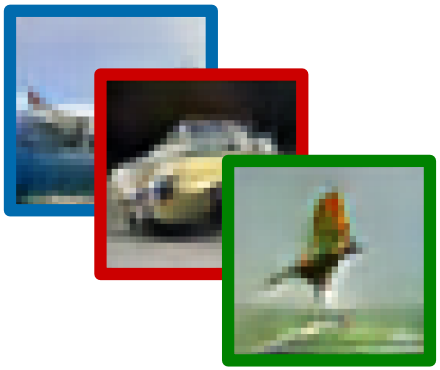
Can Random Matrices Change the Future of Machine Learning?
Digicosme "Junior Conference on Wireless and Optical Communications 2020"

Romain COUILLET

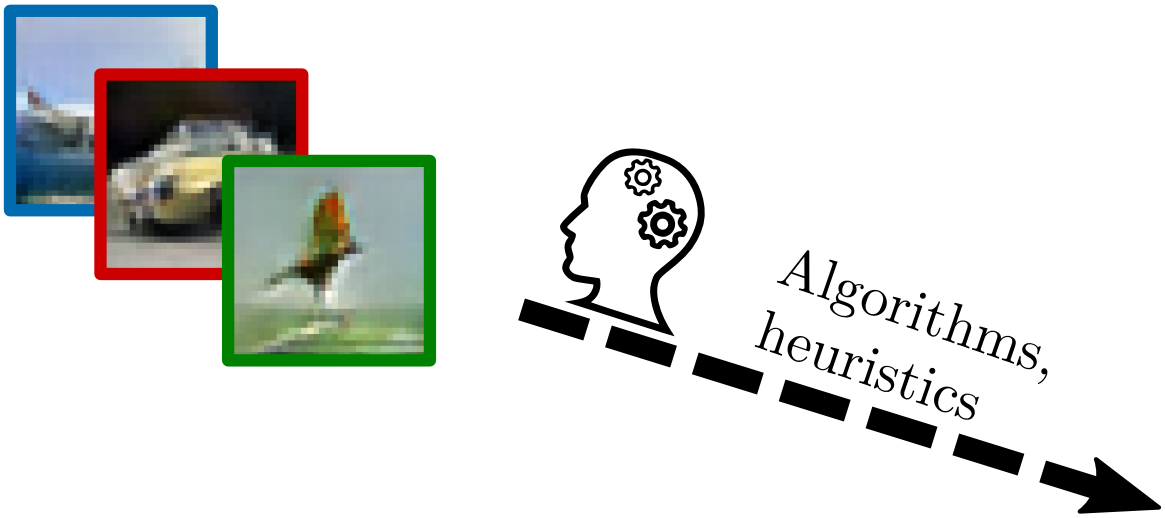
CentraleSupélec, L2S, University of ParisSaclay, France
GSTATS IDEX DataScience Chair, GIPSA-lab, University Grenoble–Alpes, France.

December 3, 2020

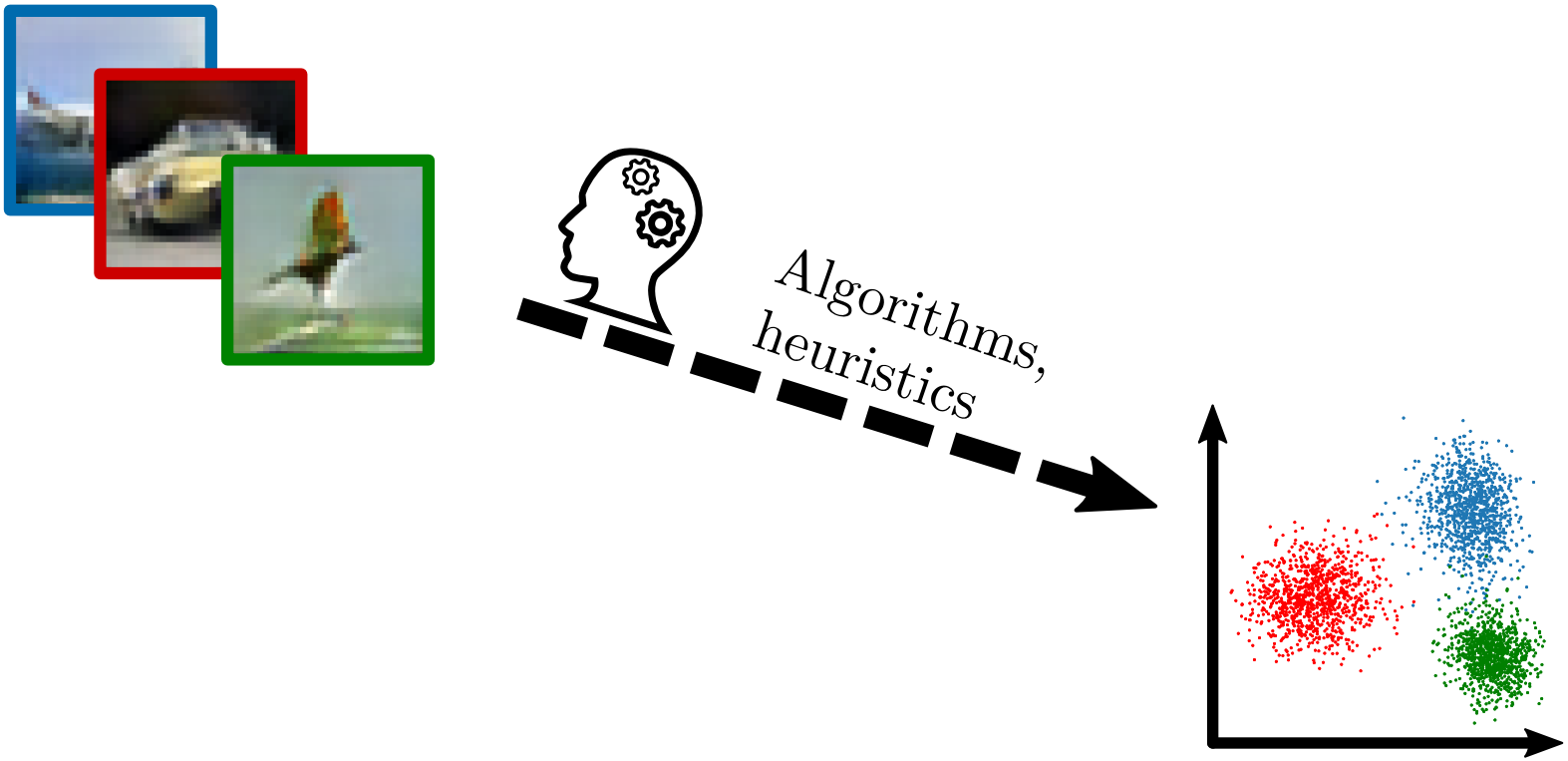
A long story short...



