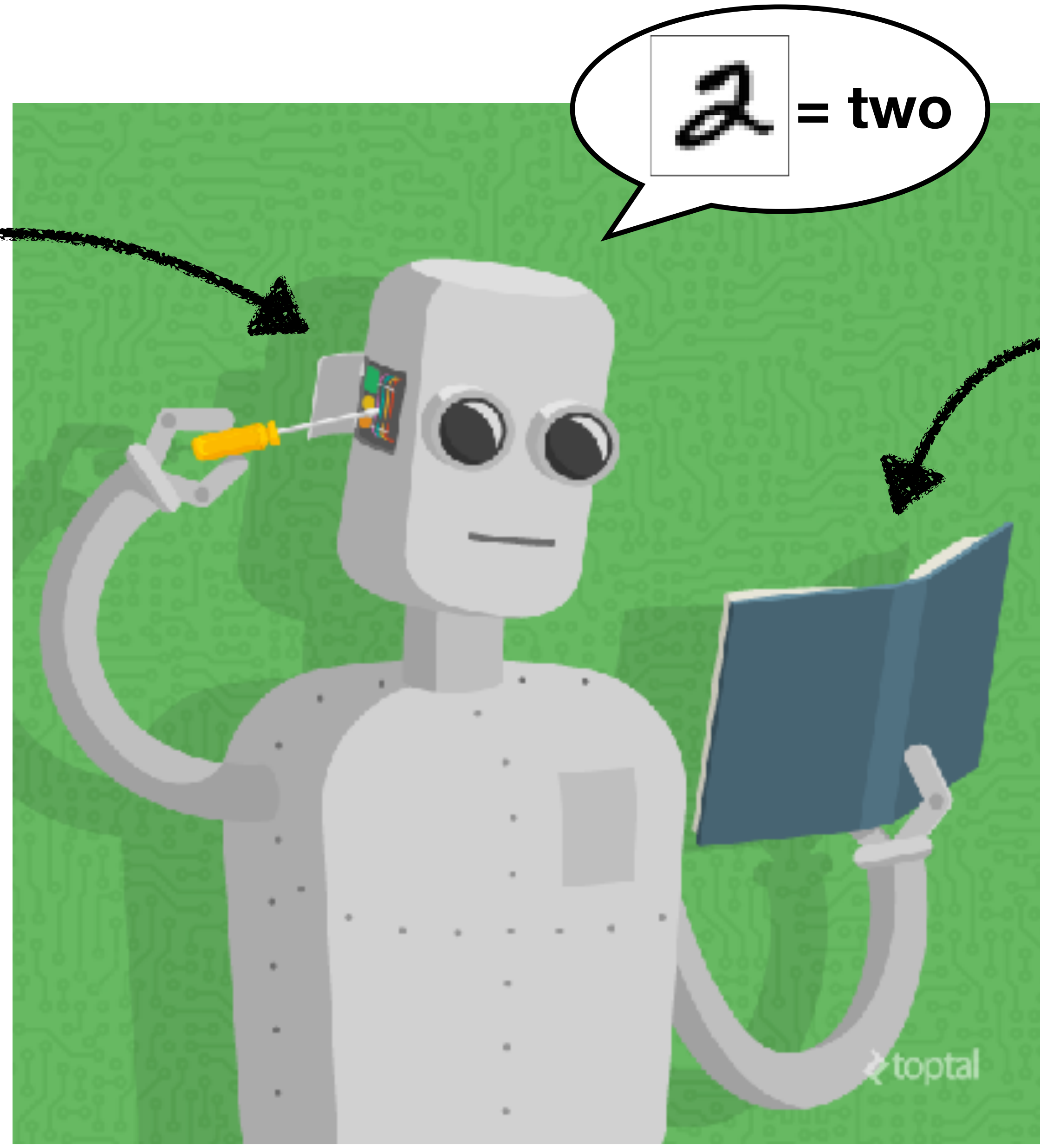
Context: machine learning

Solution: let the computer figure it out by itself!



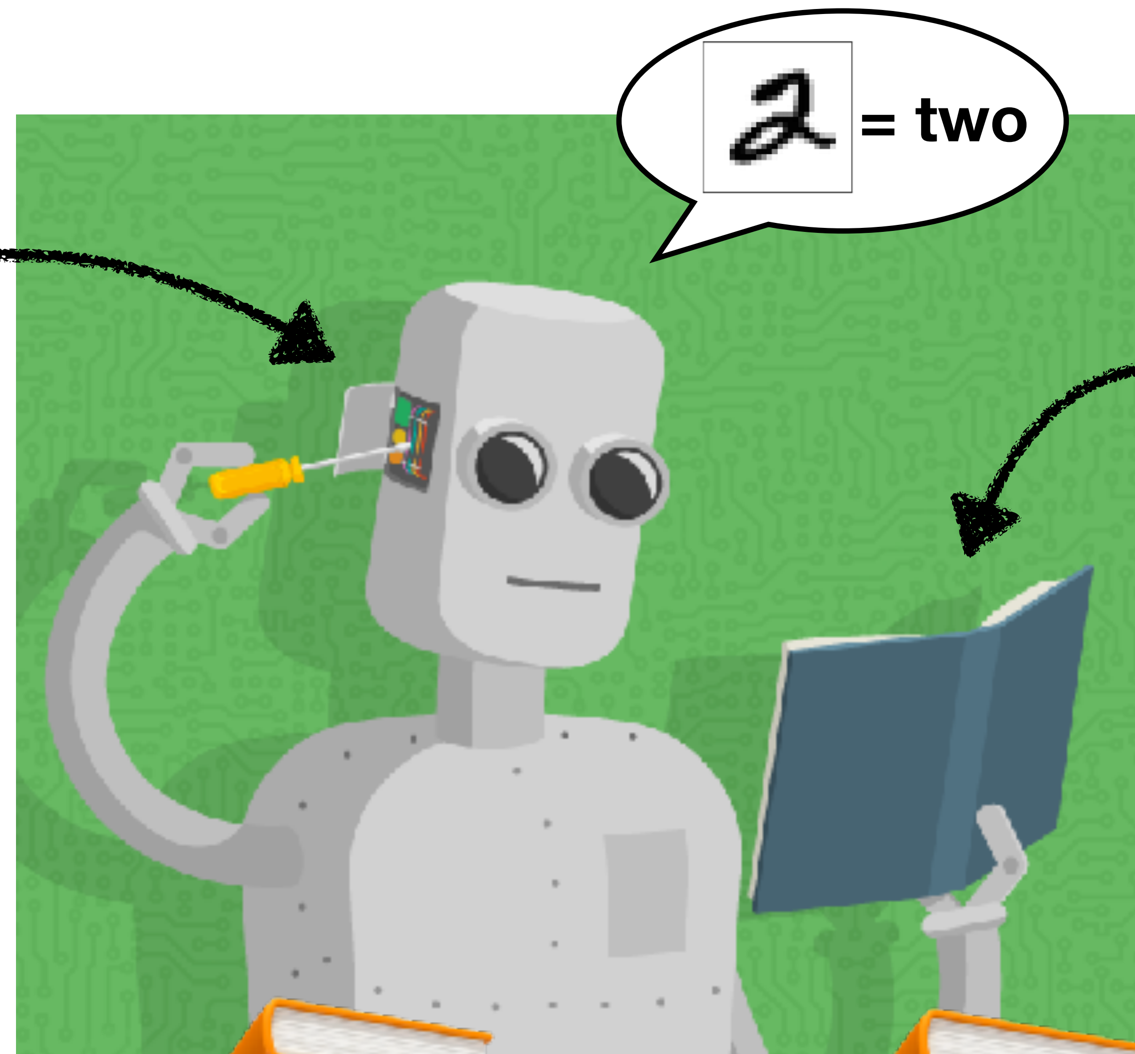
Context: machine learning

Solution: let the computer figure it out by itself!



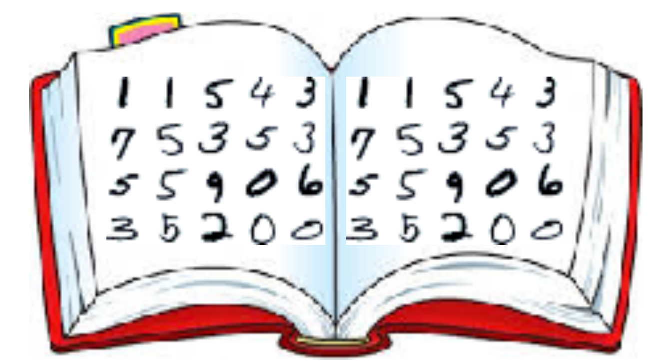
Requires a lot of training data (examples)

Context: machine learning

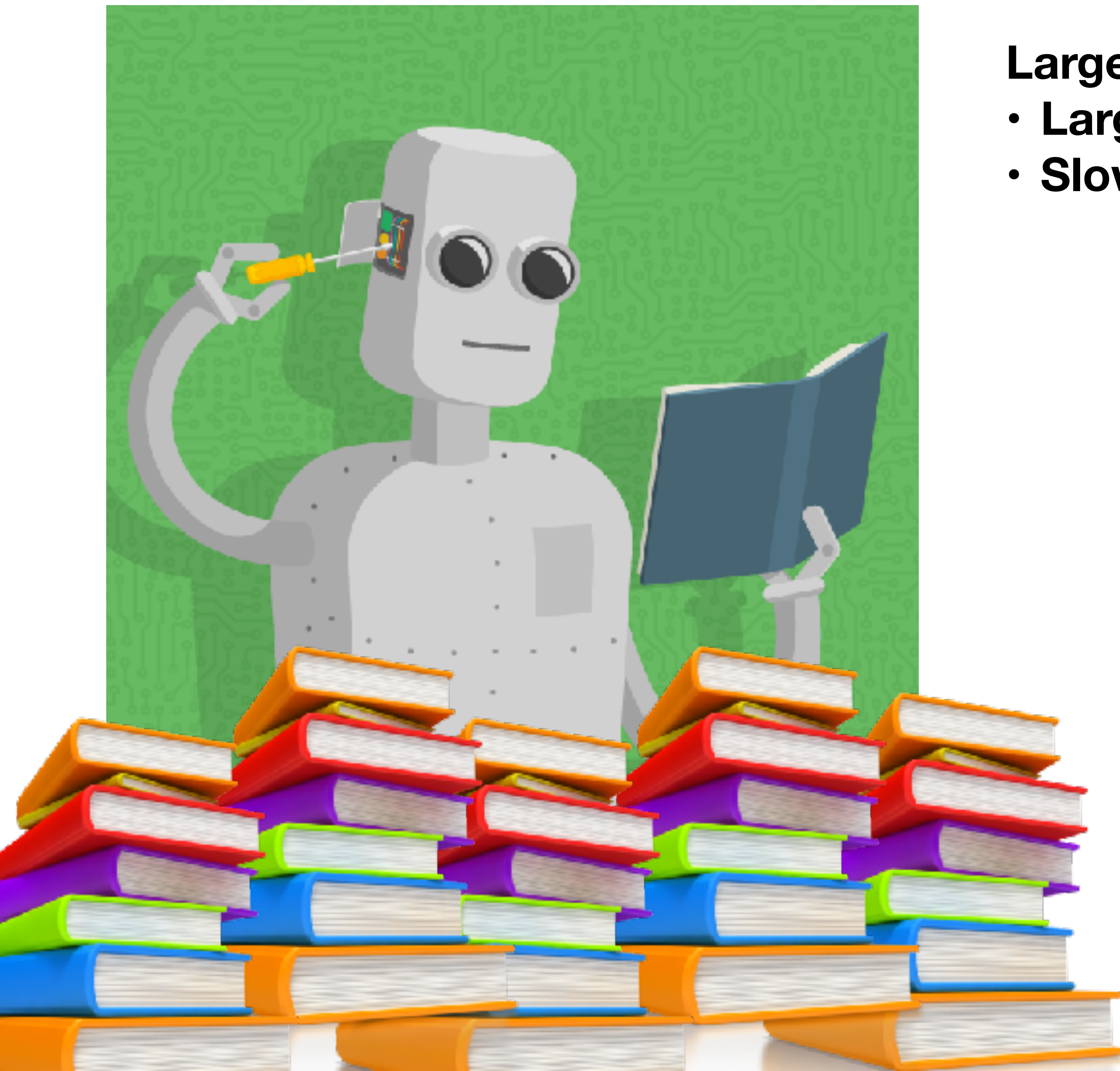


Solution: let the computer figure it out by itself!

Requires *a lot* of training data (examples)

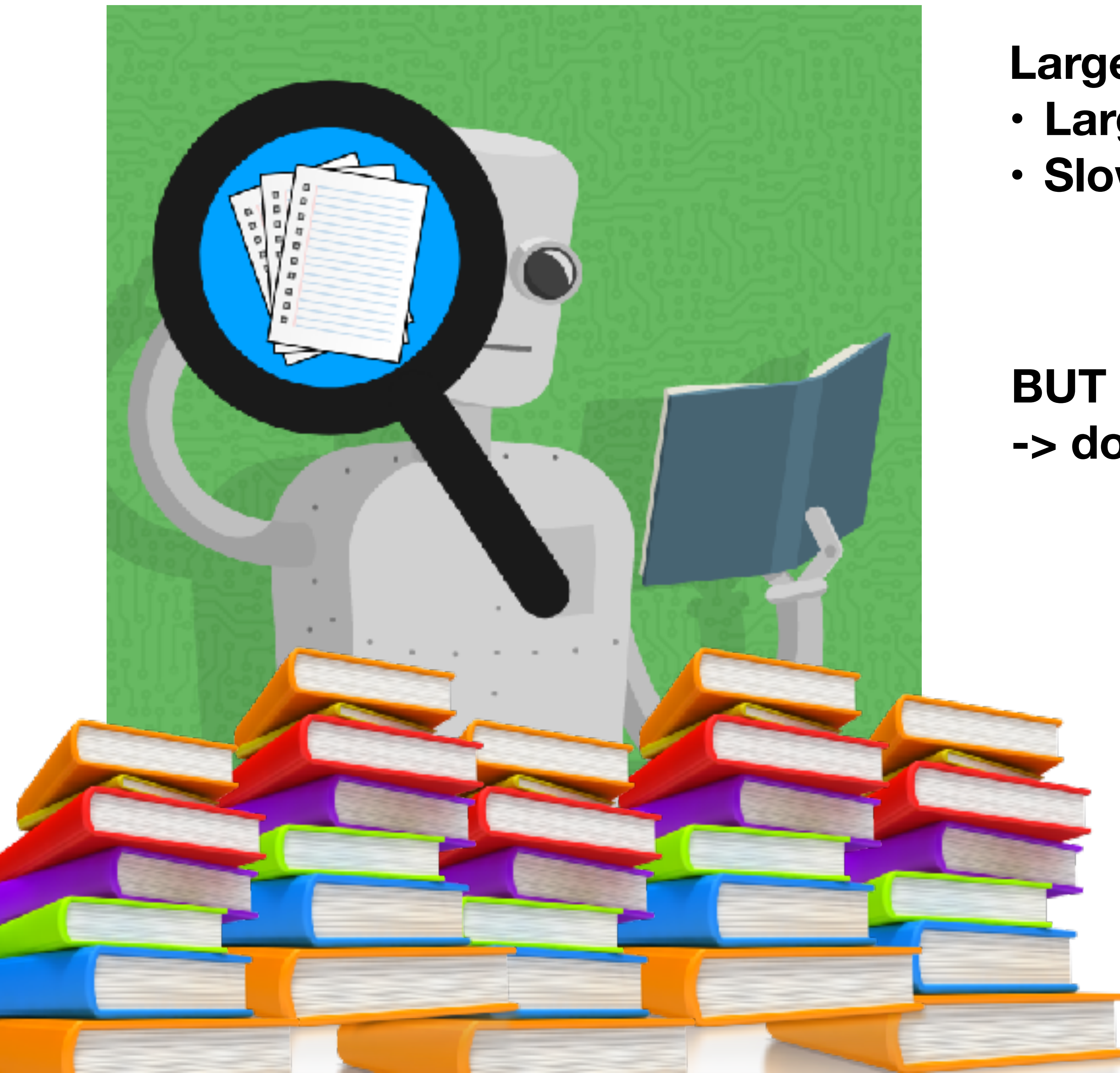


Machine learning limitations :-(



Large datasets means:
- **Large memory required**
- **Slow learning algorithm**

Machine learning limitations :-(

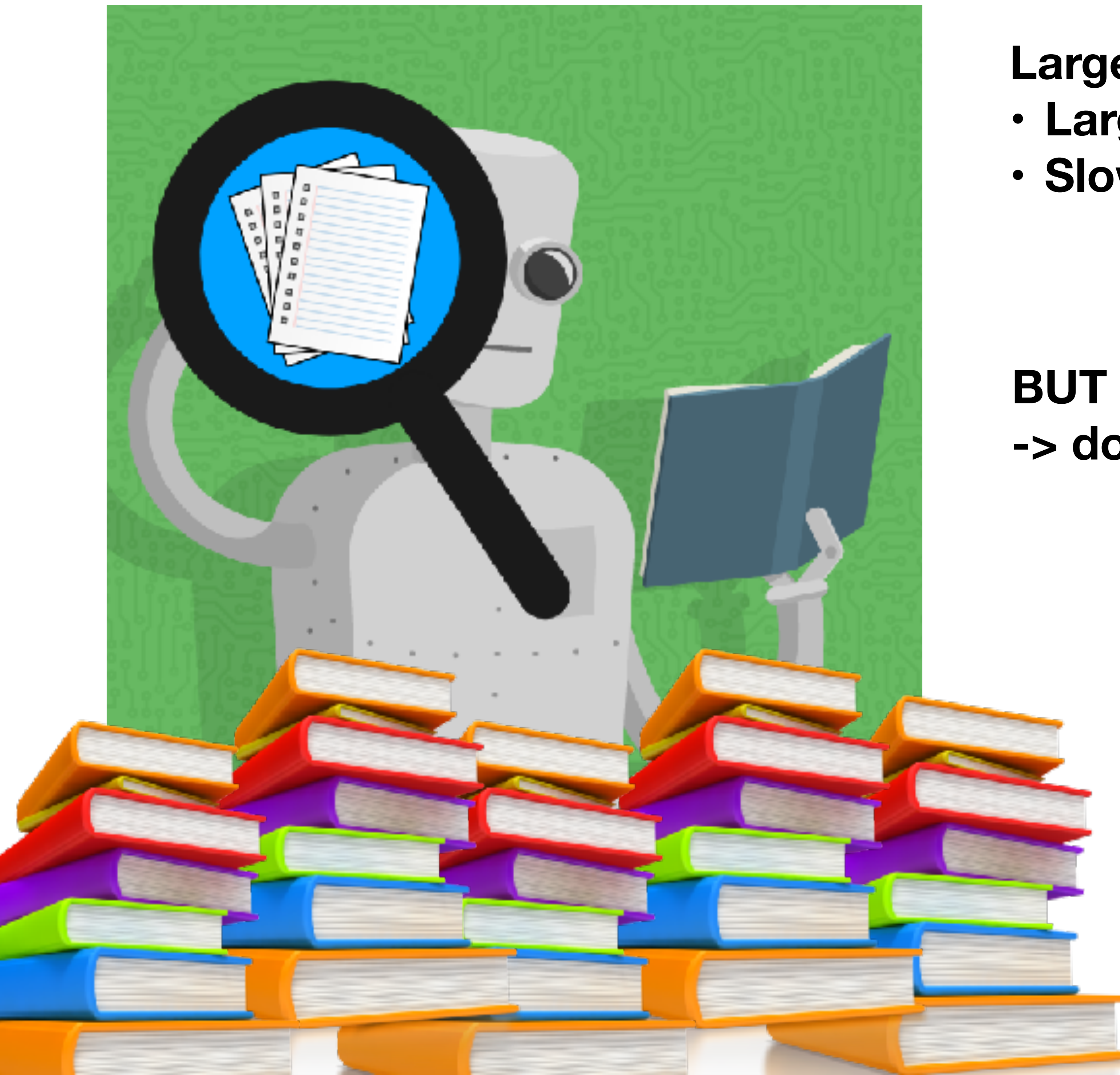


Large datasets means:

- **Large memory required**
- **Slow learning algorithm**

BUT extracted “knowledge” is “simple”
-> do we really need all this data?

Machine learning limitations :-(



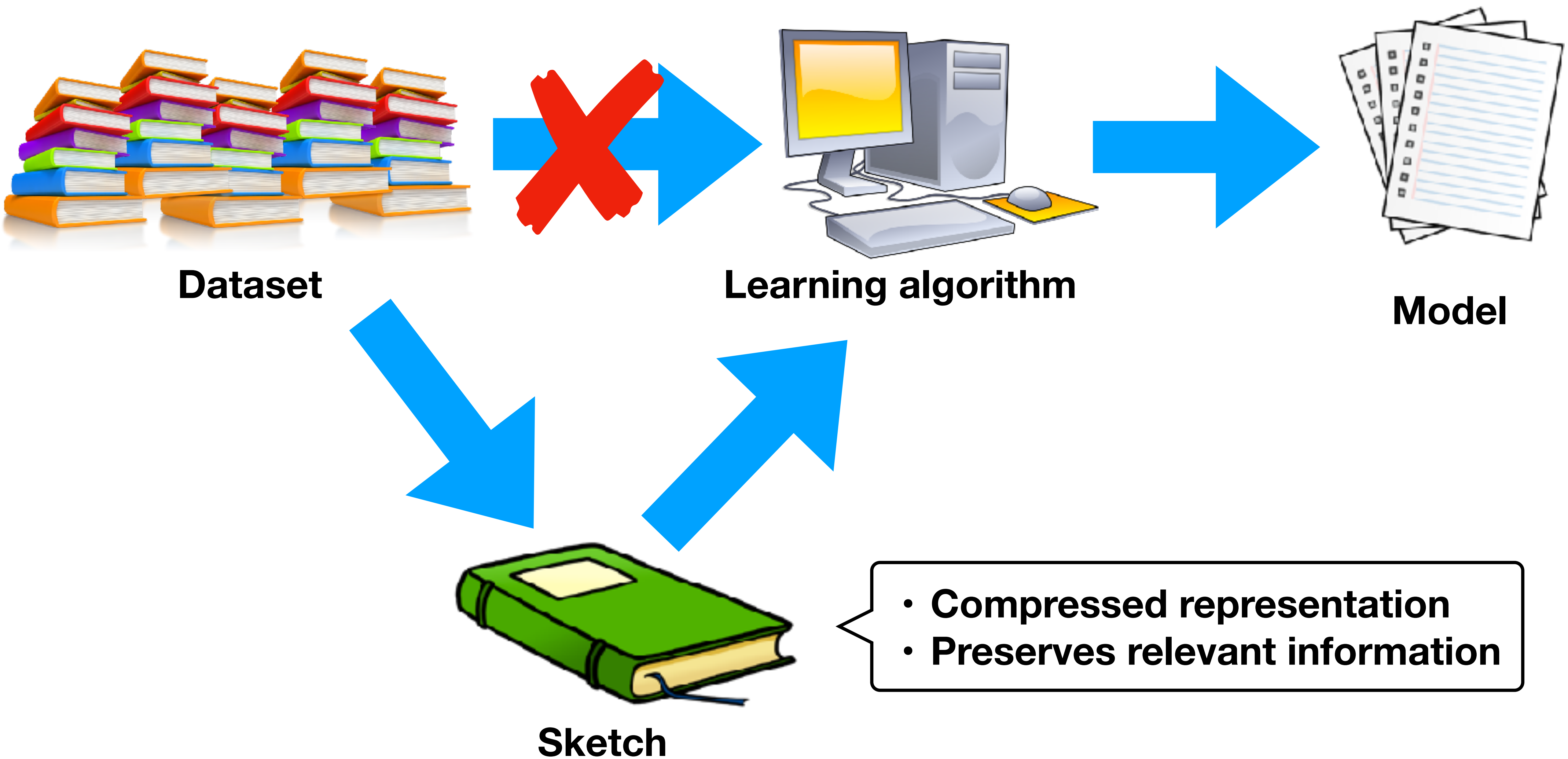
Large datasets means:
- **Large memory required**
- **Slow learning algorithm**

BUT extracted "knowledge" is "simple"
-> do we really need all this data?

NO!

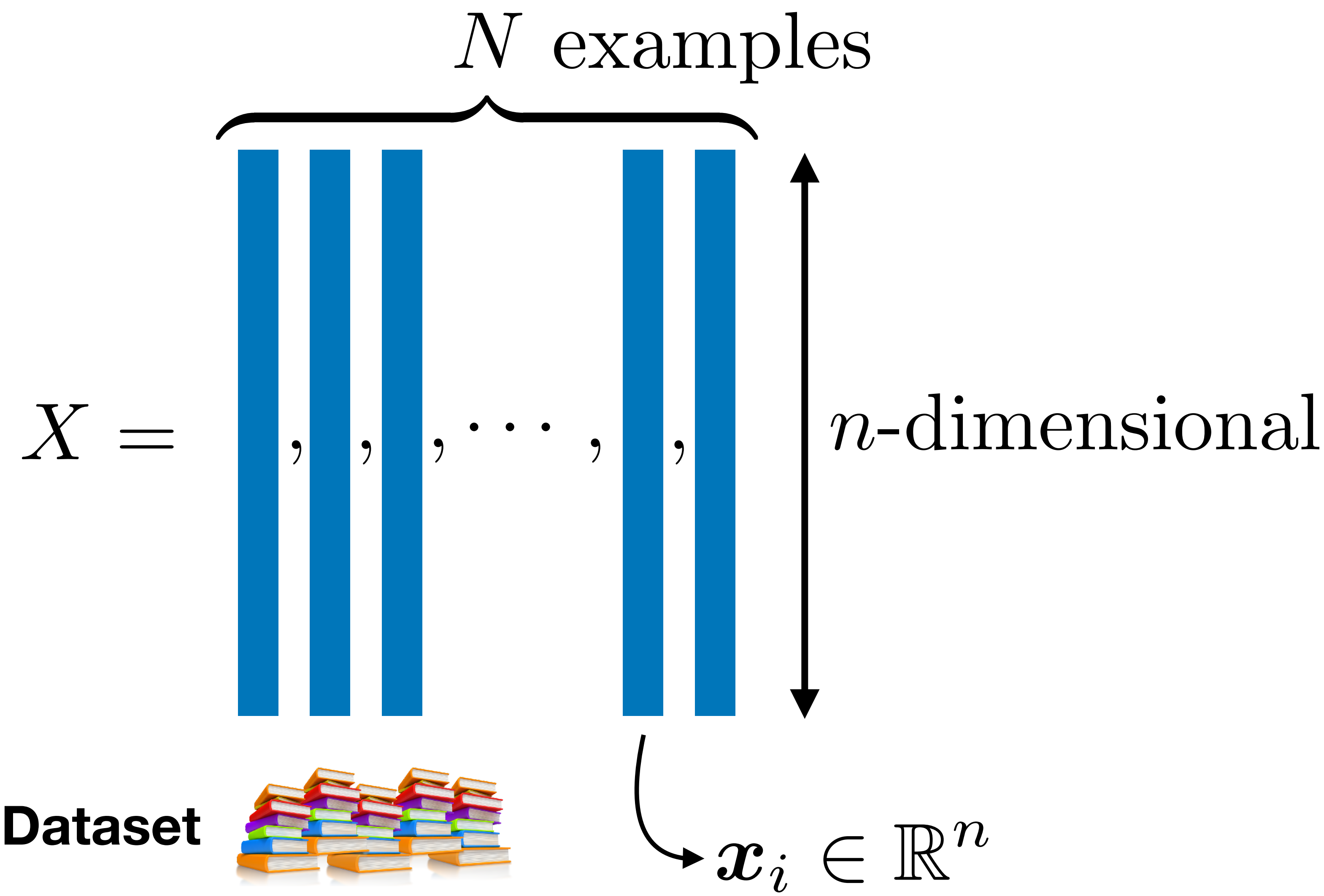
(otherwise this talk would be finished)

Compressive learning (from a *sketch*)

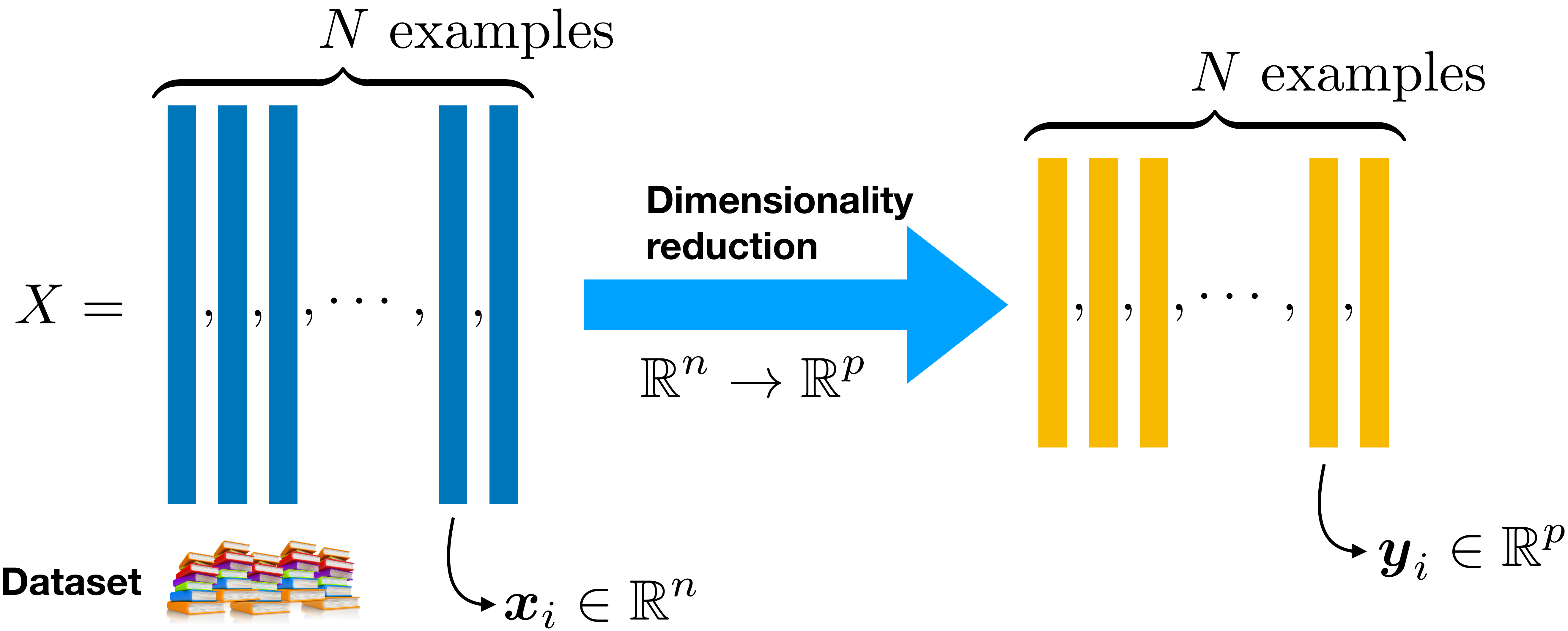


[Gribonval17]

Compressing a dataset?

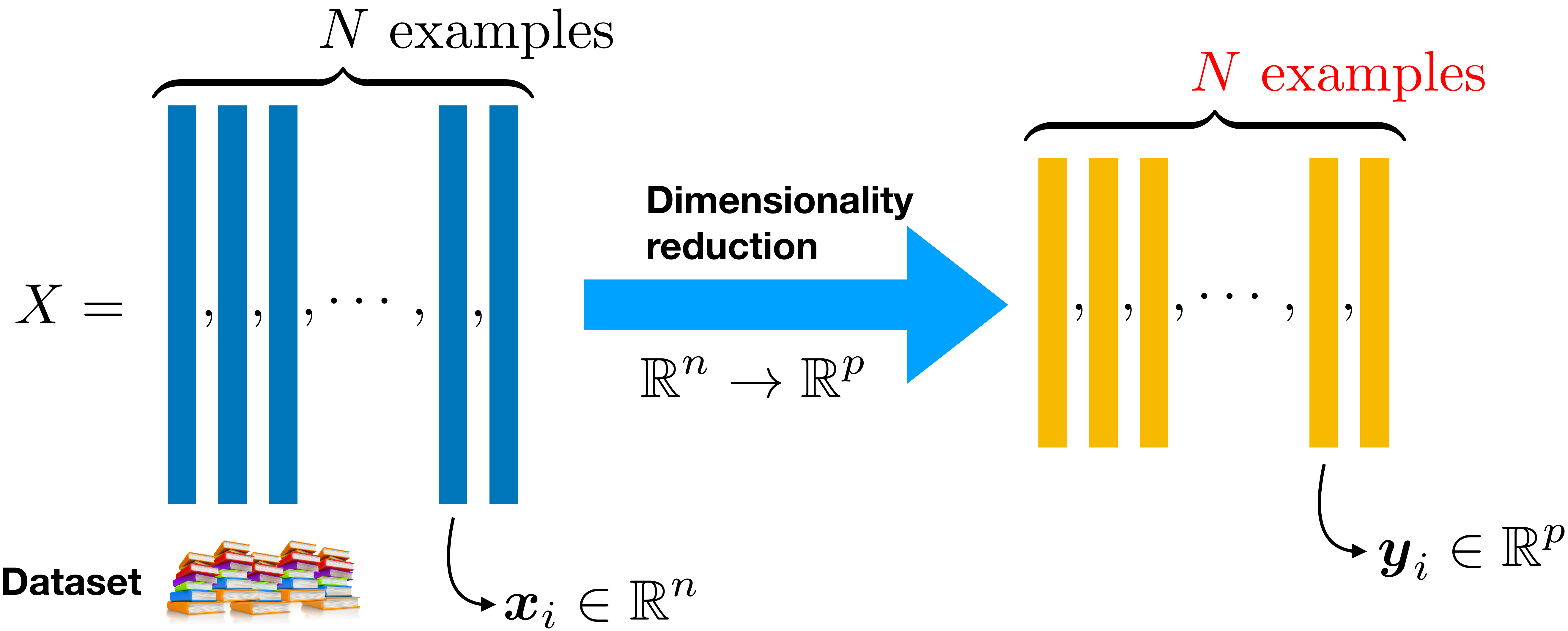


Compressing a dataset?