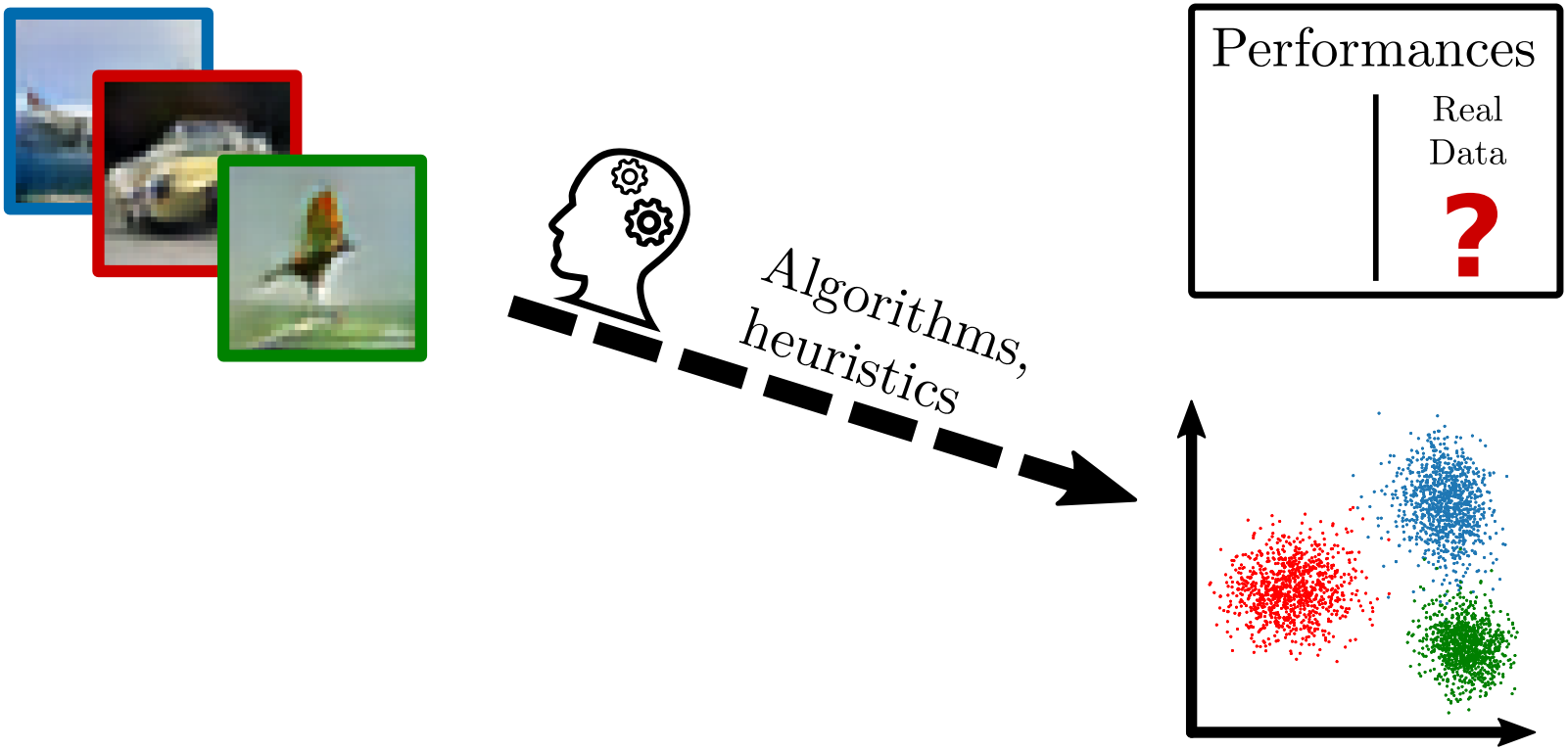
A long story short...



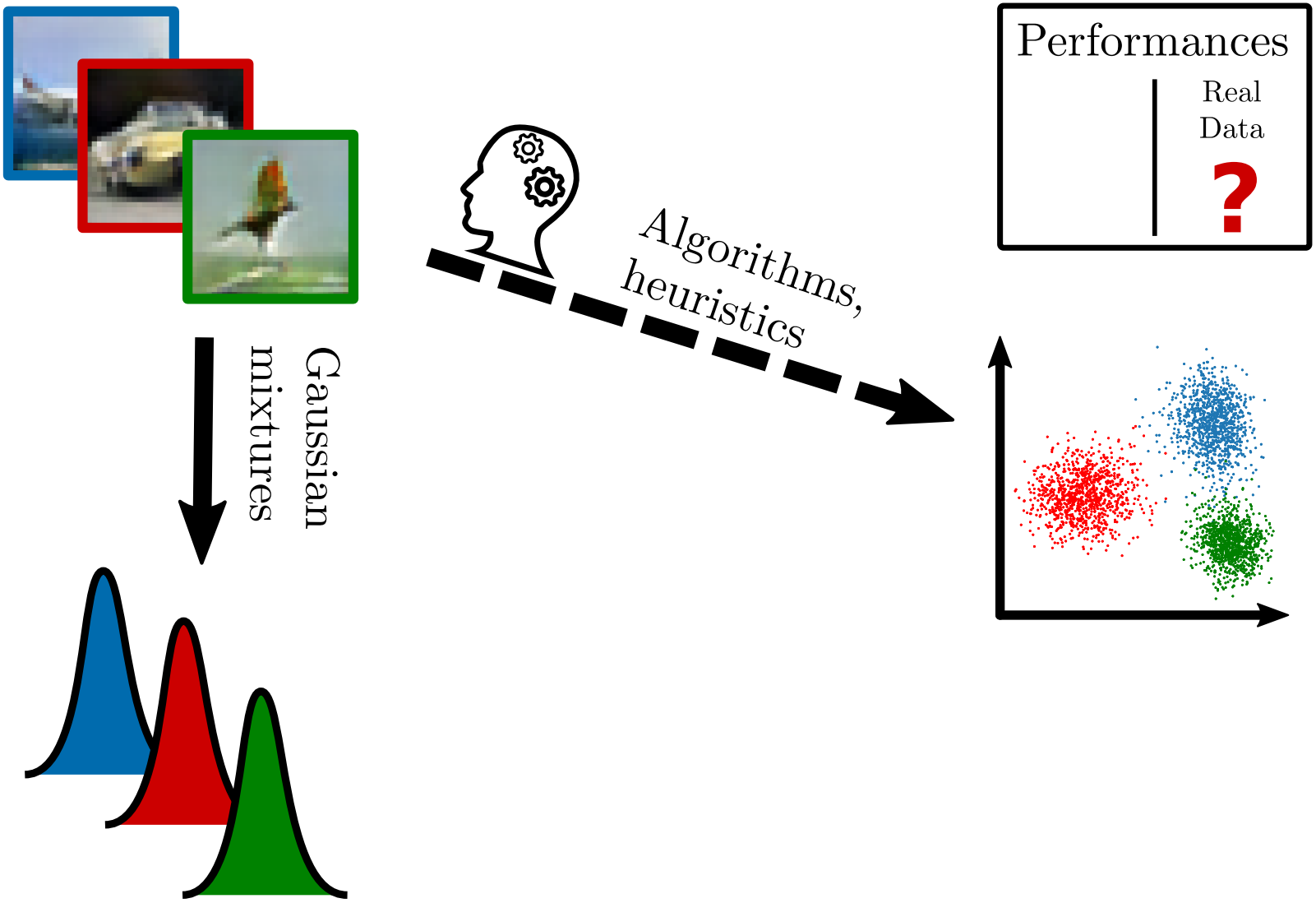
A long story short...



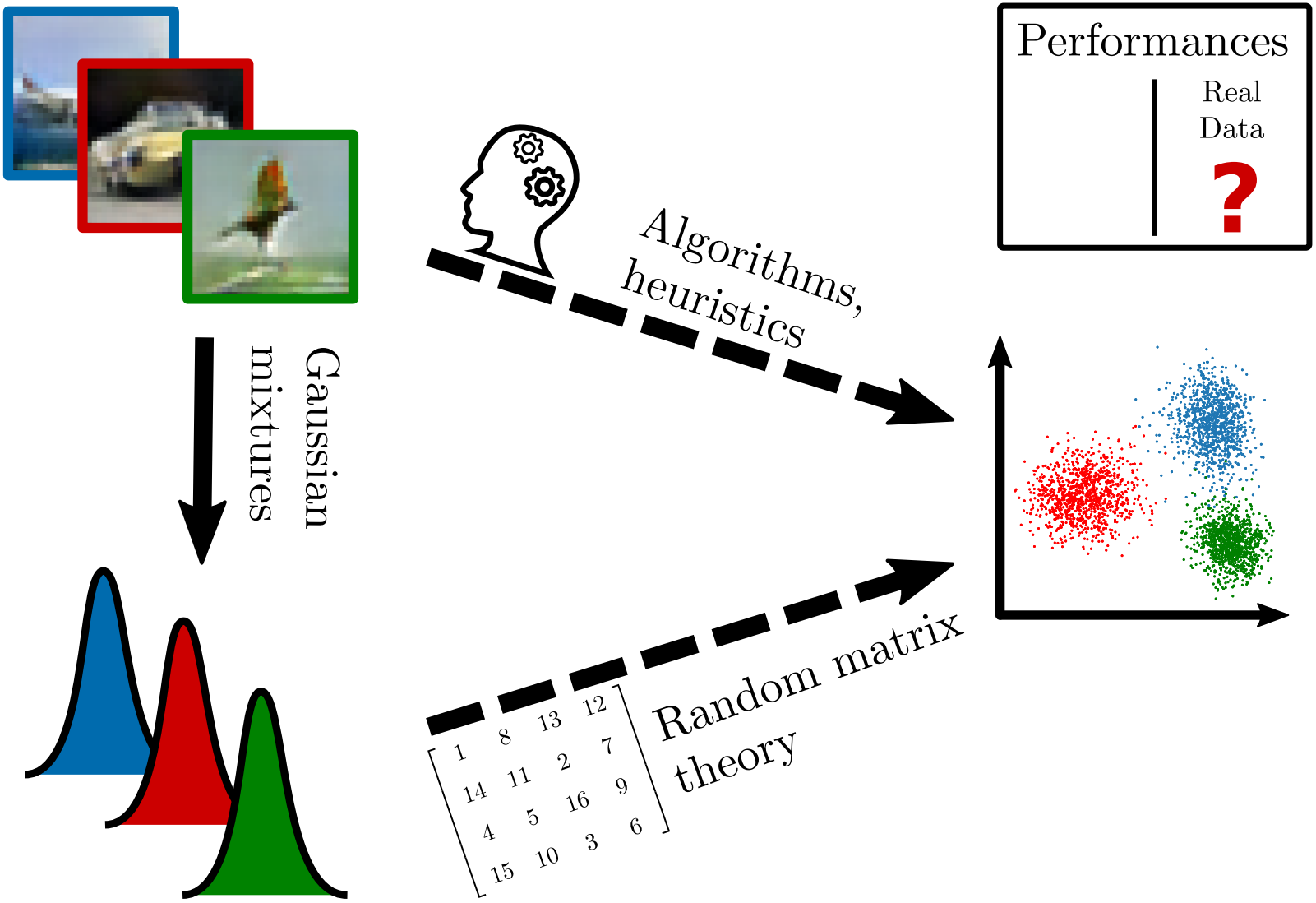
A long story short...



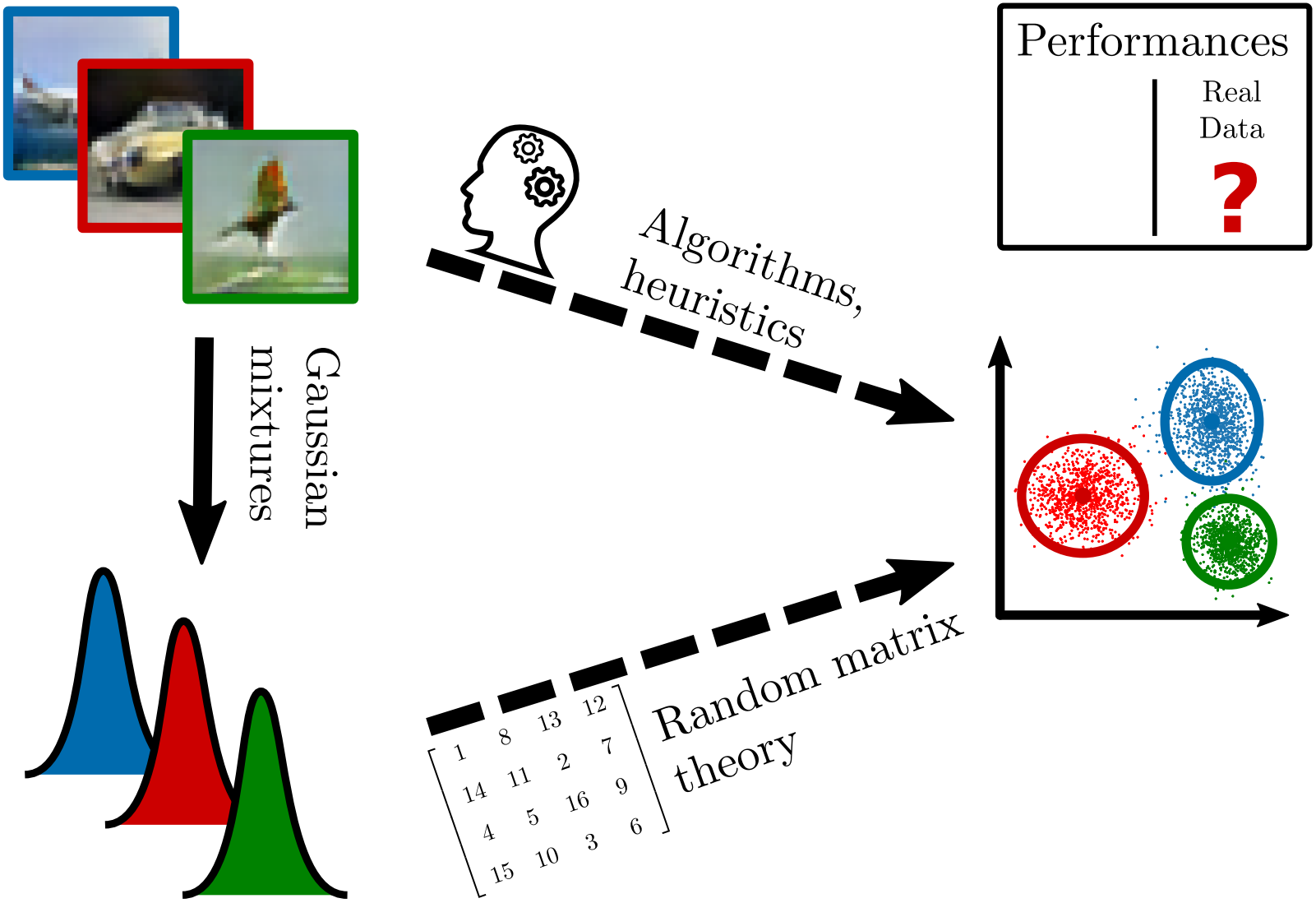
A long story short...