- **Compressed representation** ✓
- **Preserves relevant information** ✓

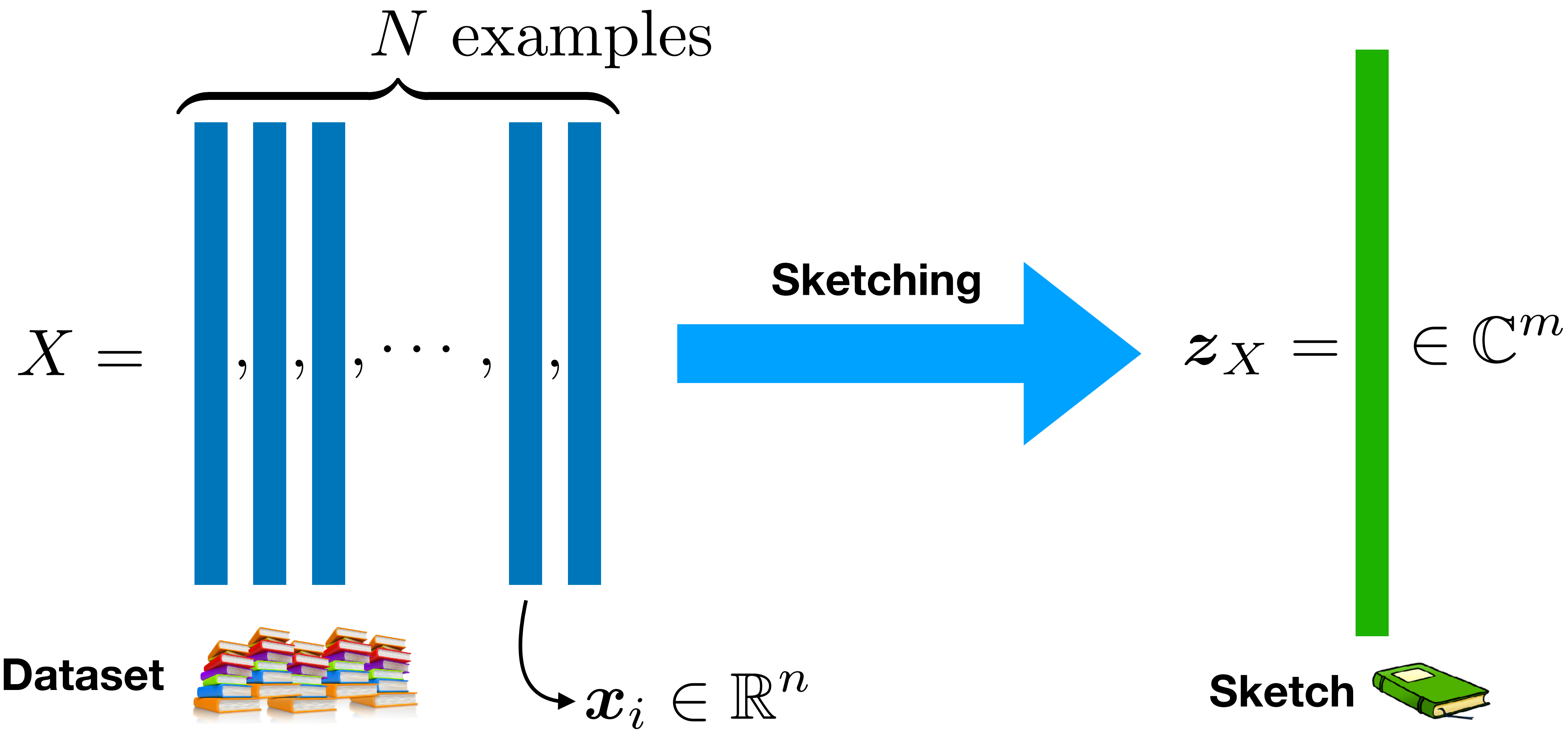
Compressing a dataset?



- Compressed representation ✓
- Preserves relevant information ✓
- Constant number of examples ✗

N can be VERY large ("big data")!

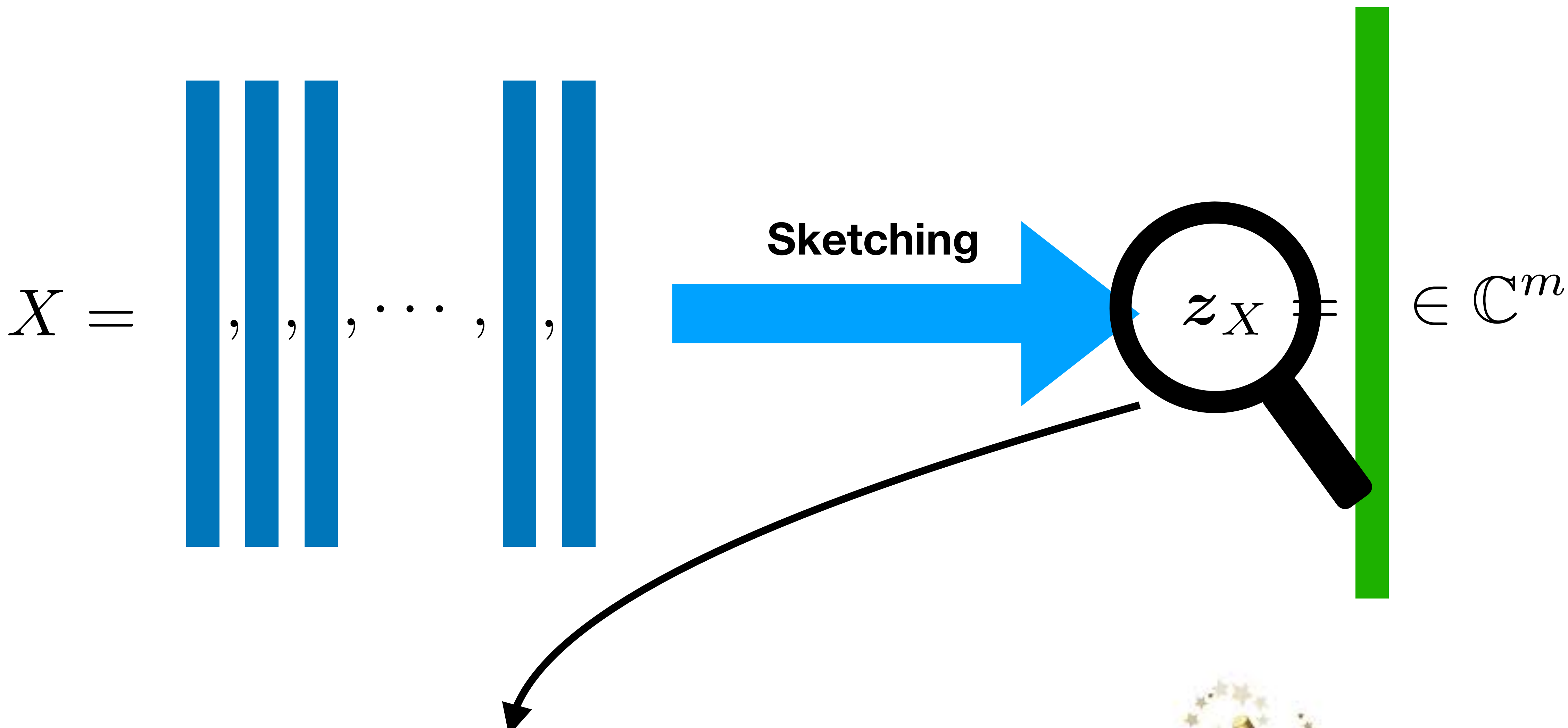
Compressing a dataset!



- **Compressed representation** ✓
- **Preserves relevant information** ✓
- **Dataset summary = single vector** ✓

[Gribonval17]

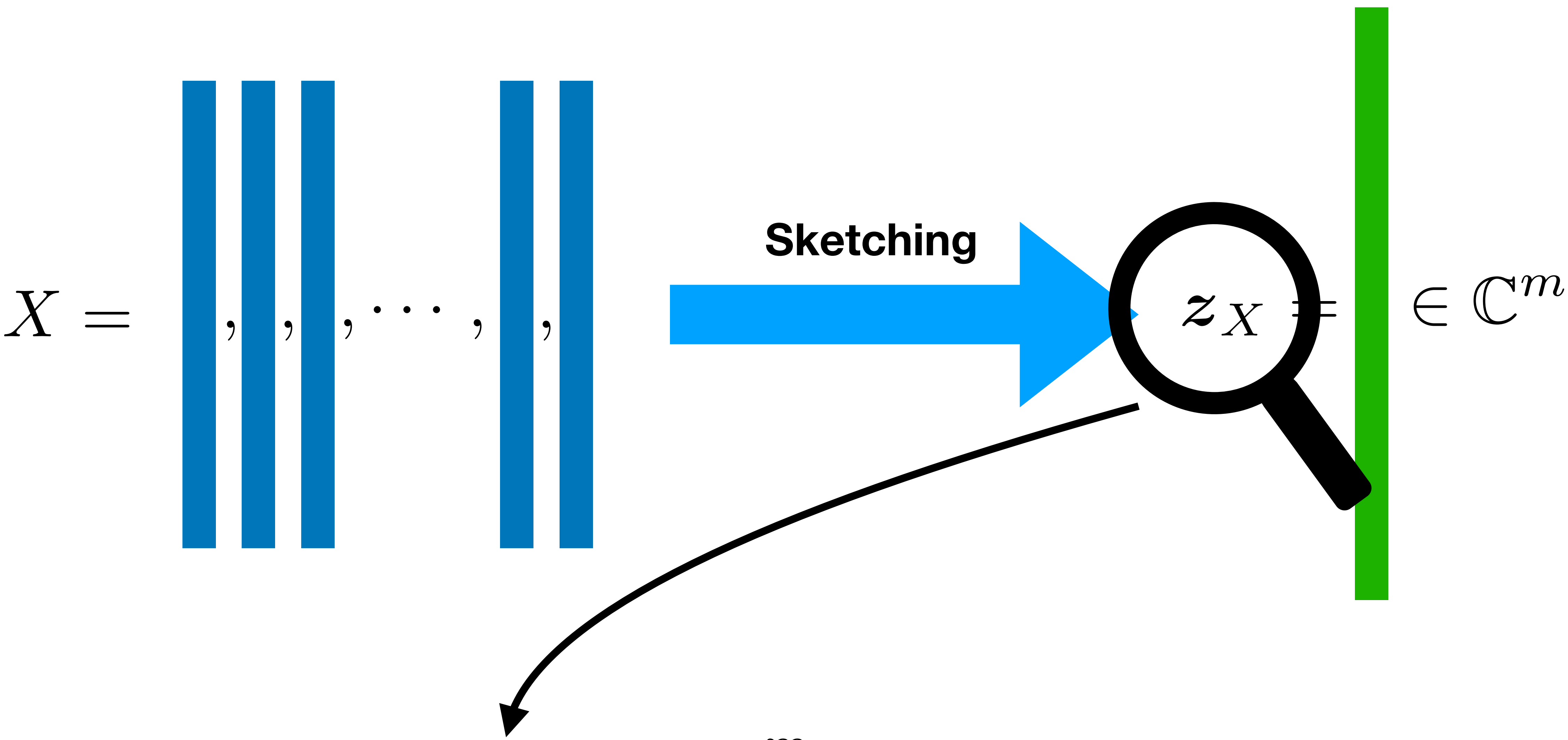
Sketch of a dataset



$$\boldsymbol{z}_X = \left[\frac{1}{N} \sum_{\boldsymbol{x}_i \in X} e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}_i} \right]_{j=1}^{m}$$

???

Sketch of a dataset

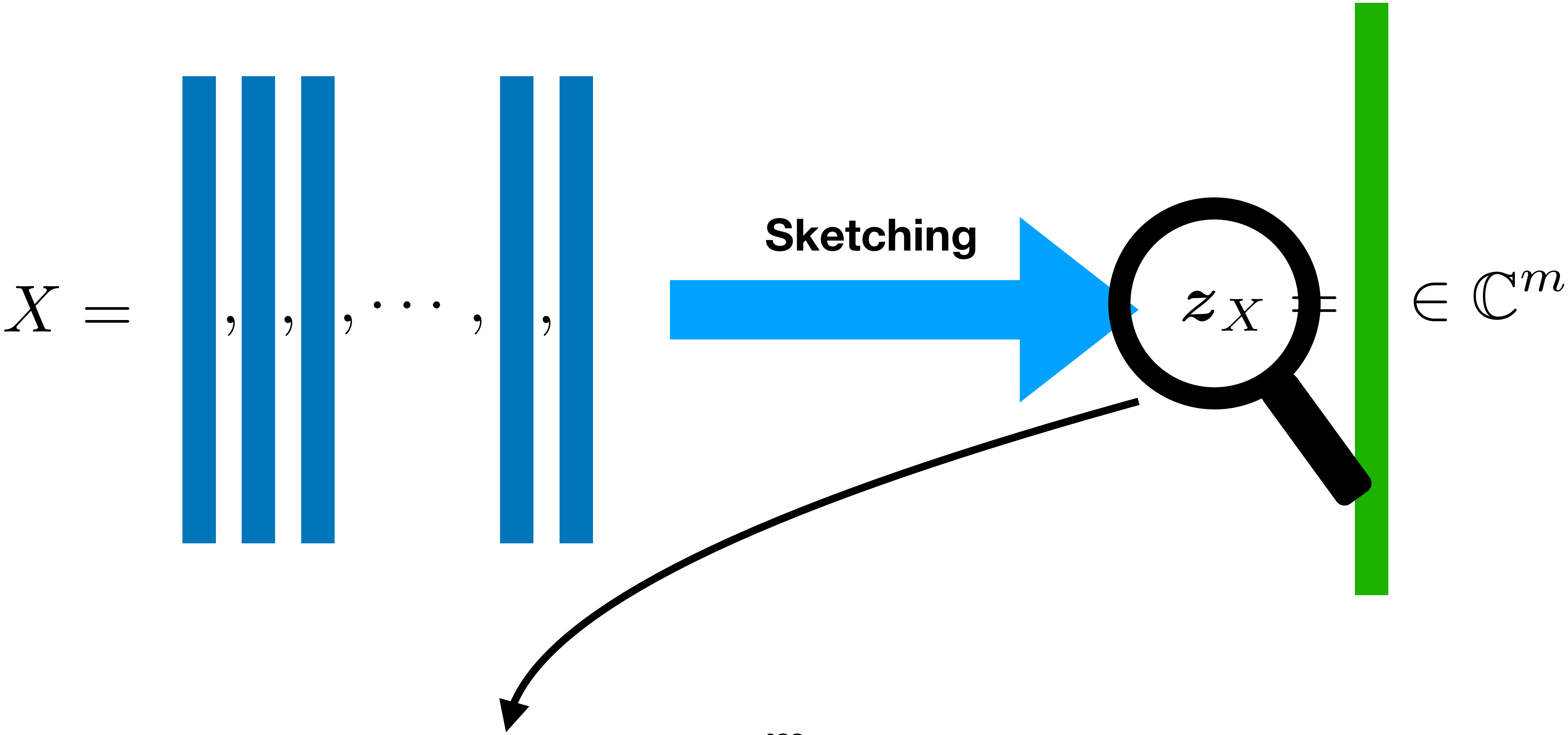


$$\boldsymbol{z}_X = \left[\frac{1}{N} \sum_{\boldsymbol{x}_i \in X} e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}_i} \right]_{j=1}^{m}$$

1. Project on (random) vectors

$\boldsymbol{\omega}_j \sim \Lambda$ **(cfr. later)**

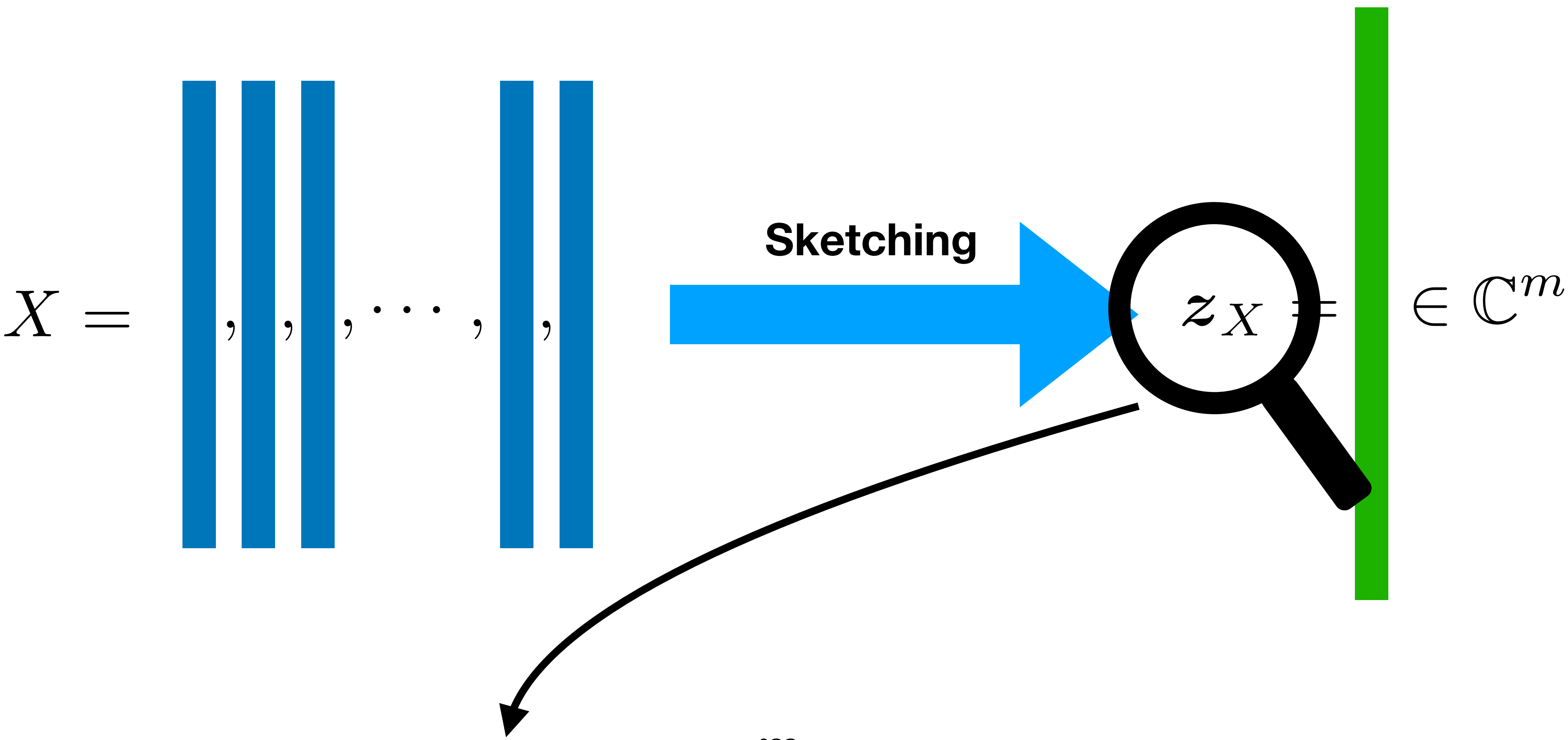
Sketch of a dataset



$$\boldsymbol{z}_X = \left[\frac{1}{N} \sum_{\boldsymbol{x}_i \in X} e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}_i} \right]_{j=1}^{m}$$

1. **Project on (random) vectors**
2. **Nonlinear periodic signature function**

Sketch of a dataset



$$\boldsymbol{z}_X = \left[\frac{1}{N} \sum_{\boldsymbol{x}_i \in X} e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}_i} \right]_{j=1}^{m}$$

1. Project on (random) vectors
2. Nonlinear periodic signature function
3. Pooling (average)

Sketch of a distribution

Sketching: an operator on probability distributions!

$$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T\boldsymbol{x}}\right]_{j=1}^{m}$$

Input: probability distribution

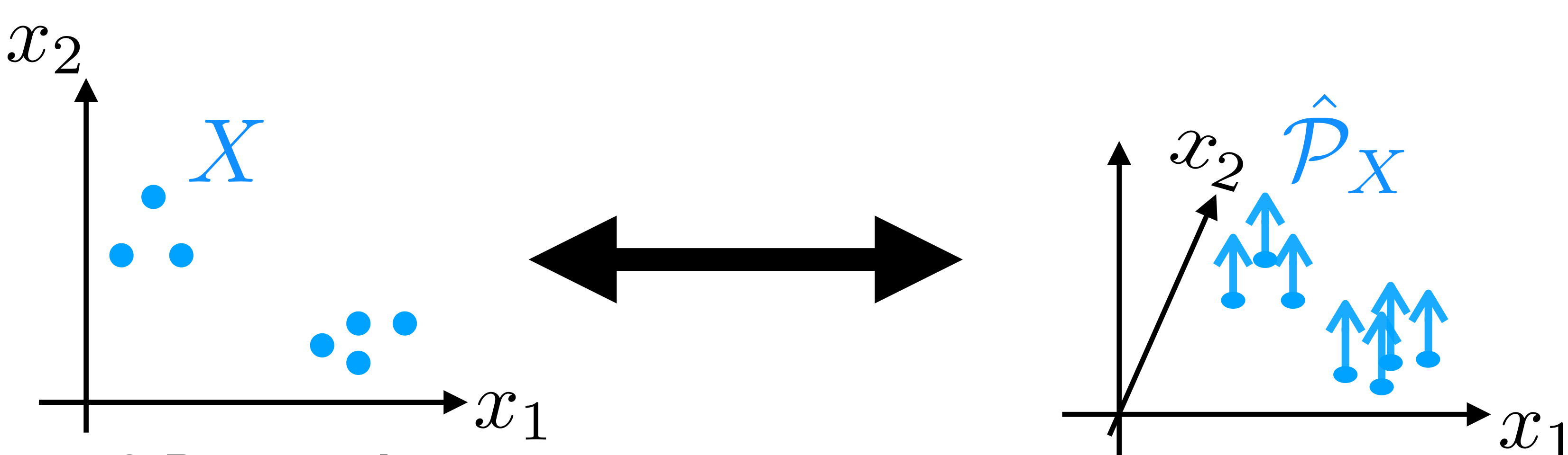
Output: m **moments of it**

Sketch of a distribution

Sketching: an operator on probability distributions!

$$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}} \right]_{j=1}^{m}$$

Particular case: dataset <-> empirical distribution $\boldsymbol{z}_X = \mathcal{A}(\hat{\mathcal{P}}_X)$

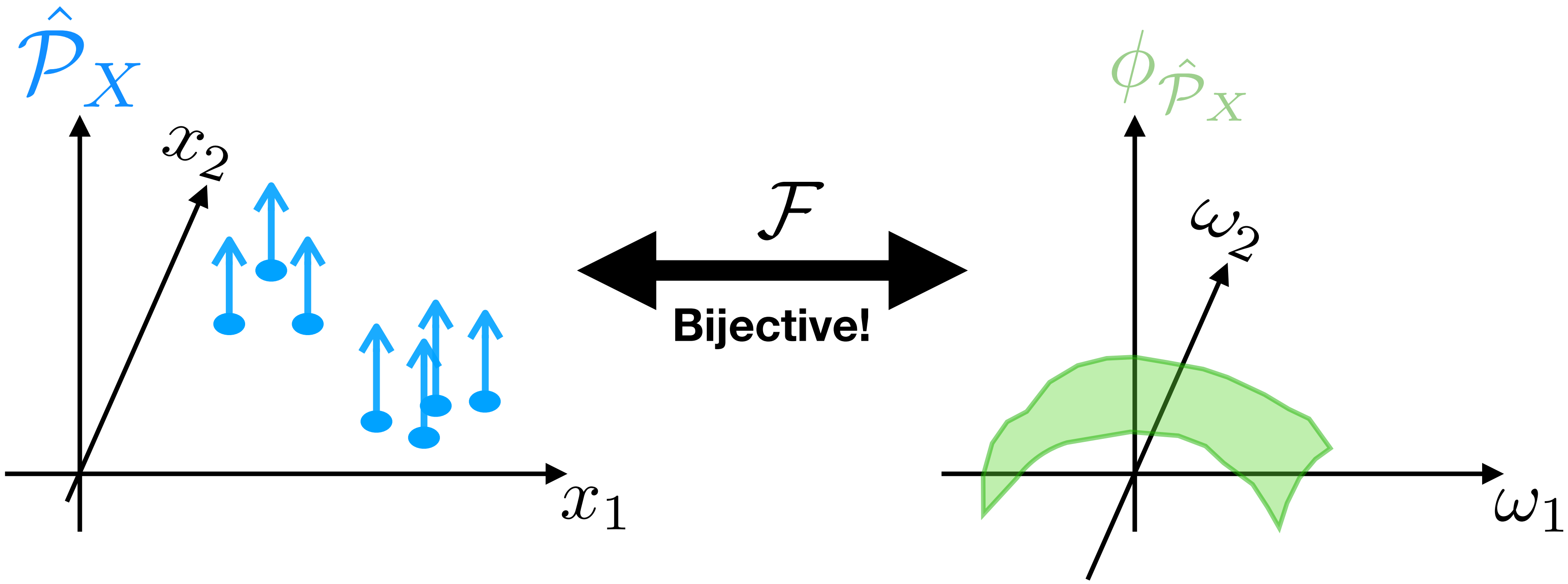


2-D example

$$\hat{\mathcal{P}}_X = \frac{1}{N} \sum_{\boldsymbol{x}_i \in X} \delta_{\boldsymbol{x}_i}$$

Sketch interpretation (1)

Sketch of P = *Random Fourier sampling* of P



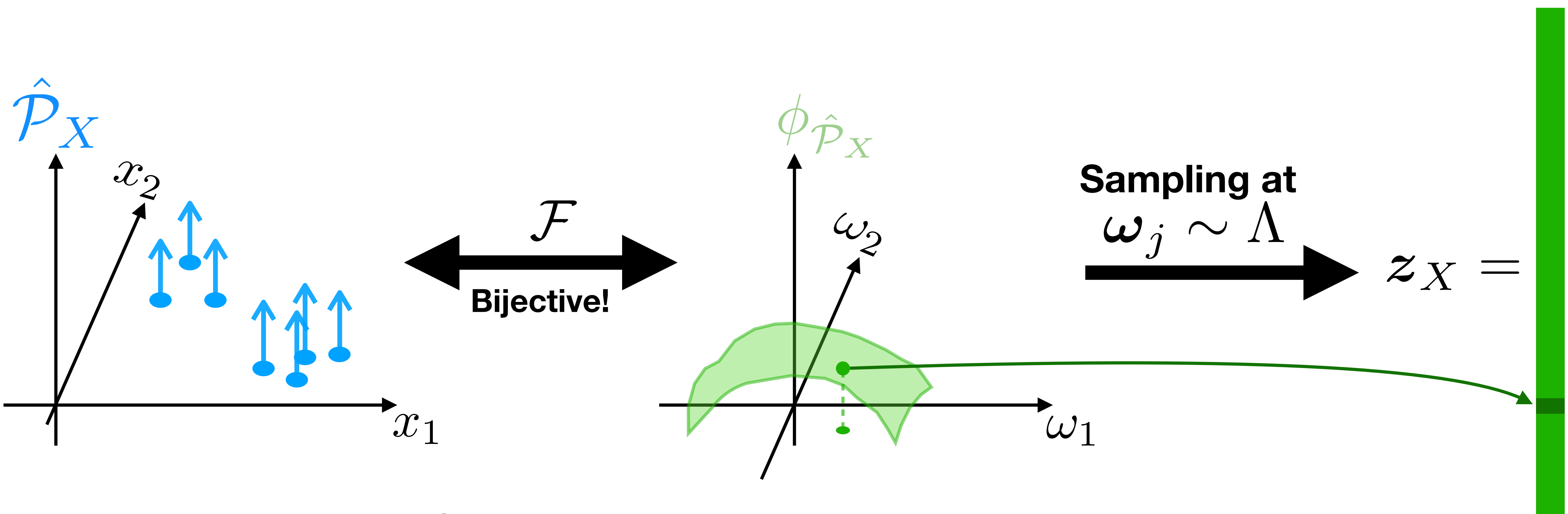
Characteristic function: characterises P
(who would have guessed?)

$$\phi_{\mathcal{P}}(\boldsymbol{\omega}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}}\, e^{-\mathrm{i}\boldsymbol{\omega}^T\boldsymbol{x}}$$

$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T\boldsymbol{x}}\right]_{j=1}^{m}$

Sketch interpretation (1)

Sketch of P = *Random Fourier sampling* of P $\mathcal{A}(\mathcal{P})_j = \phi_{\mathcal{P}}(\boldsymbol{\omega}_j)$



Characteristic function: characterises P

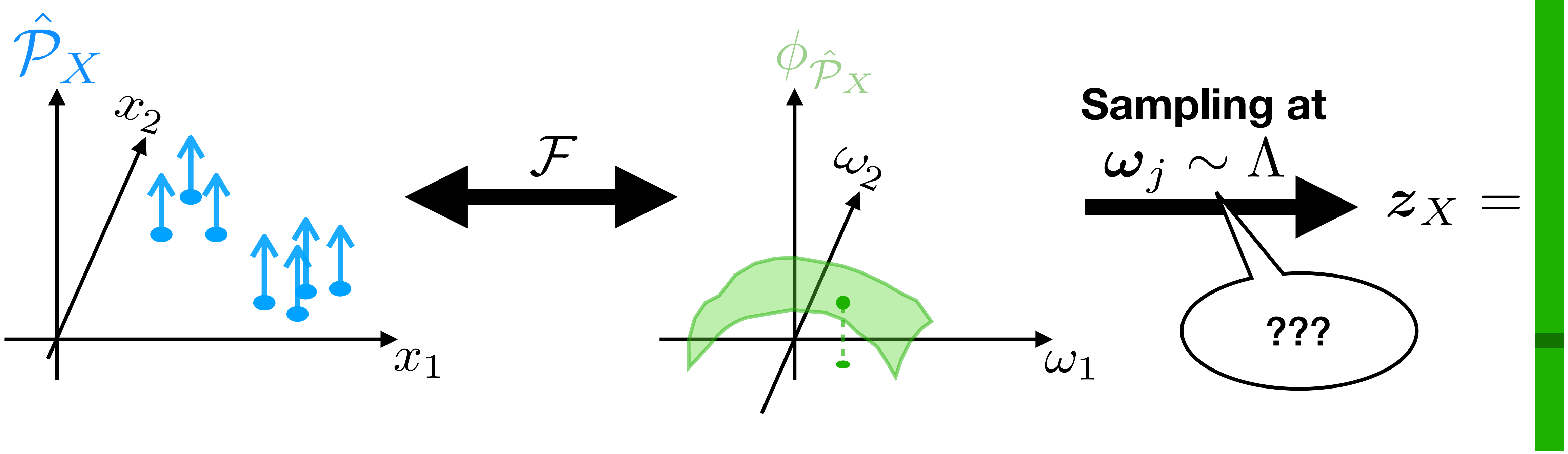
(who would have guessed?)

$$\phi_{\mathcal{P}}(\boldsymbol{\omega}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}}\, e^{-\mathrm{i}\boldsymbol{\omega}^T\boldsymbol{x}}$$

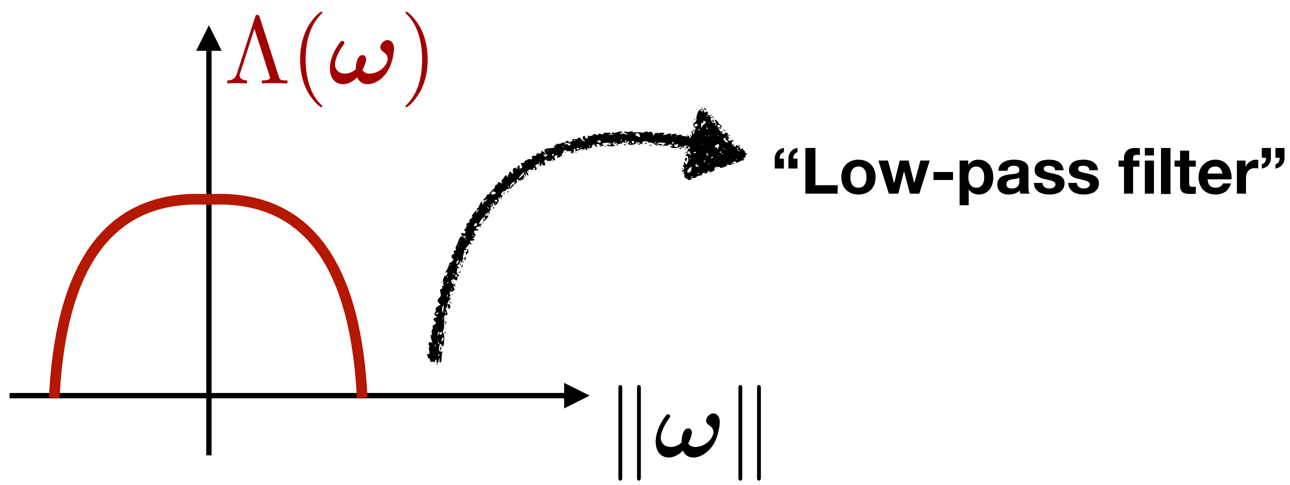
$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T\boldsymbol{x}}\right]_{j=1}^{m}$

Sketch interpretation (1)

Sketch of P = *Random Fourier sampling* of P $\mathcal{A}(\mathcal{P})_j = \phi_{\mathcal{P}}(\boldsymbol{\omega}_j)$



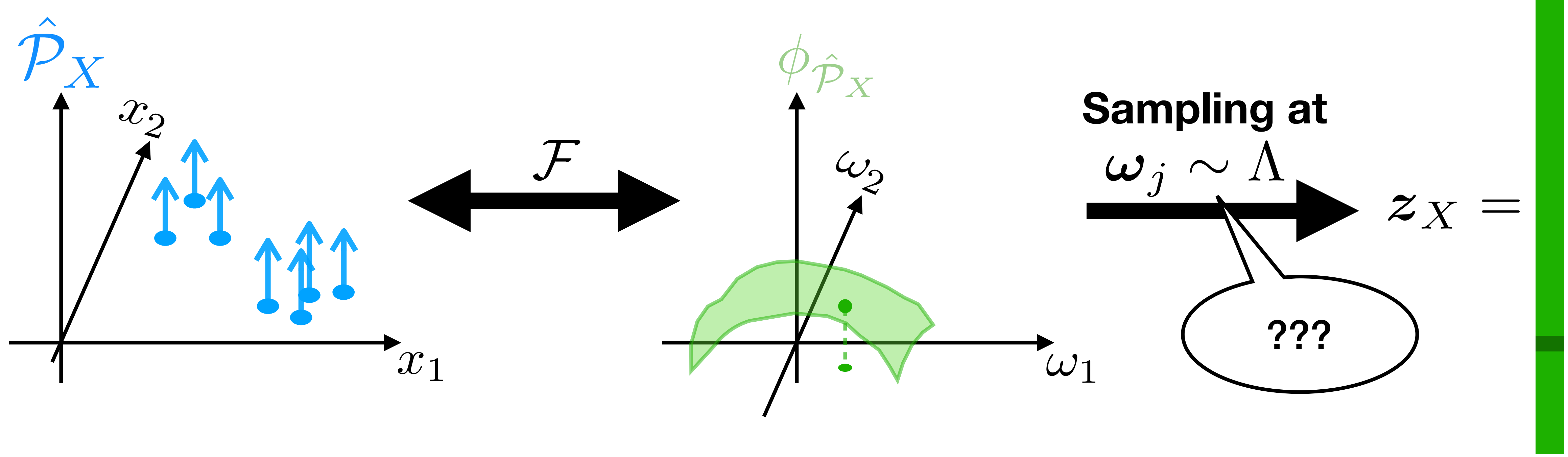
Typically:



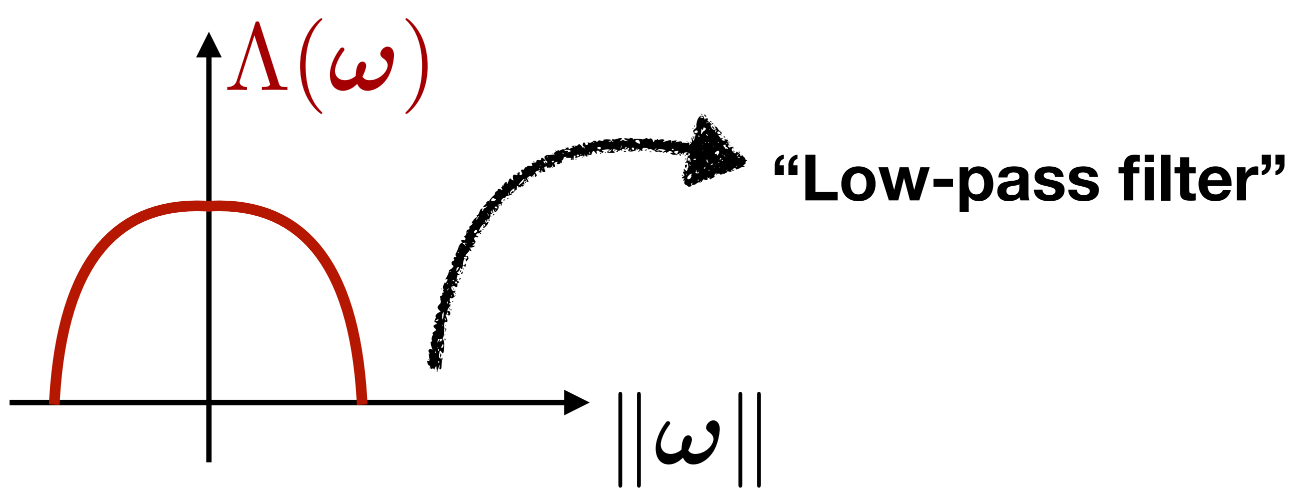
$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}}\right]_{j=1}^{m}$