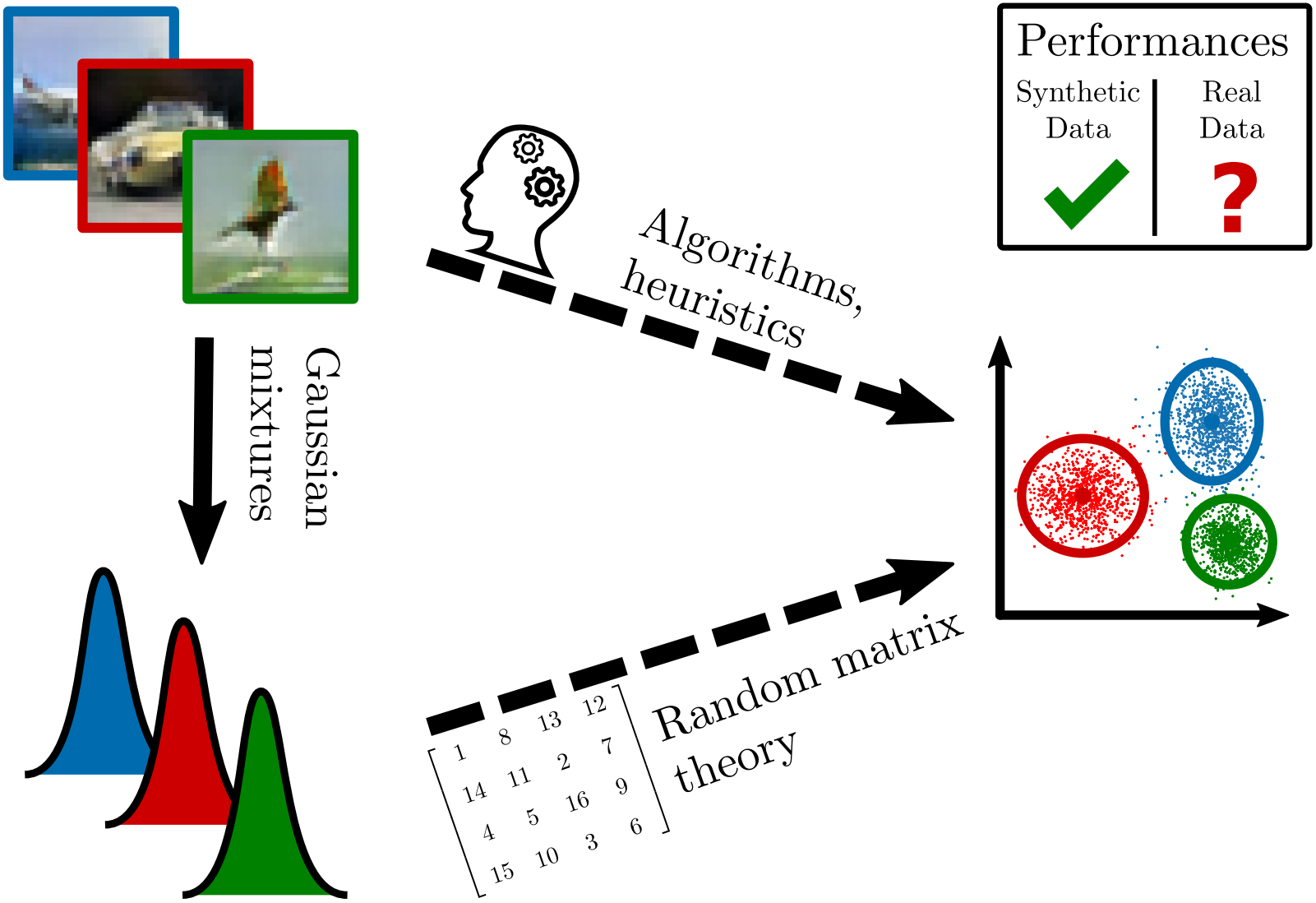
A long story short...



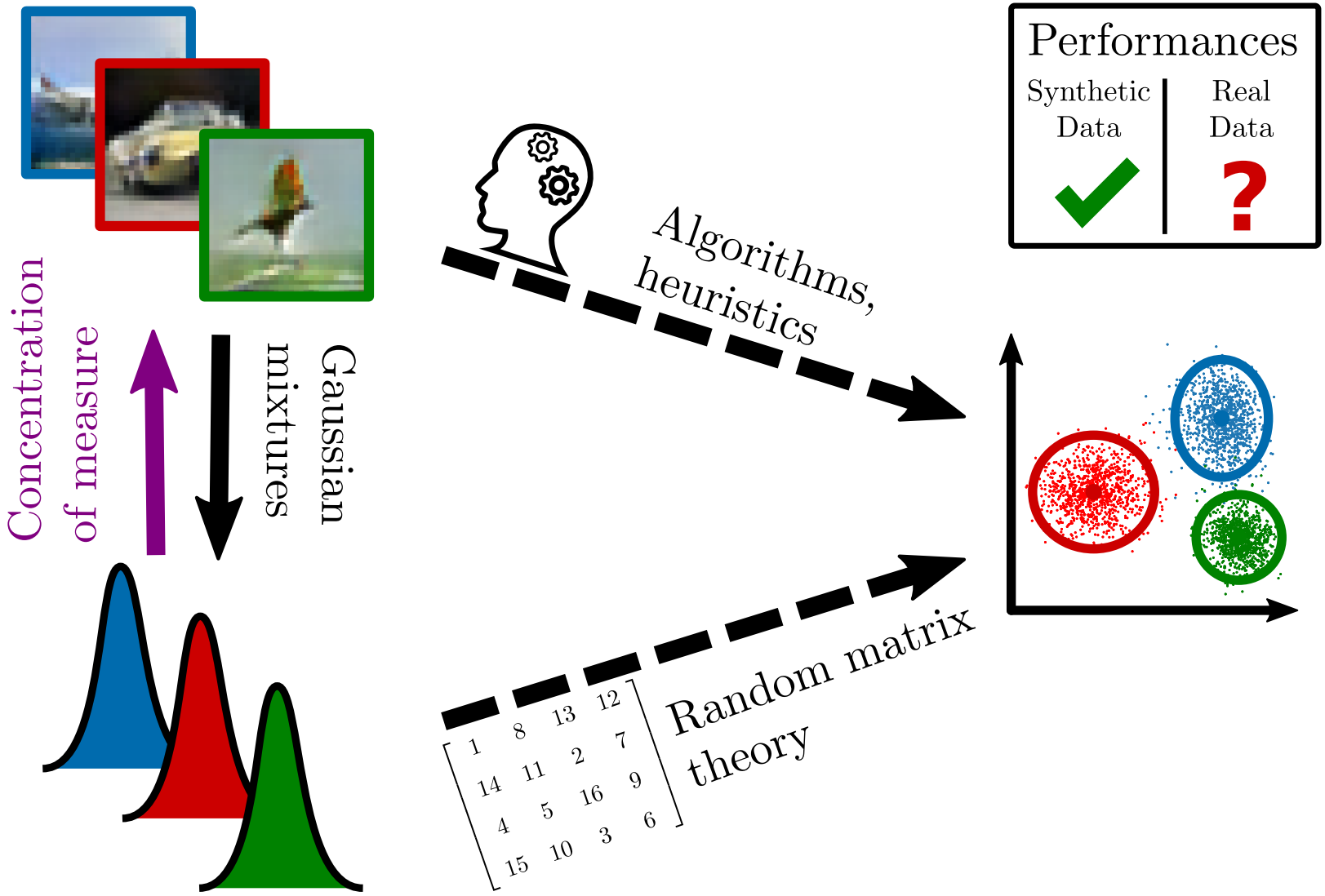
A long story short...