Sketch interpretation (1)

Sketch of P = ***Random Fourier sampling*** **of P** $\mathcal{A}(\mathcal{P})_j = \phi_{\mathcal{P}}(\boldsymbol{\omega}_j)$



Typically:



In practice:

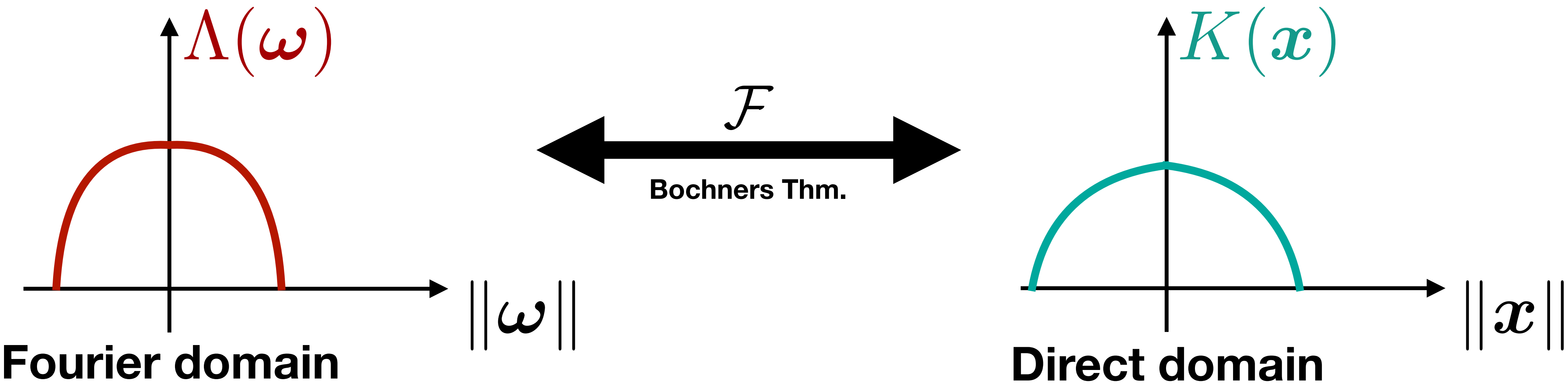
- **Application-dependent**
- **Requires some data**

$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}} \right]_{j=1}^m$

See also: Distilled sensing

Sketch interpretation (2)

Sketch of P = view P through *kernel* K : "Similarity measure"

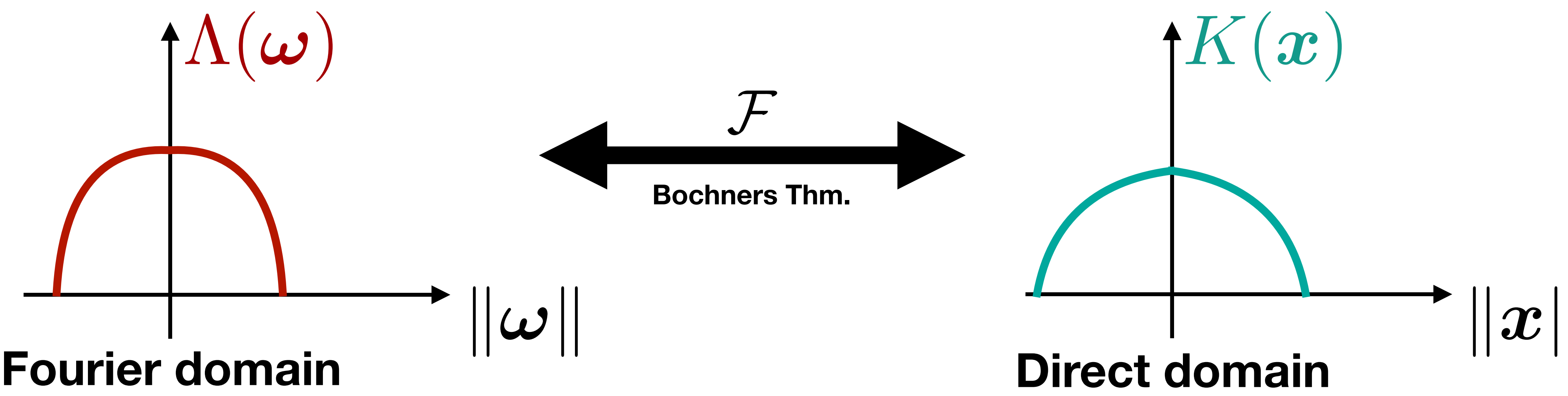


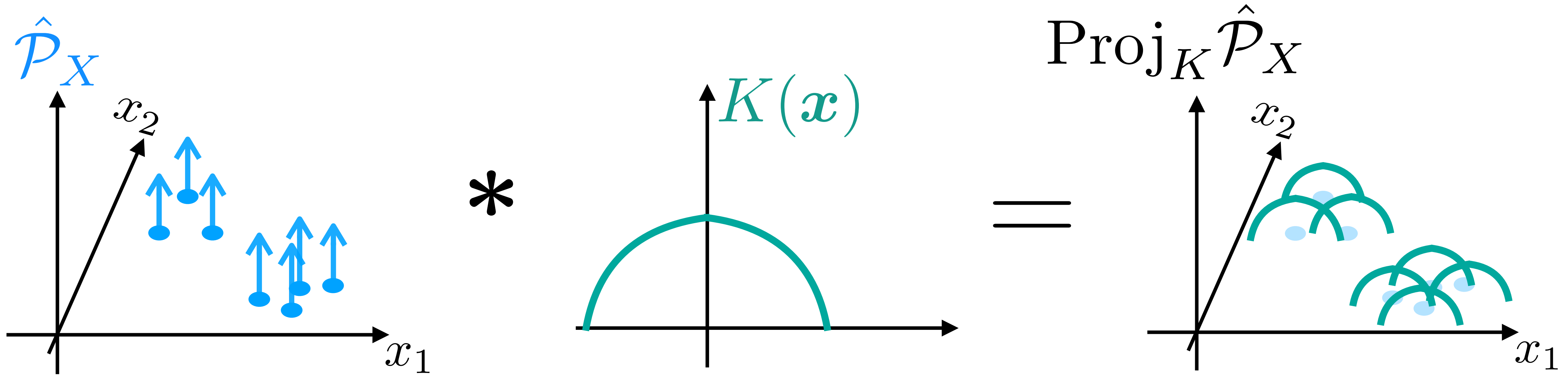
$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T\boldsymbol{x}}\right]_{j=1}^m$

[Rahimi08]

Sketch interpretation (2)

Sketch of P = view P through *kernel* K : “Similarity measure”

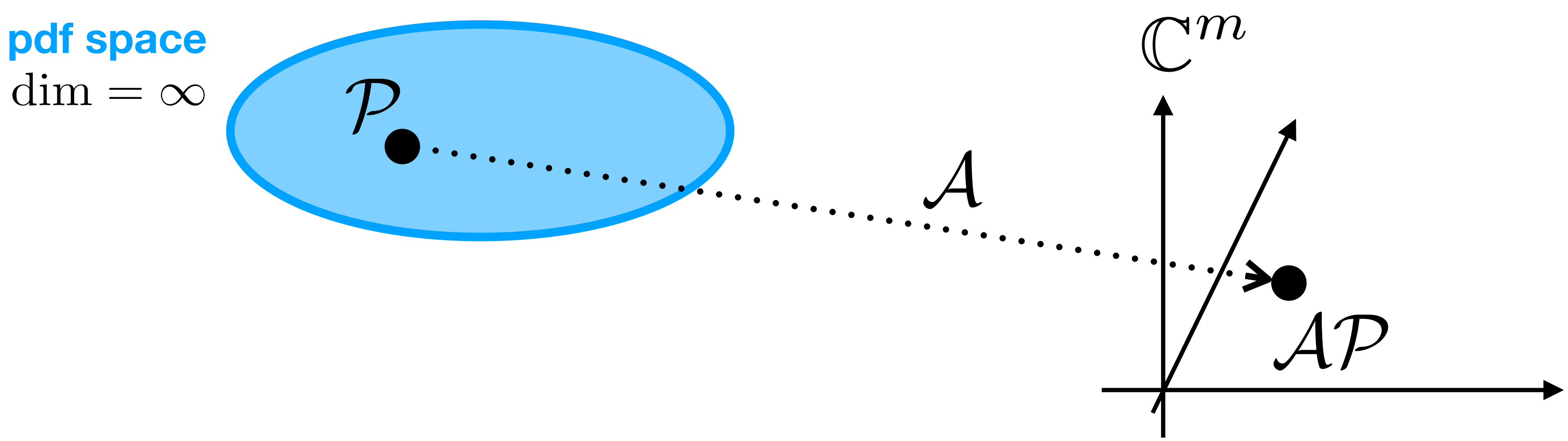




$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}} \right]_{j=1}^{m}$

Sketch interpretation (3)

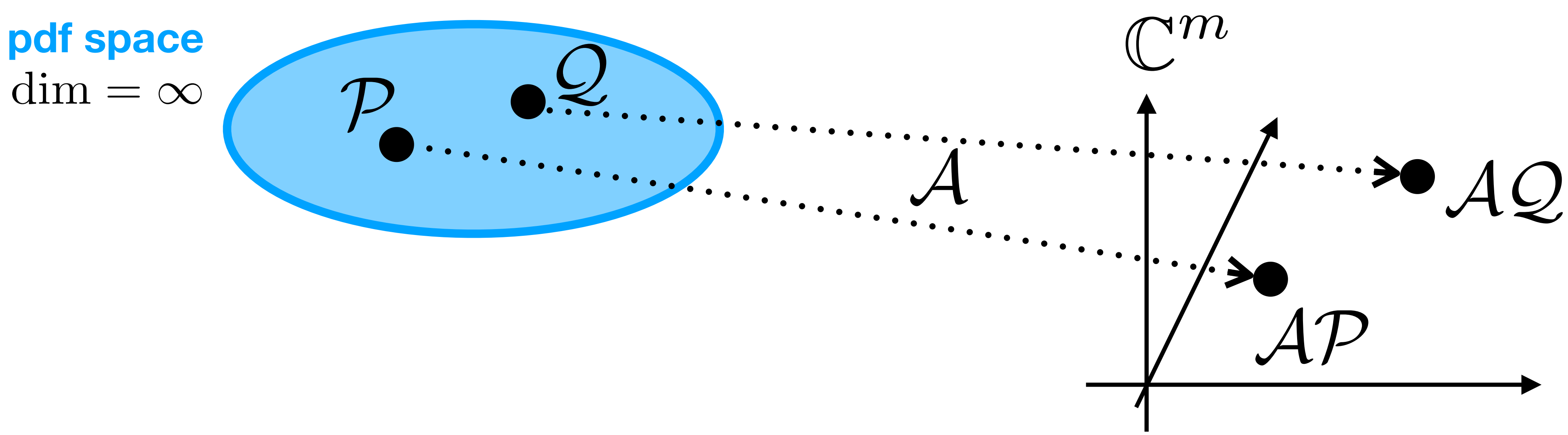
Sketch of P = low-dimensional embedding of P



$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}}\right]_{j=1}^{m}$

Sketch interpretation (3)

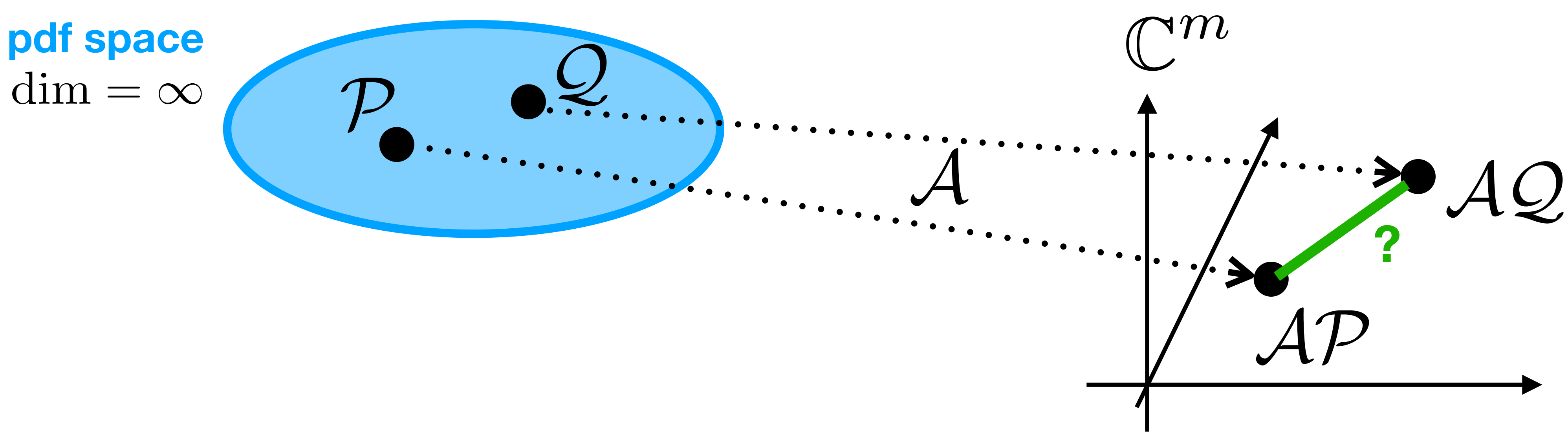
Sketch of P = low-dimensional embedding of P



$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T\boldsymbol{x}}\right]_{j=1}^m$

Sketch interpretation (3)

Sketch of P = low-dimensional embedding of P

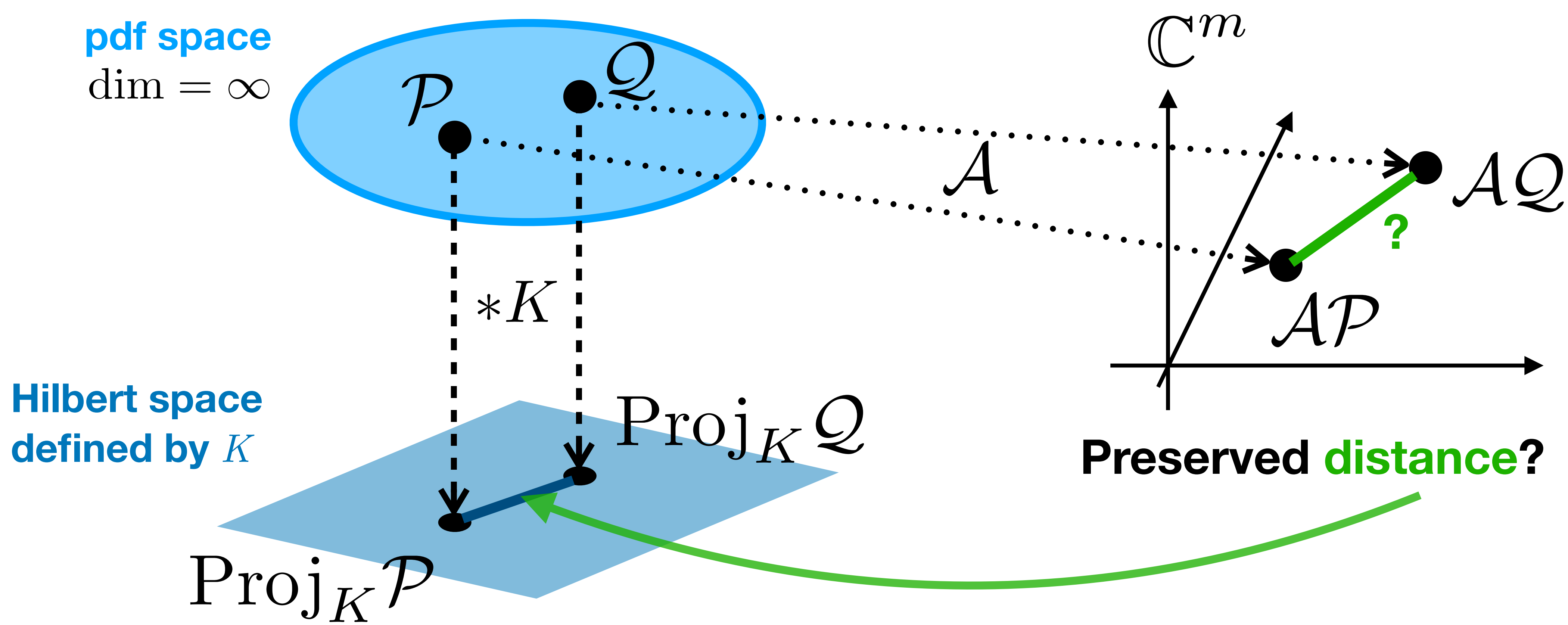


Preserved distance?