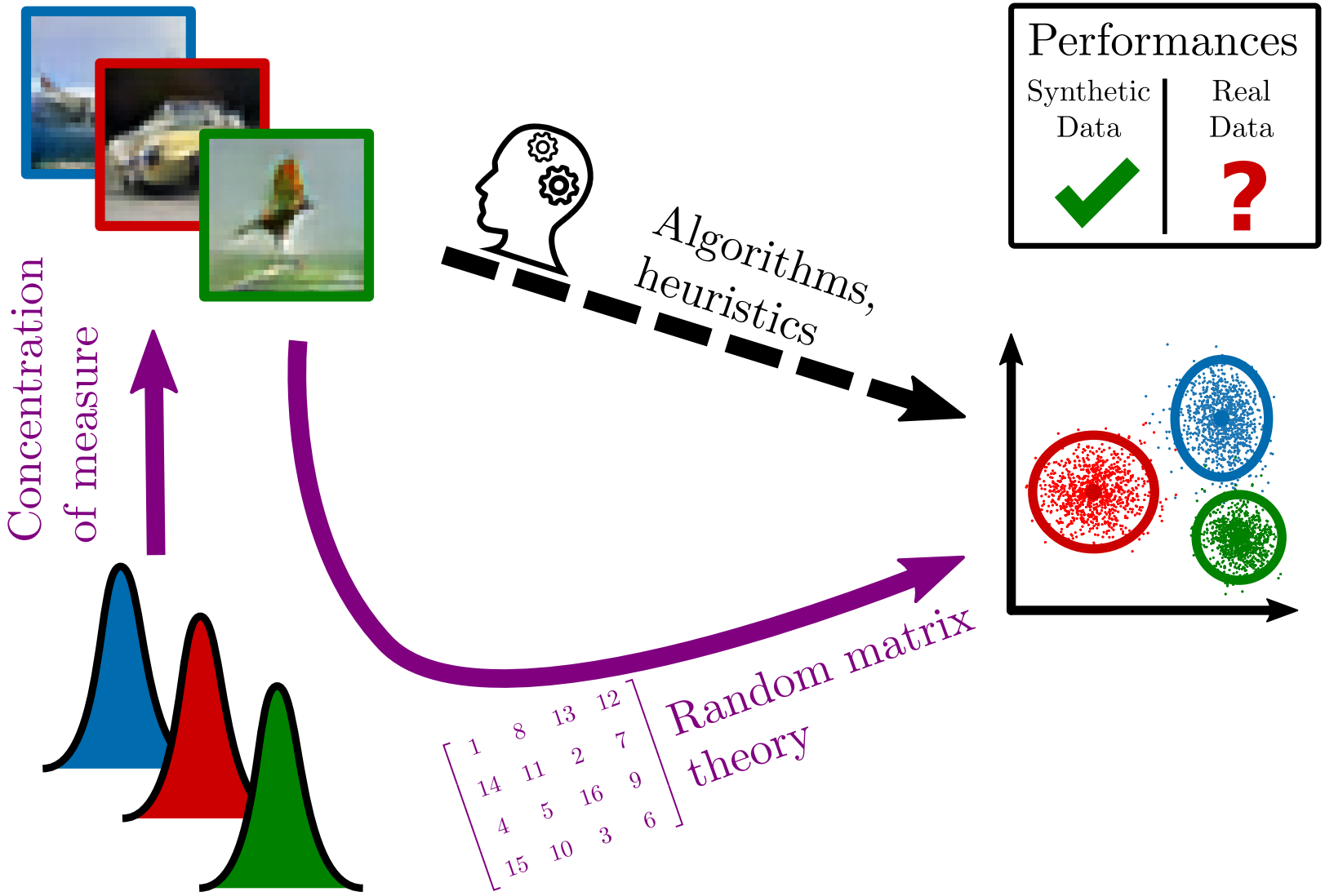
A long story short...



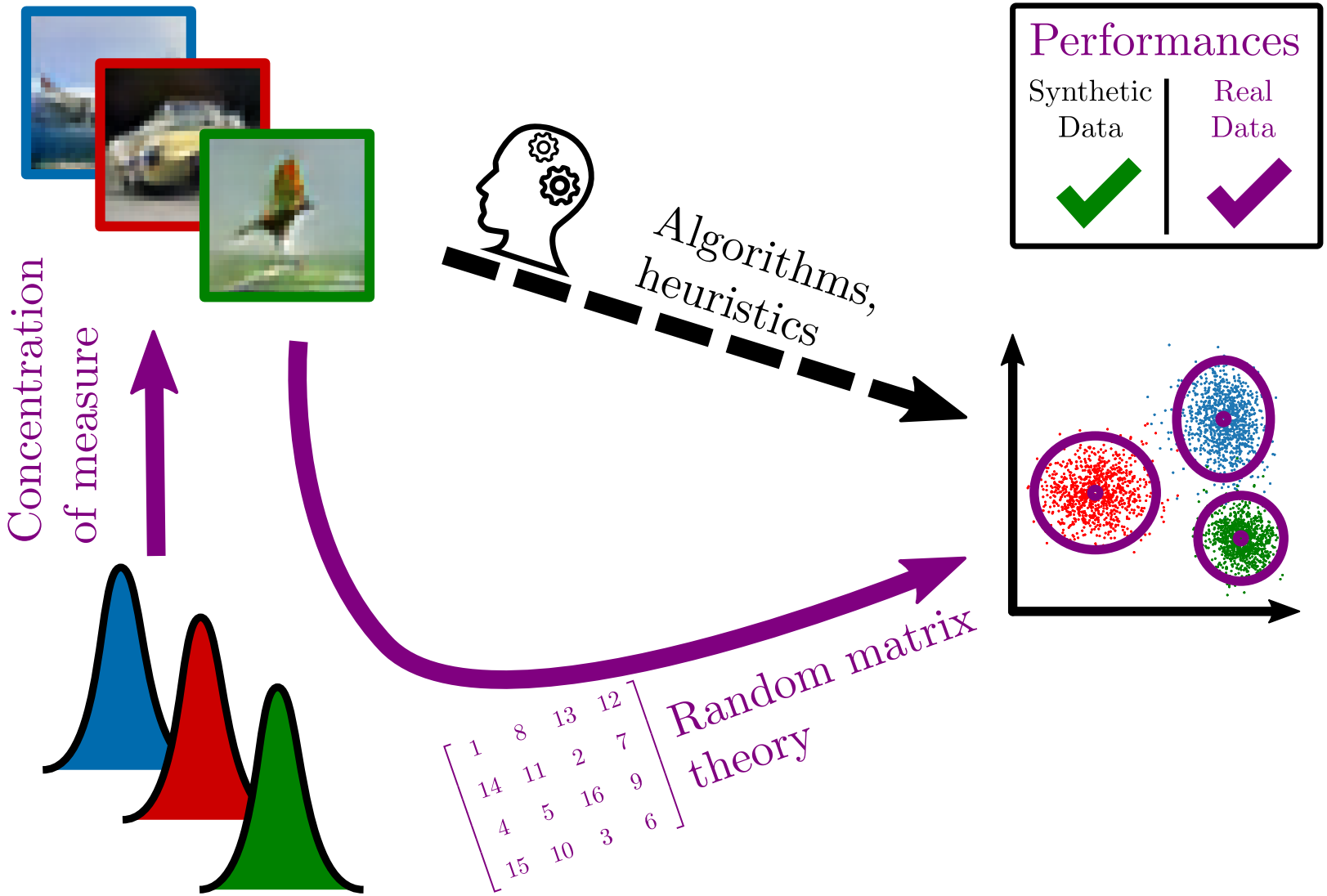
A long story short...



A long story short...



A long story short...



Outline

Outline

Outline

Context

Baseline scenario: $y_1, \ldots, y_n \in \mathbb{C}^p$ (or $\mathbb{R}^p$) i.i.d. with $E[y_1] = 0$, $E[y_1 y_1^*] = C_p$:

Context

Baseline scenario: $y_1, \ldots, y_n \in \mathbb{C}^p$ (or $\mathbb{R}^p$) i.i.d. with $E[y_1] = 0$, $E[y_1 y_1^*] = C_p$:

- If $y_1 \sim \mathcal{N}(0, C_p)$, ML estimator for C_p is the sample covariance matrix (SCM)

$$\hat{C}_p = \frac{1}{n} Y_p Y_p^* = \frac{1}{n} \sum_{i=1}^{n} y_i y_i^*$$

$(Y_p = [y_1, \ldots, y_n] \in \mathbb{C}^{p \times n})$.