$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}}\right]_{j=1}^{m}$

Sketch interpretation (3)

Sketch of P = low-dimensional embedding of P

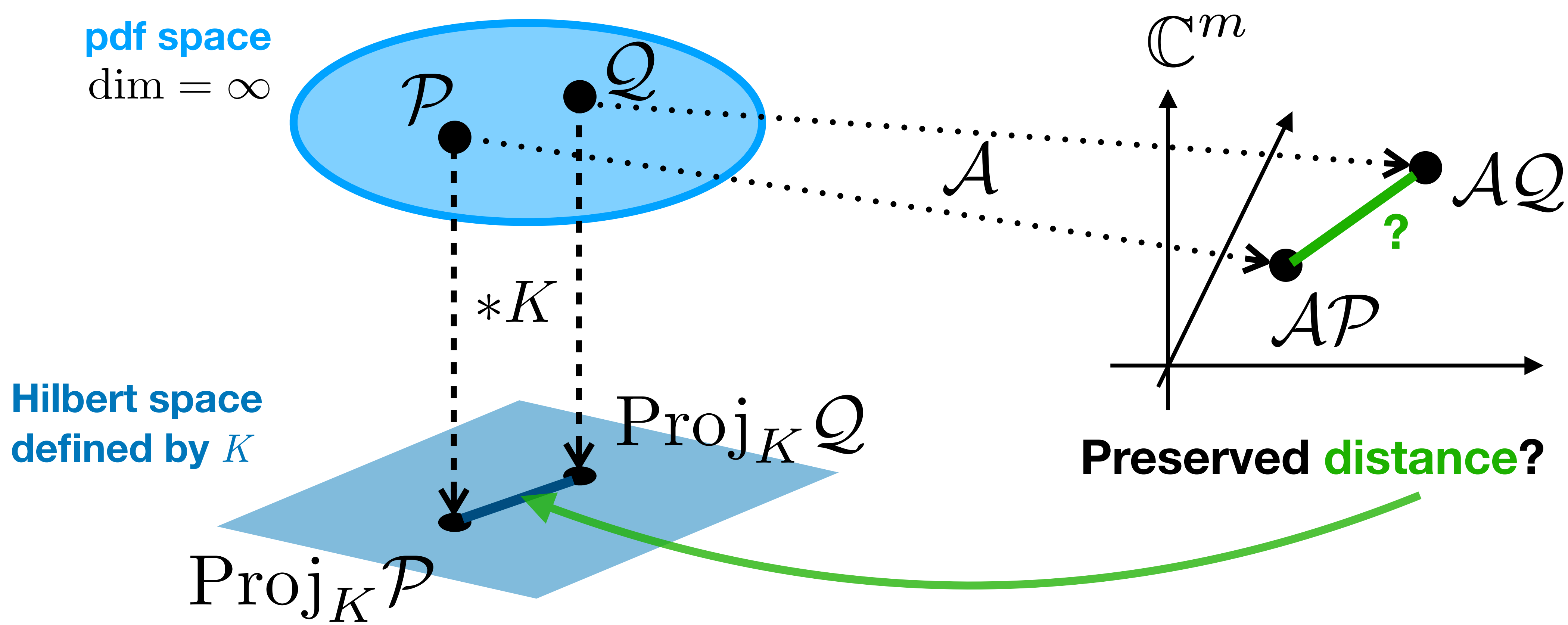


$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}} \right]_{j=1}^{m}$

[Sriperumbudur11]

Sketch interpretation (3)

Sketch of P = low-dimensional embedding of P



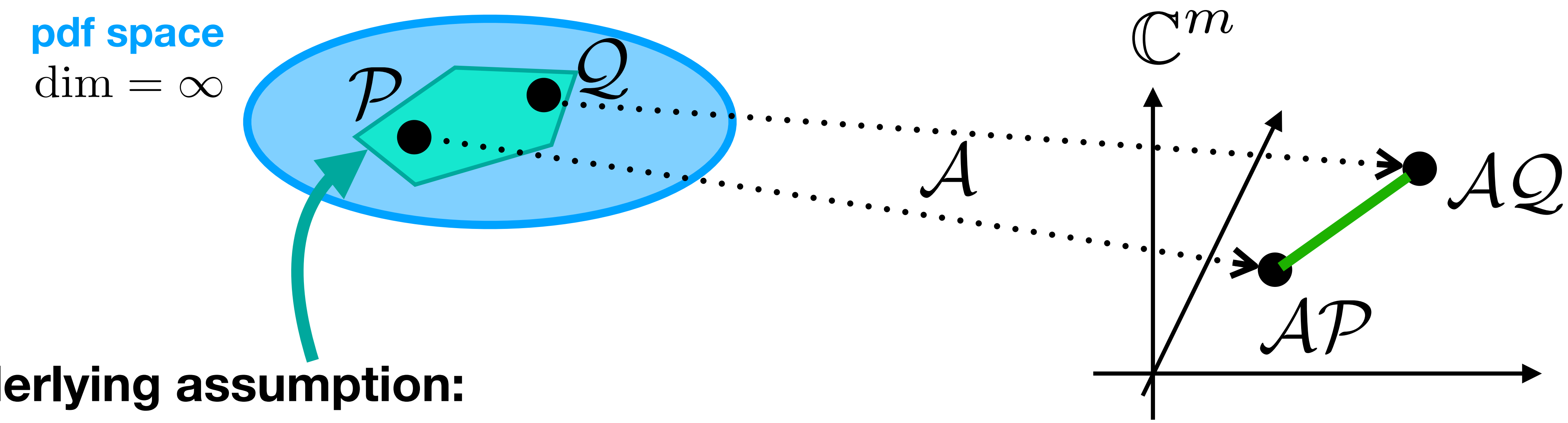
Infinite-dimensional Compressed Sensing!

$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}}\right]_{j=1}^m$

[Keriven16-GMM]

Sketch interpretation (3)

Sketch of P = low-dimensional embedding of P



Underlying assumption:

$$\mathcal{P} \in (\text{insert sparse set here})$$

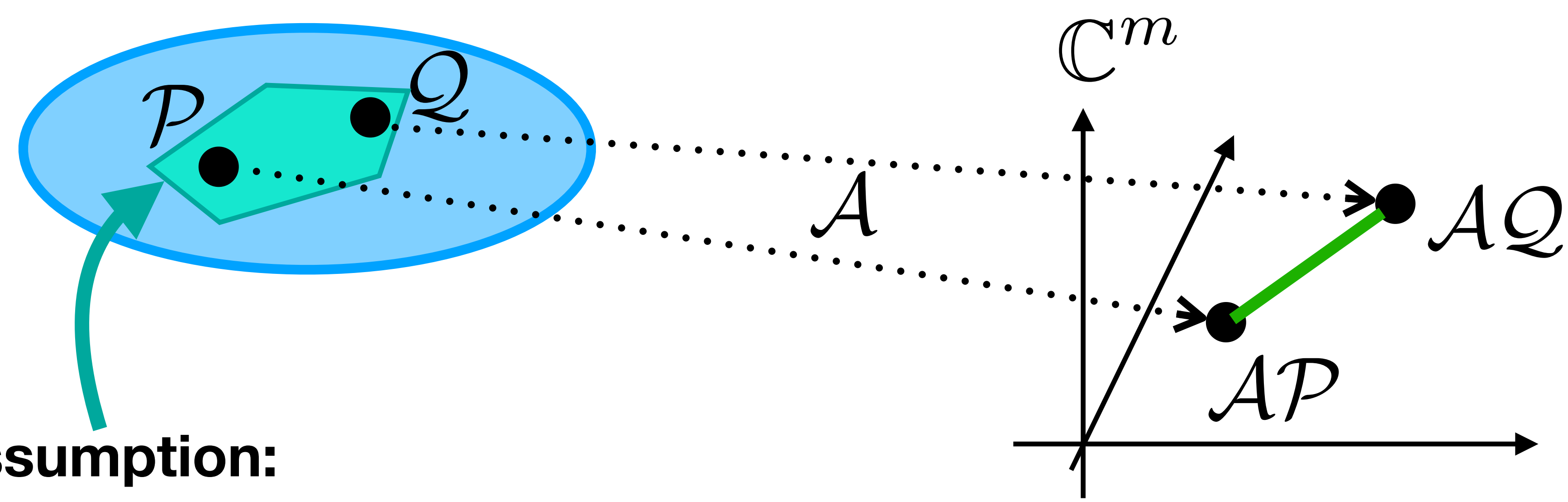
Infinite-dimensional Compressed Sensing!

$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}} \right]_{j=1}^{m}$

Sketch interpretation (3)

Sketch of P = low-dimensional embedding of P

pdf space
$\dim = \infty$



Underlying assumption:

$$\mathcal{P} \in (\text{insert sparse set here})$$

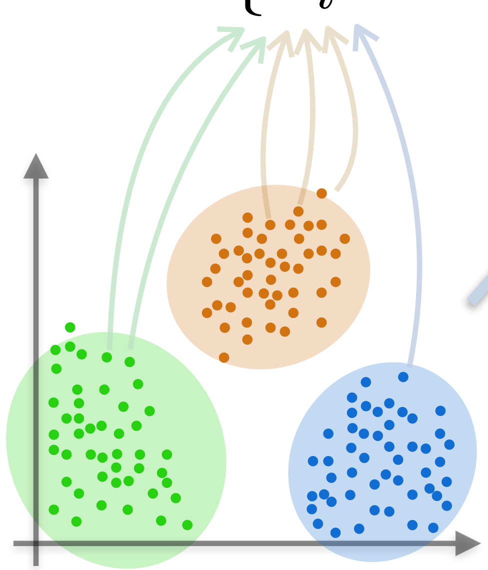
Defined by the application

Infinite-dimensional Compressed Sensing!

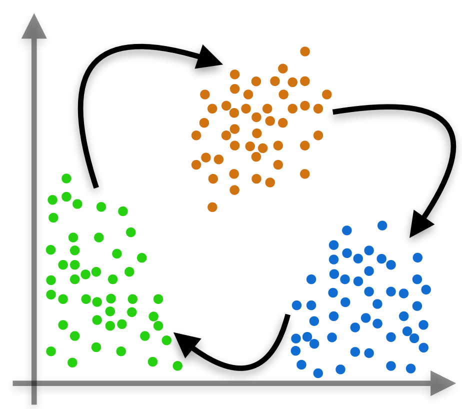
$\mathcal{A}(\mathcal{P}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}} \left[e^{-\mathrm{i}\boldsymbol{\omega}_j^T\boldsymbol{x}}\right]_{j=1}^{m}$

Compressive clustering

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



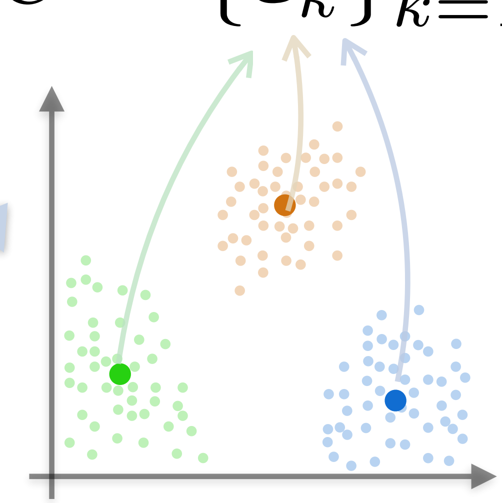
Usually (K-means,...)



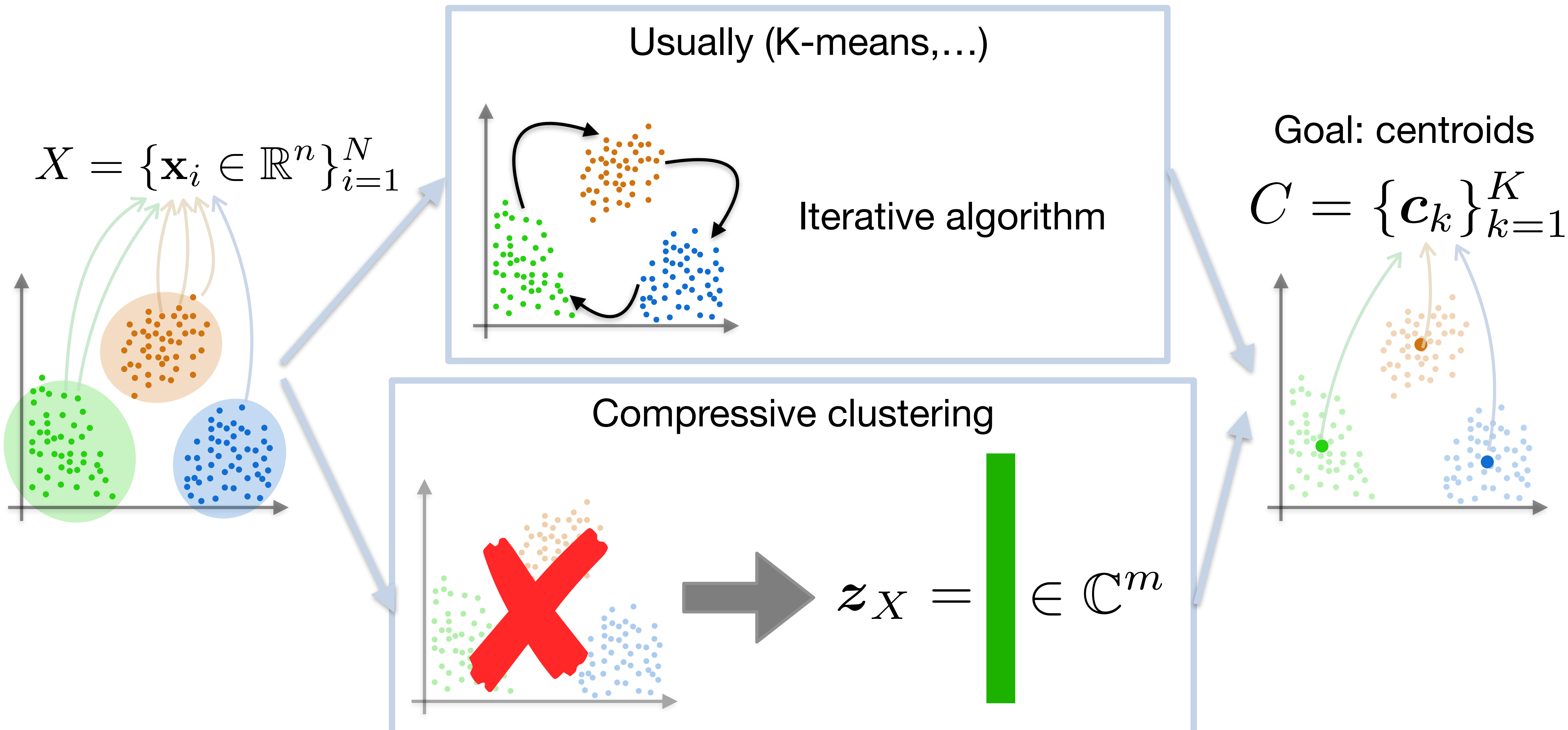
Iterative algorithm

Goal: centroids

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



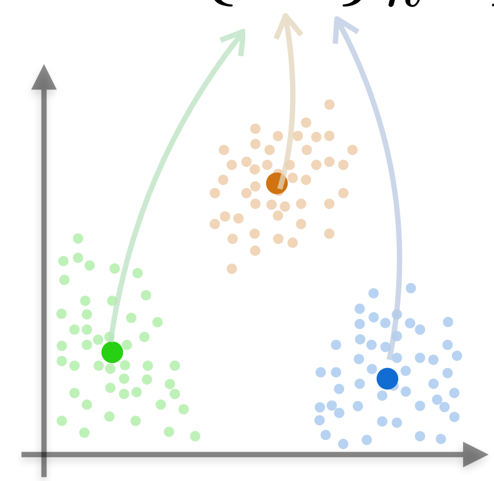
Compressive clustering



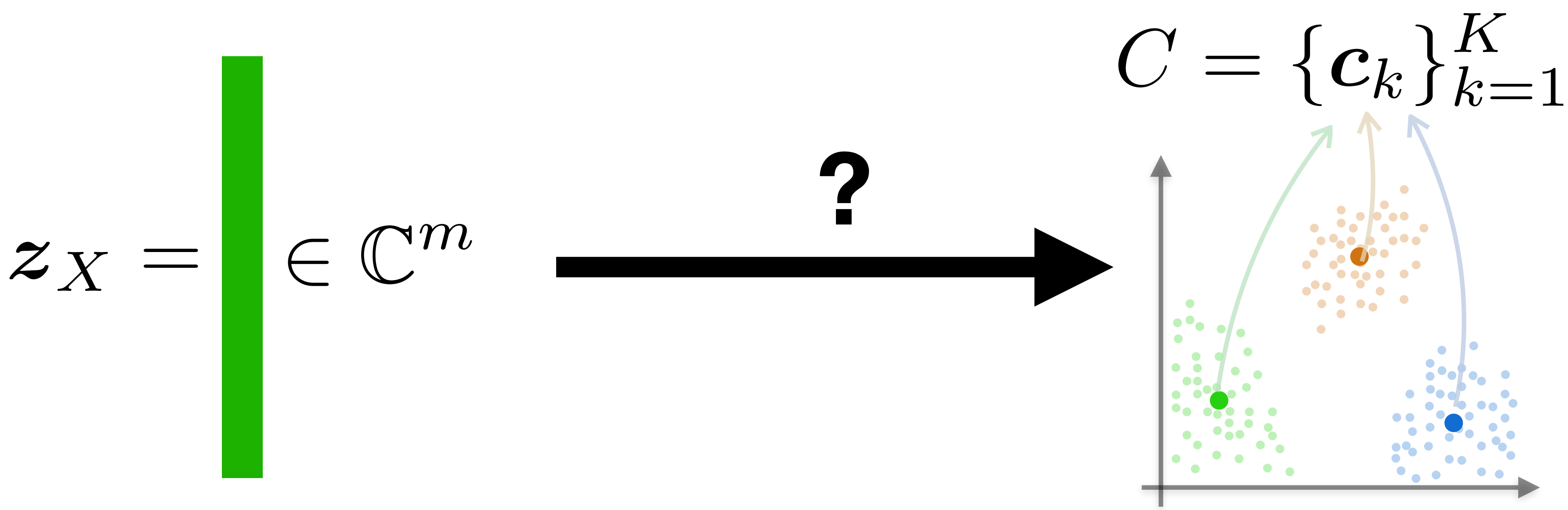
[Keriven16-CKM]