Context

Baseline scenario: $y_1, \ldots, y_n \in \mathbb{C}^p$ (or $\mathbb{R}^p$) i.i.d. with $E[y_1] = 0$, $E[y_1 y_1^*] = C_p$:

- If $y_1 \sim \mathcal{N}(0, C_p)$, ML estimator for C_p is the sample covariance matrix (SCM)

$$\hat{C}_p = \frac{1}{n} Y_p Y_p^* = \frac{1}{n} \sum_{i=1}^{n} y_i y_i^*$$

  ($Y_p = [y_1, \ldots, y_n] \in \mathbb{C}^{p \times n}$).
- If $n \to \infty$, then, strong law of large numbers

$$\hat{C}_p \xrightarrow{\text{a.s.}} C_p.$$

  or equivalently, in spectral norm

$$\left\| \hat{C}_p - C_p \right\| \xrightarrow{\text{a.s.}} 0.$$

Context

Baseline scenario: $y_1, \ldots, y_n \in \mathbb{C}^p$ (or $\mathbb{R}^p$) i.i.d. with $E[y_1] = 0$, $E[y_1 y_1^*] = C_p$:

- If $y_1 \sim \mathcal{N}(0, C_p)$, ML estimator for C_p is the sample covariance matrix (SCM)

$$\hat{C}_p = \frac{1}{n} Y_p Y_p^* = \frac{1}{n} \sum_{i=1}^{n} y_i y_i^*$$

  ($Y_p = [y_1, \ldots, y_n] \in \mathbb{C}^{p \times n}$).
- If $n \to \infty$, then, strong law of large numbers

$$\hat{C}_p \xrightarrow{\text{a.s.}} C_p.$$

  or equivalently, in spectral norm

$$\left\| \hat{C}_p - C_p \right\| \xrightarrow{\text{a.s.}} 0.$$

Random Matrix Regime

- No longer valid if $p, n \to \infty$ with $p/n \to c \in (0, \infty)$,

$$\left\| \hat{C}_p - C_p \right\| \not\to 0.$$

Context

Baseline scenario: $y_1, \ldots, y_n \in \mathbb{C}^p$ (or $\mathbb{R}^p$) i.i.d. with $E[y_1] = 0$, $E[y_1 y_1^*] = C_p$:

- If $y_1 \sim \mathcal{N}(0, C_p)$, ML estimator for C_p is the sample covariance matrix (SCM)

$$\hat{C}_p = \frac{1}{n} Y_p Y_p^* = \frac{1}{n} \sum_{i=1}^{n} y_i y_i^*$$

  ($Y_p = [y_1, \ldots, y_n] \in \mathbb{C}^{p \times n}$).
- If $n \to \infty$, then, strong law of large numbers

$$\hat{C}_p \xrightarrow{\text{a.s.}} C_p.$$

  or equivalently, in spectral norm

$$\left\| \hat{C}_p - C_p \right\| \xrightarrow{\text{a.s.}} 0.$$

Random Matrix Regime

- No longer valid if $p, n \to \infty$ with $p/n \to c \in (0, \infty)$,

$$\left\| \hat{C}_p - C_p \right\| \not\to 0.$$

- For practical p, n with $p \simeq n$, leads to dramatically wrong conclusions

Context

Baseline scenario: $y_1, \ldots, y_n \in \mathbb{C}^p$ (or $\mathbb{R}^p$) i.i.d. with $E[y_1] = 0$, $E[y_1 y_1^*] = C_p$:

- If $y_1 \sim \mathcal{N}(0, C_p)$, ML estimator for C_p is the sample covariance matrix (SCM)

$$\hat{C}_p = \frac{1}{n} Y_p Y_p^* = \frac{1}{n} \sum_{i=1}^{n} y_i y_i^*$$

  ($Y_p = [y_1, \ldots, y_n] \in \mathbb{C}^{p \times n}$).
- If $n \to \infty$, then, strong law of large numbers

$$\hat{C}_p \xrightarrow{\text{a.s.}} C_p.$$

  or equivalently, in spectral norm

$$\left\| \hat{C}_p - C_p \right\| \xrightarrow{\text{a.s.}} 0.$$

Random Matrix Regime

- No longer valid if $p, n \to \infty$ with $p/n \to c \in (0, \infty)$,

$$\left\| \hat{C}_p - C_p \right\| \not\to 0.$$

- For practical p, n with $p \simeq n$, leads to dramatically wrong conclusions
- **Even for** $p = n/100$.

The Marčenko–Pastur law

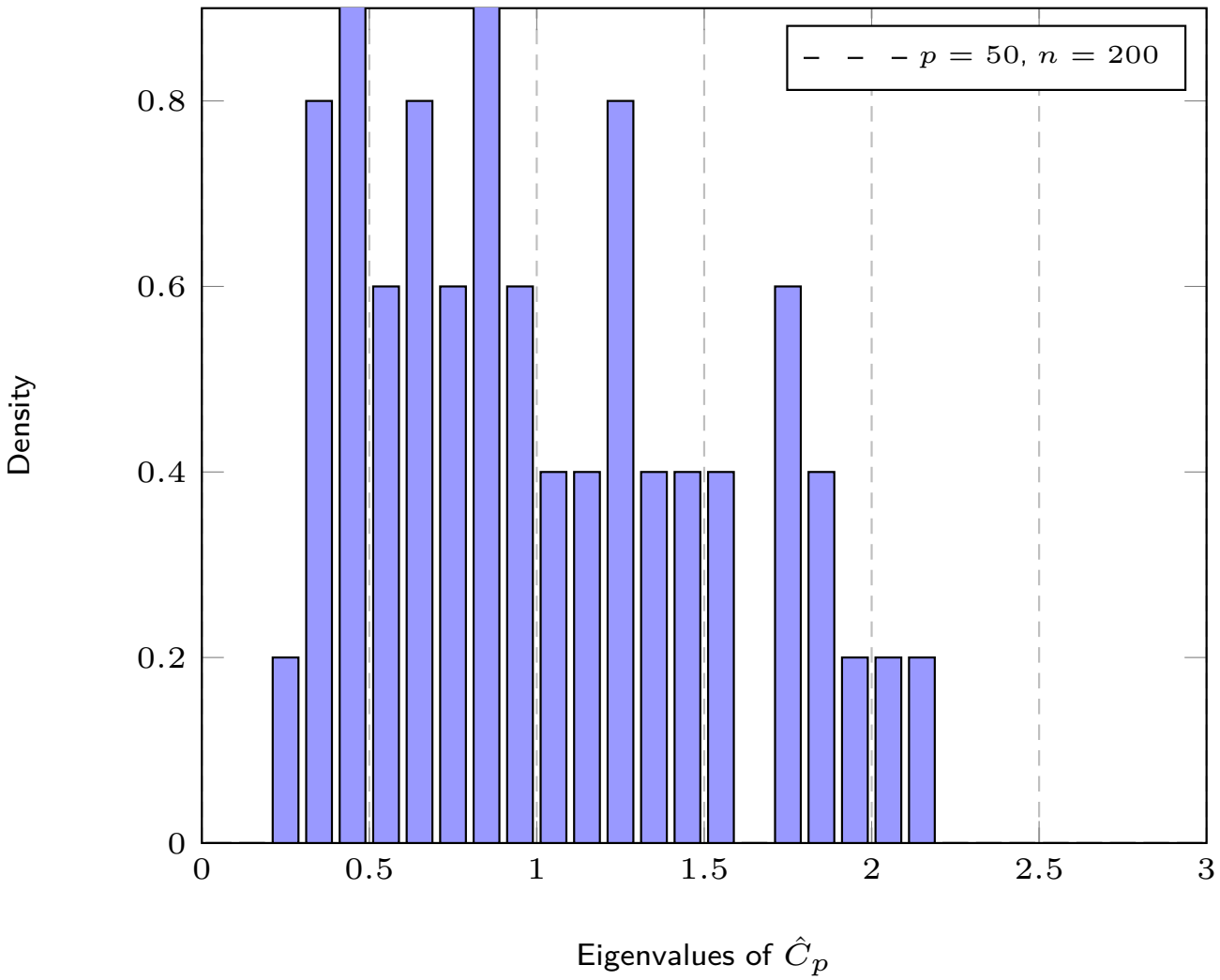


Figure: Histogram of the eigenvalues of $\hat{C}_p$ for $c = 1/4$, $C_p = I_p$.