Compressive clustering

$$\boldsymbol{z}_X \in \mathbb{C}^m \xrightarrow{\;?\;} C = \{\boldsymbol{c}_k\}_{k=1}^{K}$$

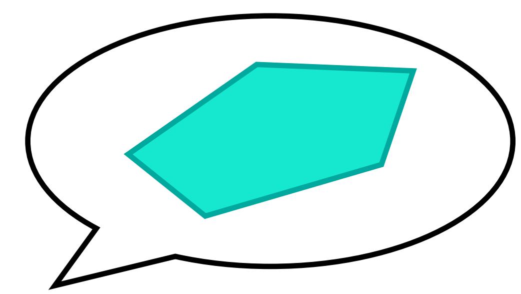


Compressive clustering



Sketch matching! (cfr. CS)

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

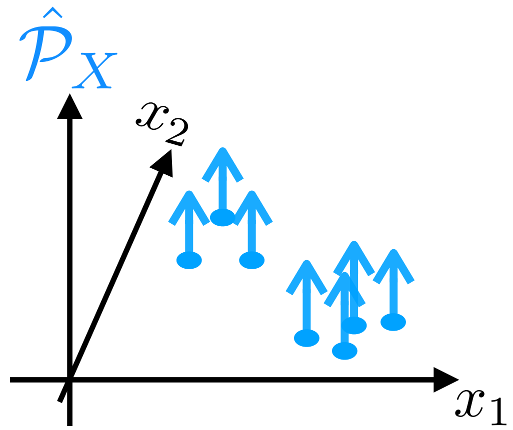


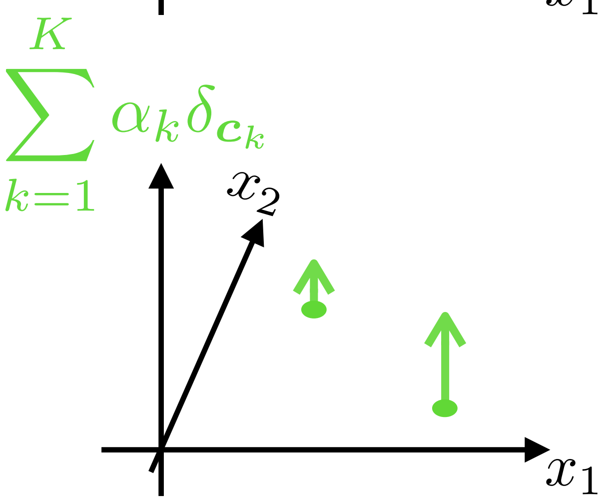
Sparse model: mixture of K diracs

Compressive clustering

Sketch matching! (cfr. CS)

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

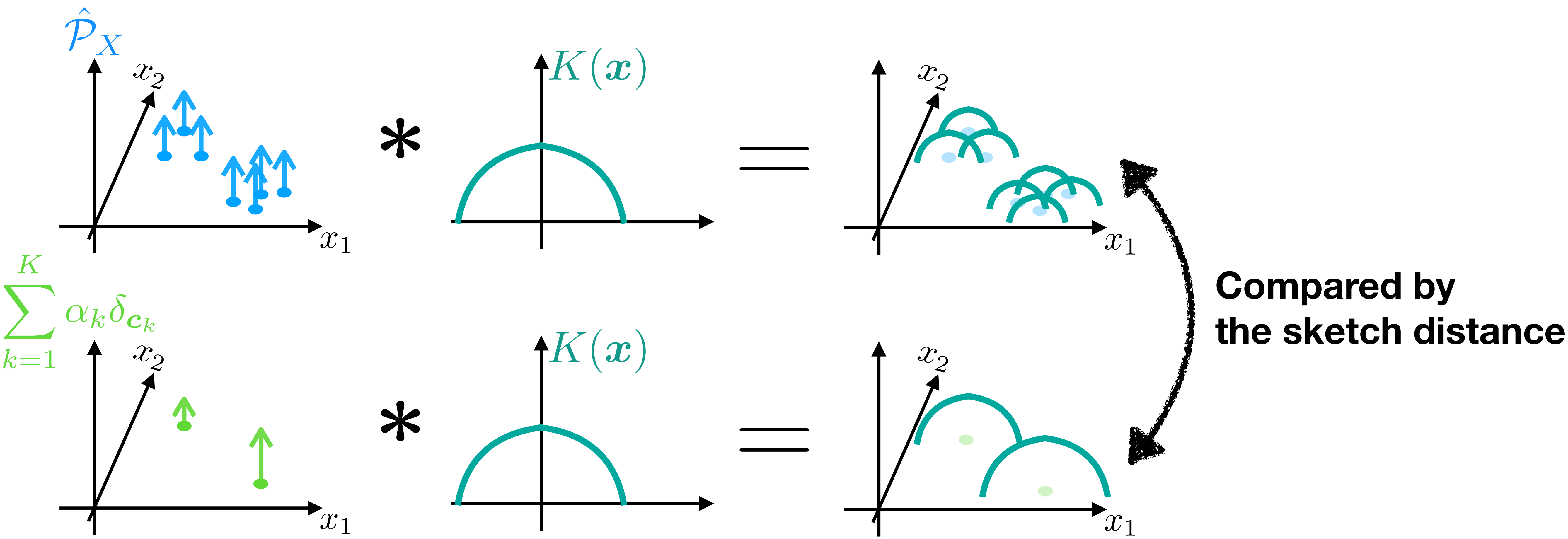




Compressive clustering

Sketch matching! (cfr. CS)

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

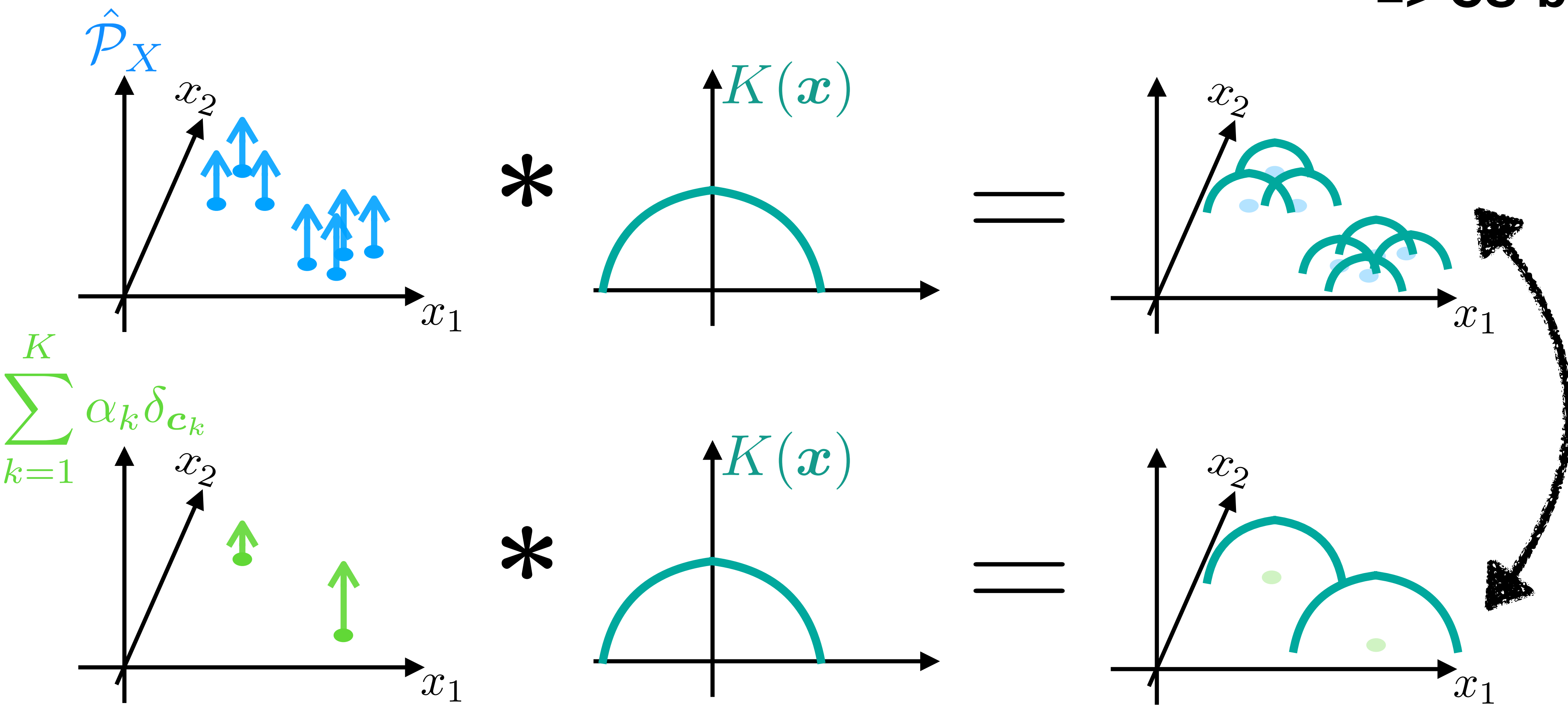


Compressive clustering

Sketch matching! (cfr. CS)

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

Highly non-convex problem!
=> CS-based heuristics



Compared by the sketch distance

The power of the sketch

Number of "measurements" m needed?

~ information rate

$$m = \mathcal{O}(nK)$$

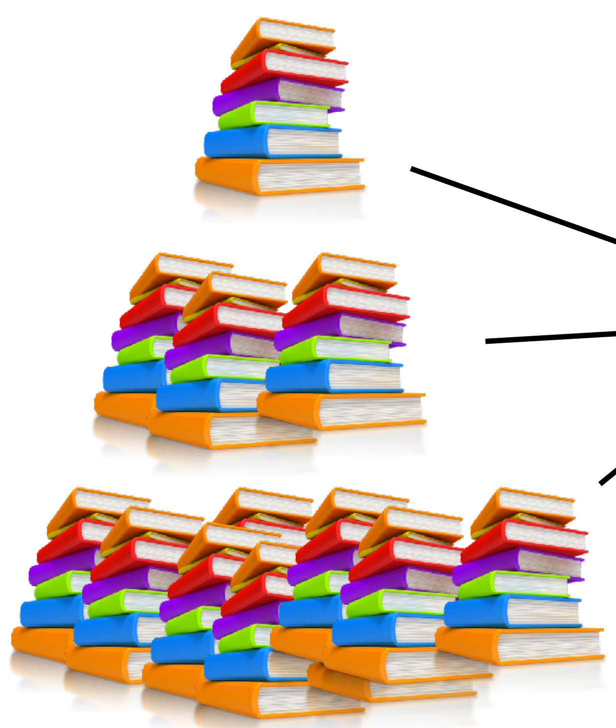
The power of the sketch

Number of "measurements" m needed?

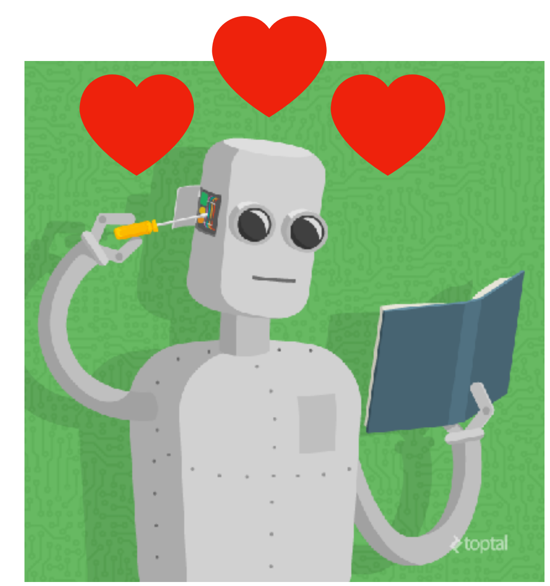
~ information rate

$$m = \mathcal{O}(nK)$$

No dependence on N!



Same learning time!
More data = better estimation of pdf



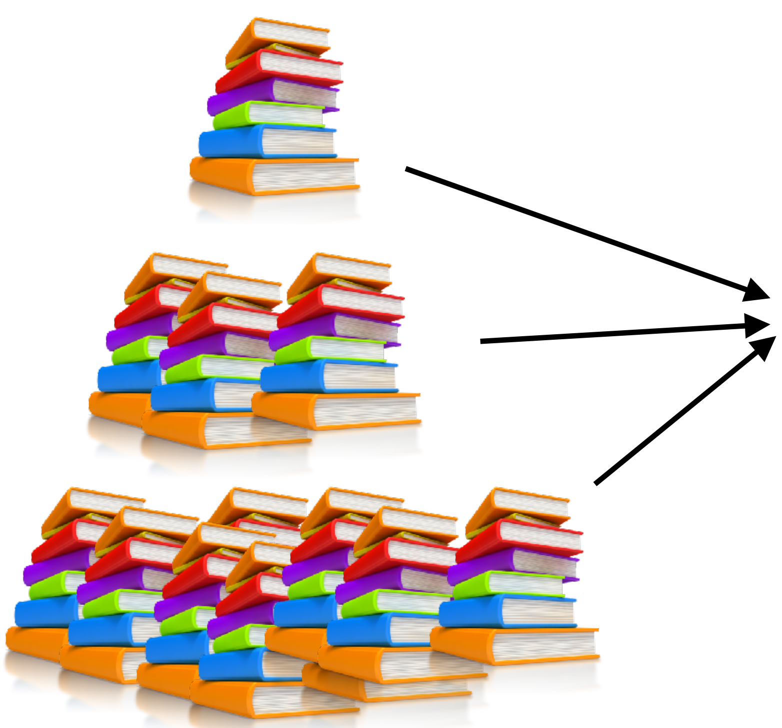
The power of the sketch

Number of "measurements" m needed?

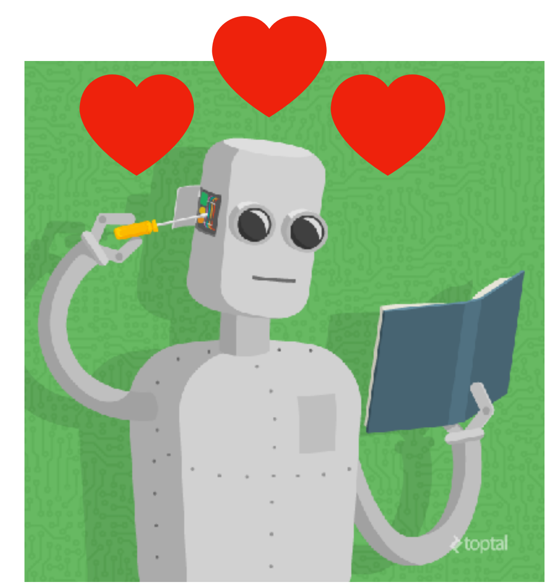
~ information rate

$$m = \mathcal{O}(nK)$$

No dependence on N!

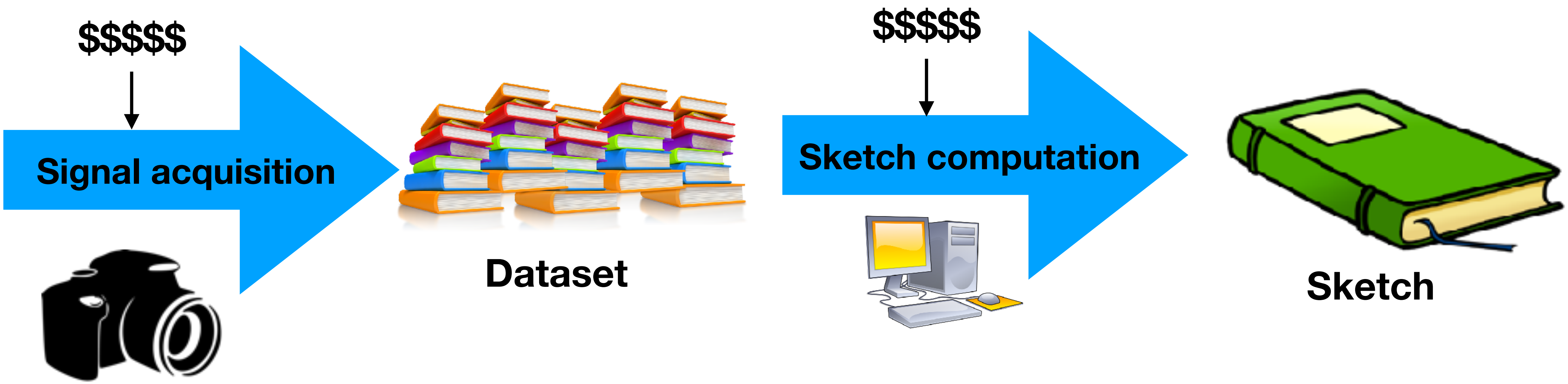


Same learning time!
More data = better estimation of pdf

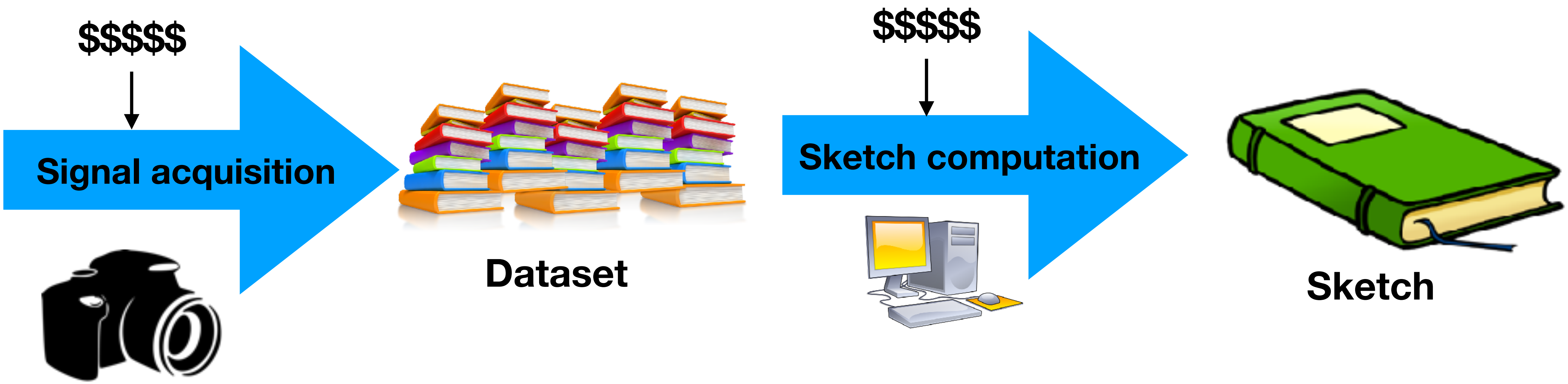


+ easy update/parallel computing of z_X

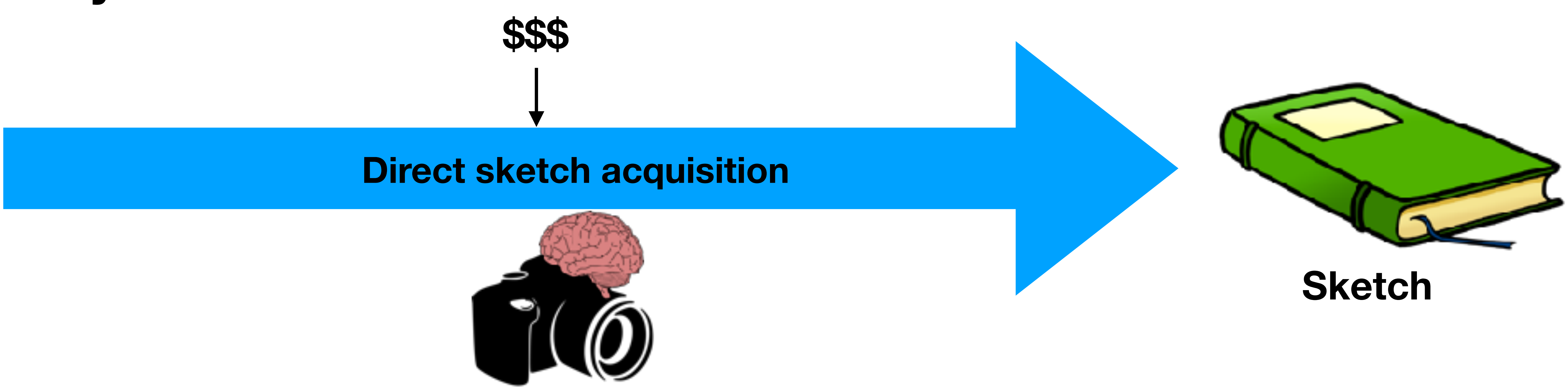
BUT...



BUT…



Why not do:

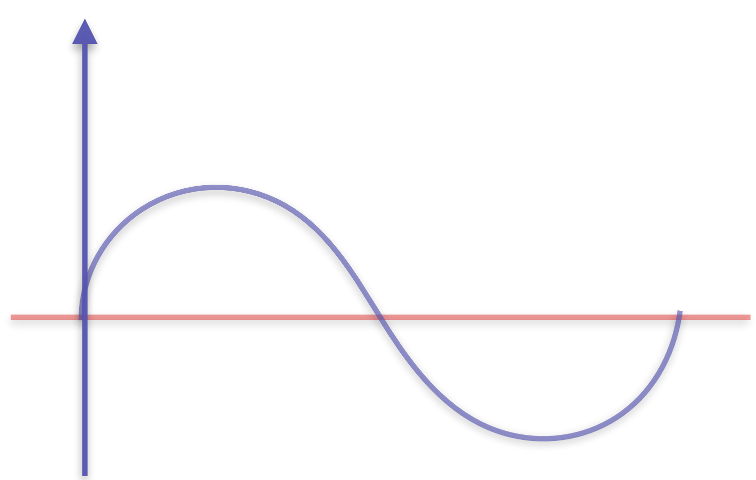


Quantized sketch (my work)

= ???

$$\boldsymbol{z}_X = \left[\frac{1}{N} \sum_{\boldsymbol{x}_i \in X} e^{-\mathrm{i}\boldsymbol{\omega}_j^T \boldsymbol{x}_i} \right]_{j=1}^{m}$$

No easy hardware implementation



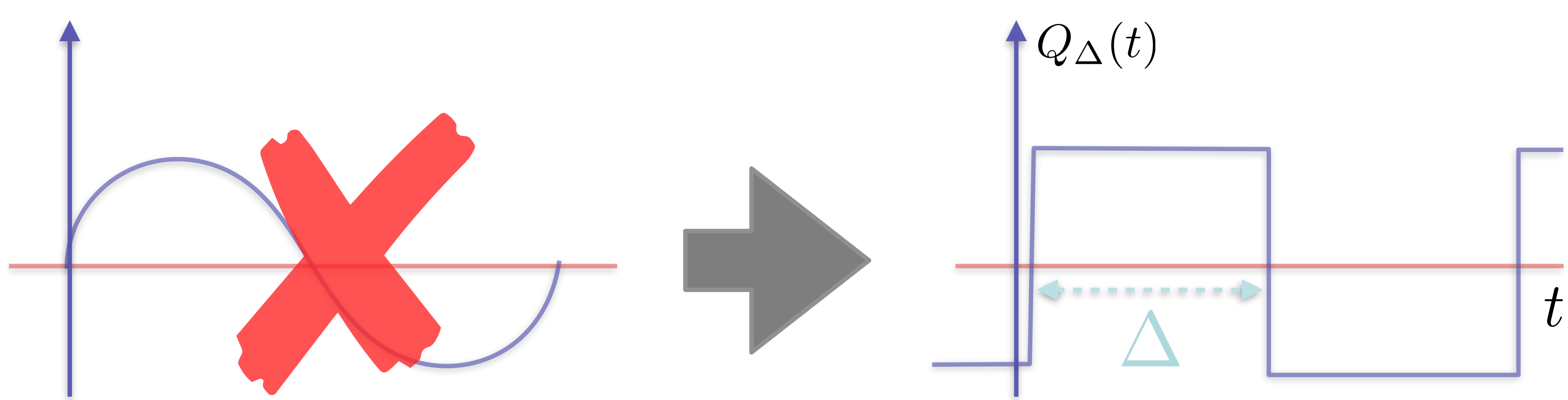
Quantized sketch (my work)

= ???

Dithering (detail)

$$z_{Q,X} = \left[\frac{1}{N} \sum_{\boldsymbol{x}_i \in X} Q_\Delta(\boldsymbol{\omega}_j^T \boldsymbol{x}_i + \xi_j) \right]_{j=1}^{m}$$

LSB of quantizer => hardware friendly



New sketch operator!

Validated on clustering
$m = O(nK)$ **increase but ok!**

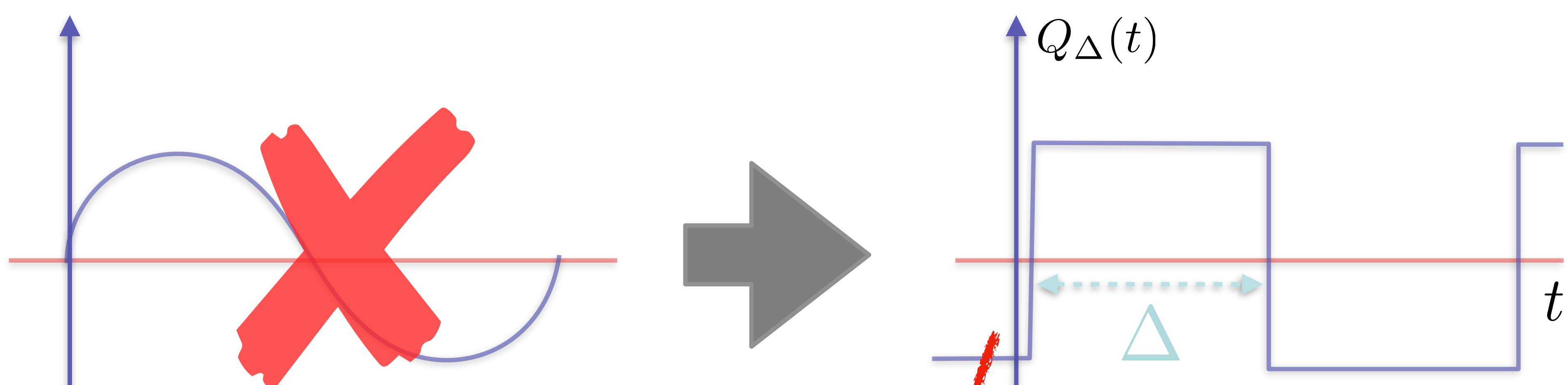
Quantized sketch (my work)

= ???

Dithering (detail)

$$\boldsymbol{z}_{Q,X} = \left[\frac{1}{N}\sum_{\boldsymbol{x}_i\in X} Q_\Delta(\boldsymbol{\omega}_j^T \boldsymbol{x}_i + \xi_j)\right]_{j=1}^{m}$$

LSB of quantizer => hardware friendly ✓



New sketch operator!

Validated on clustering

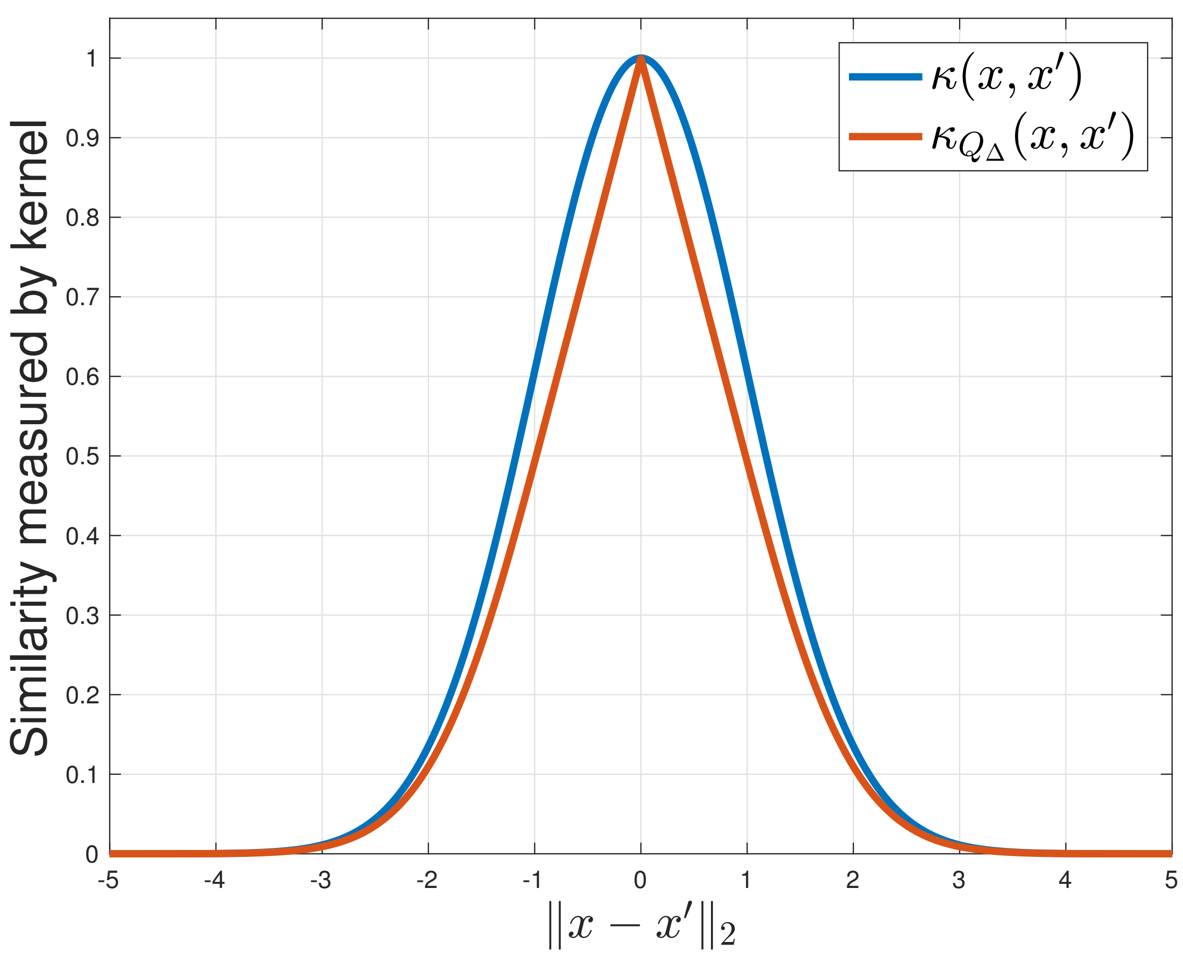
$m = O(nK)$ **increase but ok!**

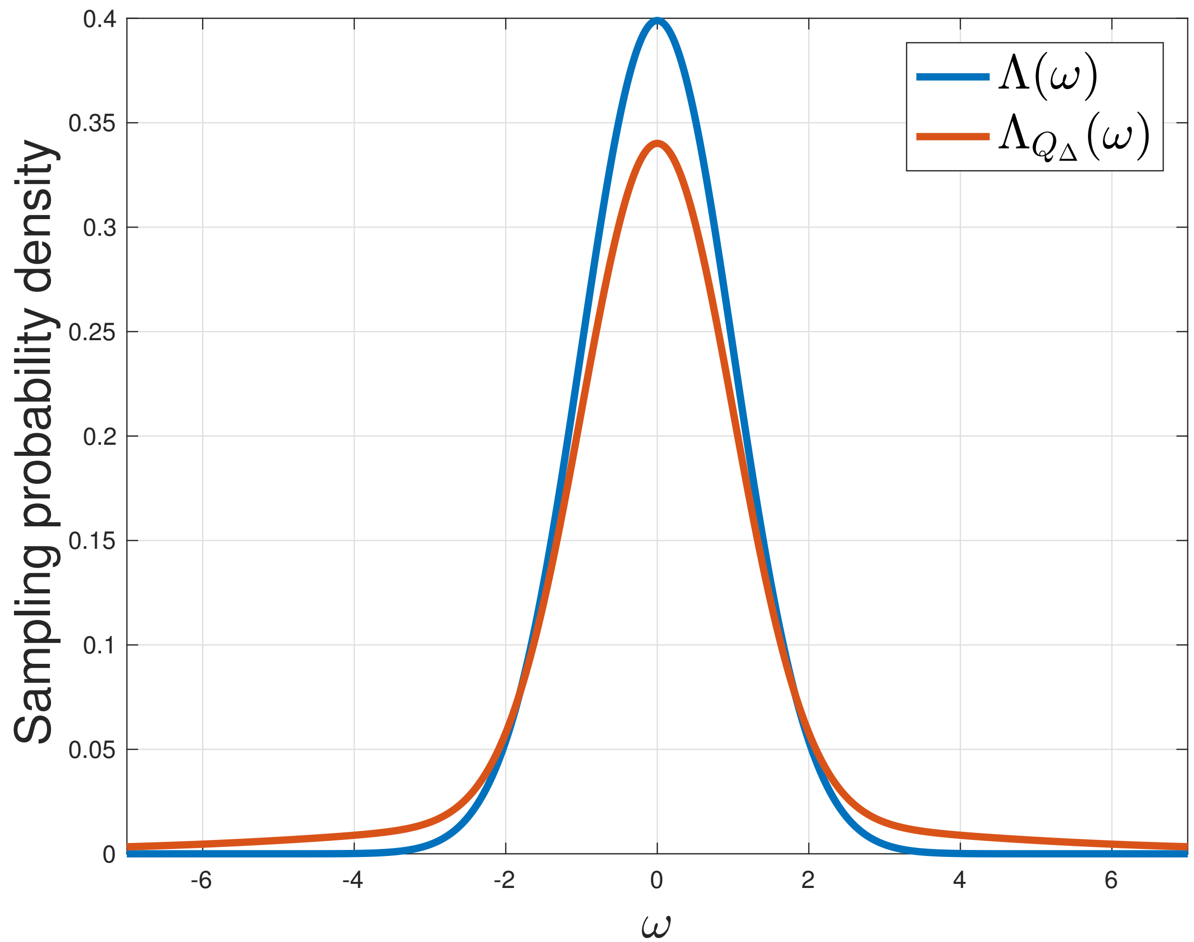
Discontinuous objective

=> gradient KO?

What does it mean?

Sketch interpretation is (only a little bit) modified





See also: 1-bit universal embeddings

What will I do next?

Some things I look forward to do:

- **Other tasks than clustering**
- **Other sketch functions**
- **Theoretical guarantees**
- **Algorithmic guarantees (local convexity?)**
- **New applications (e.g. in HS imaging?)**
- **…**

Thank you for your attention!
Questions?

References

- [Gribonval17] R. Gribonval, G. Blanchard, N. Keriven, and Y. Traonmilin, "*Compressive Statistical Learning with Random Feature Moments*," ArXiv e-prints, Jun. 2017.
- [Rahimi08] A. Rahimi and B. Recht, "*Random Features for Large-Scale Kernel Machines*," in Advances in Neural Information Processing Systems 20, 2008, pp. 1177–1184.
- [Sriperumbudur11] B. K. Sriperumbudur, "*Mixture density estimation via Hilbert space embedding of measures*," in 2011 IEEE International Symposium on Information Theory Proceedings, July 2011, pp. 1027–1030.
- [Keriven16-GMM] N. Keriven, A. Bourrier, R. Gribonval, and P. Pérez, "*Sketching for Large-Scale Learning of Mixture Models*," ArXiv e-prints, Jun. 2016.
- [Keriven16-CKM] N. Keriven, N. Tremblay, Y. Traonmilin, and R. Gribonval, "*Compressive K-means*," arXiv preprint arXiv:1610.08738, 2016.