The Marčenko–Pastur law

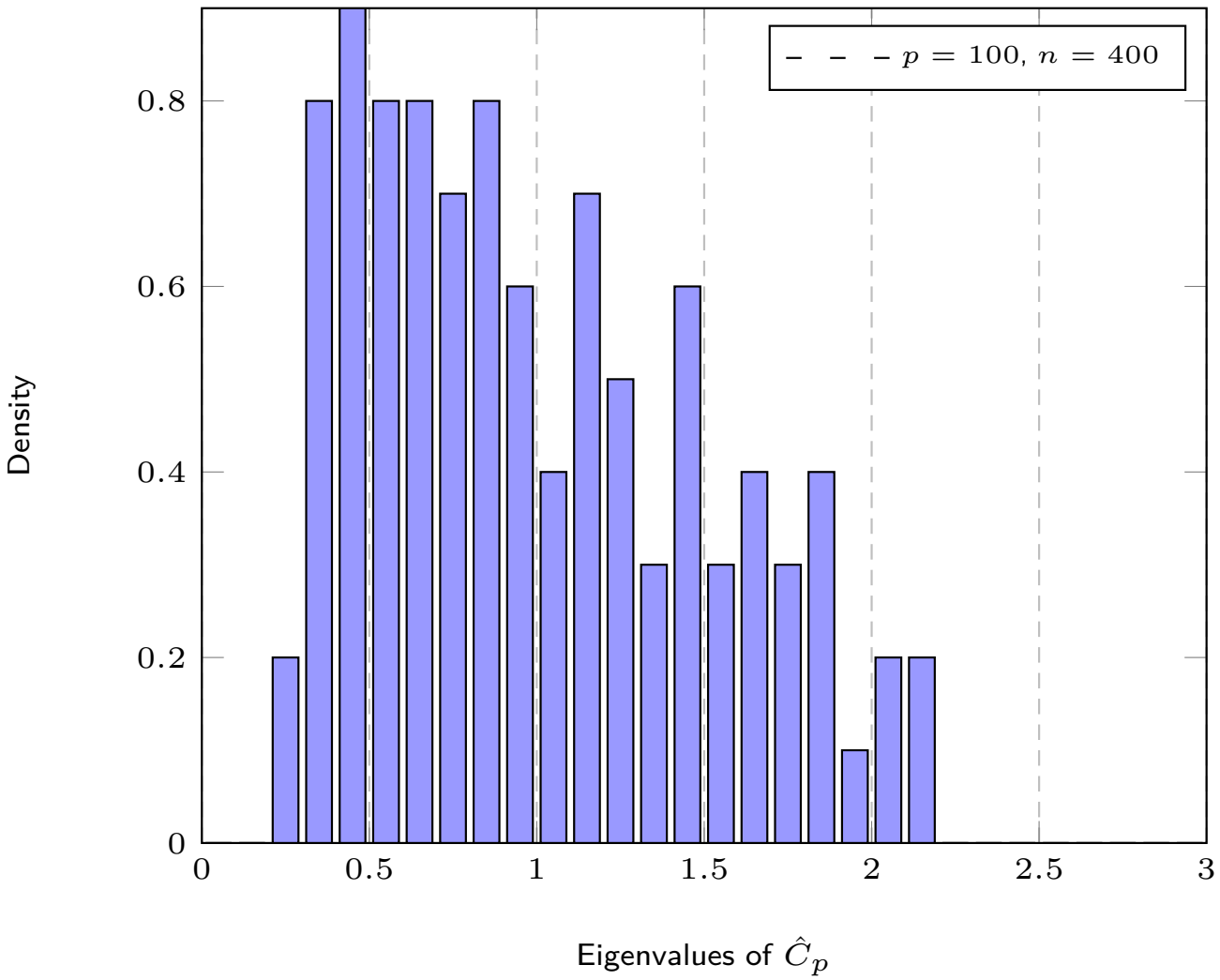


Figure: Histogram of the eigenvalues of $\hat{C}_p$ for $c = 1/4$, $C_p = I_p$.

The Marčenko–Pastur law

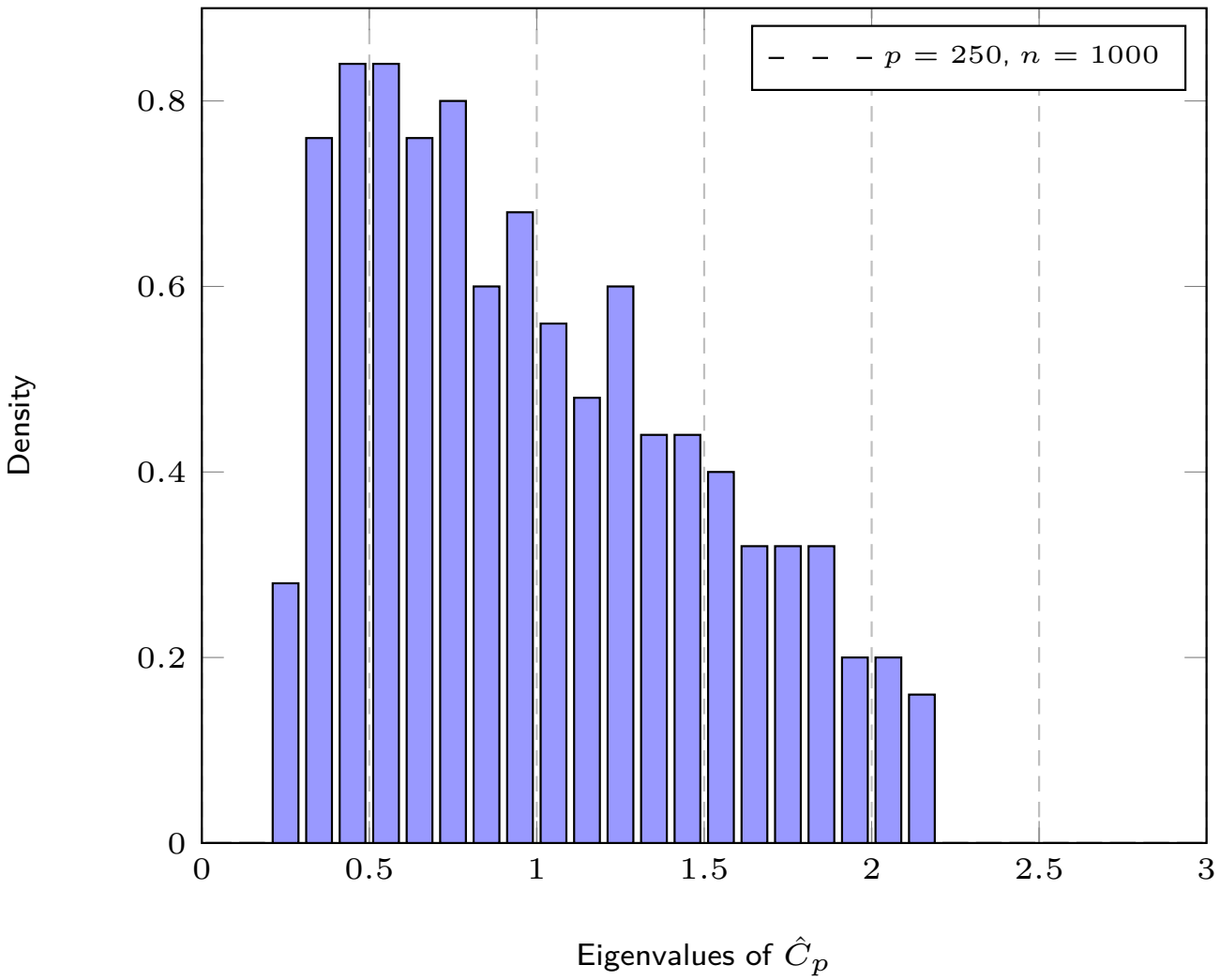


Figure: Histogram of the eigenvalues of $\hat{C}_p$ for $c = 1/4$, $C_p = I_p$.

The Marčenko–Pastur law

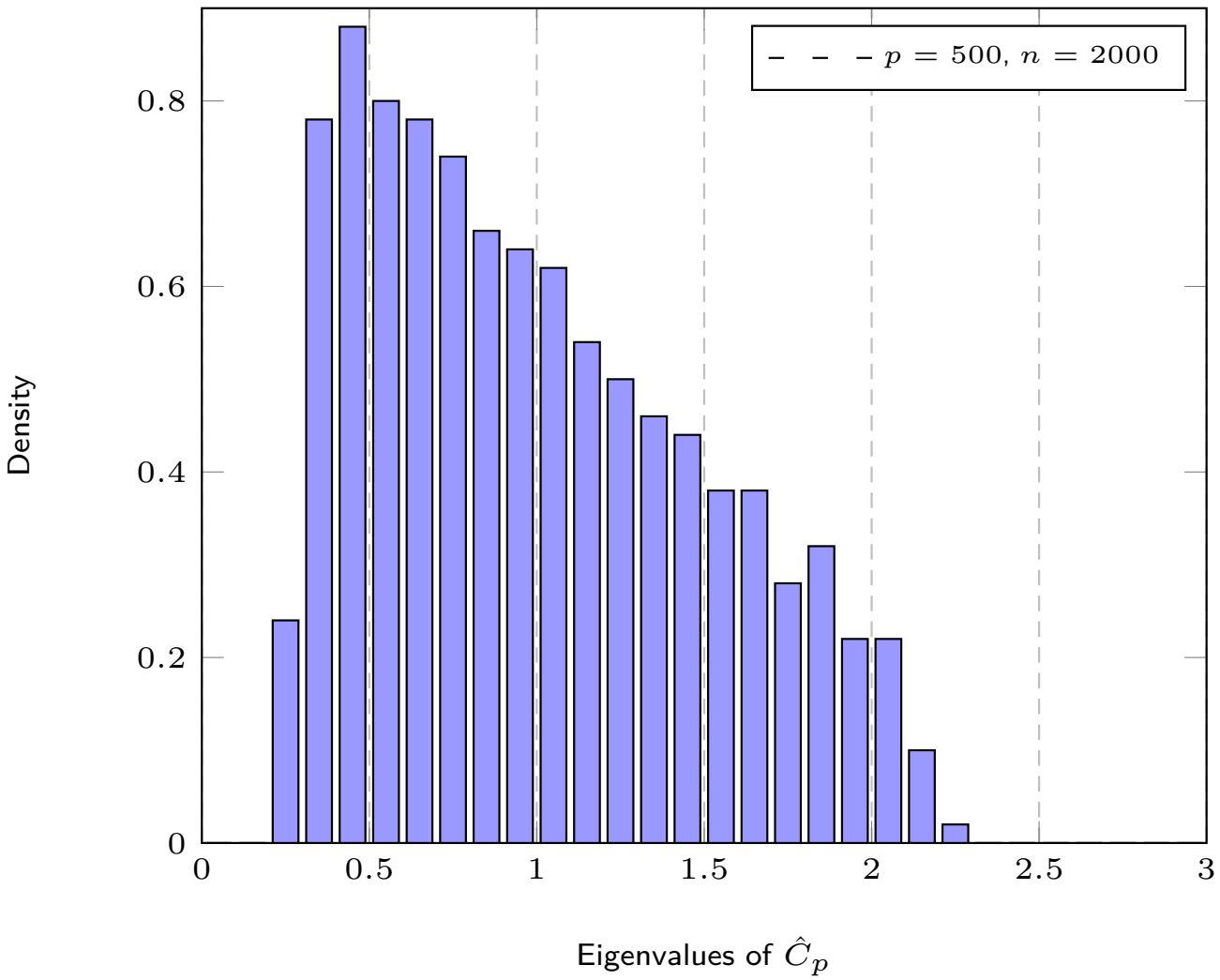


Figure: Histogram of the eigenvalues of $\hat{C}_p$ for $c = 1/4$, $C_p = I_p$.

The Marčenko–Pastur law

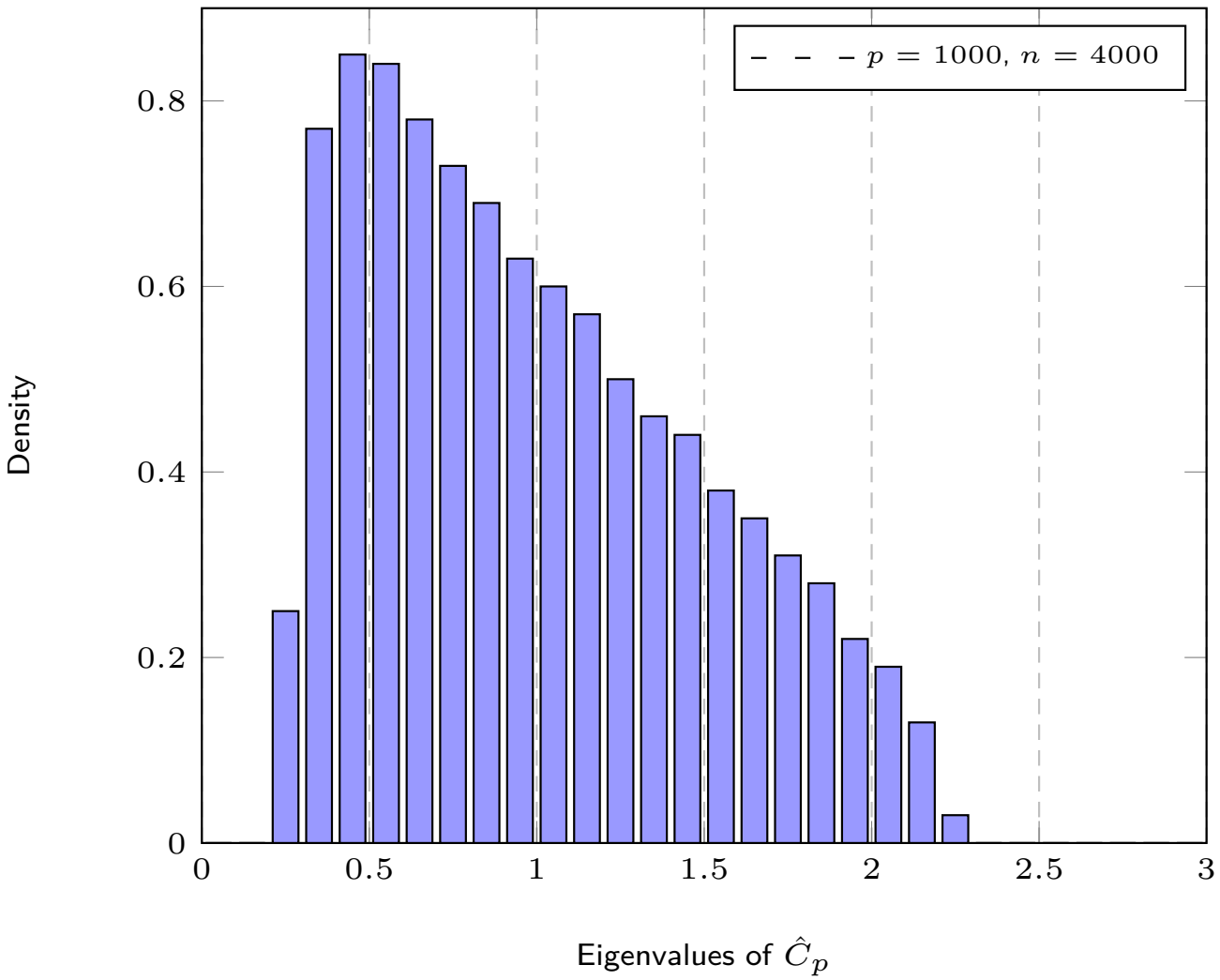


Figure: Histogram of the eigenvalues of $\hat{C}_p$ for $c = 1/4$, $C_p = I_p$.

The Marčenko–Pastur law

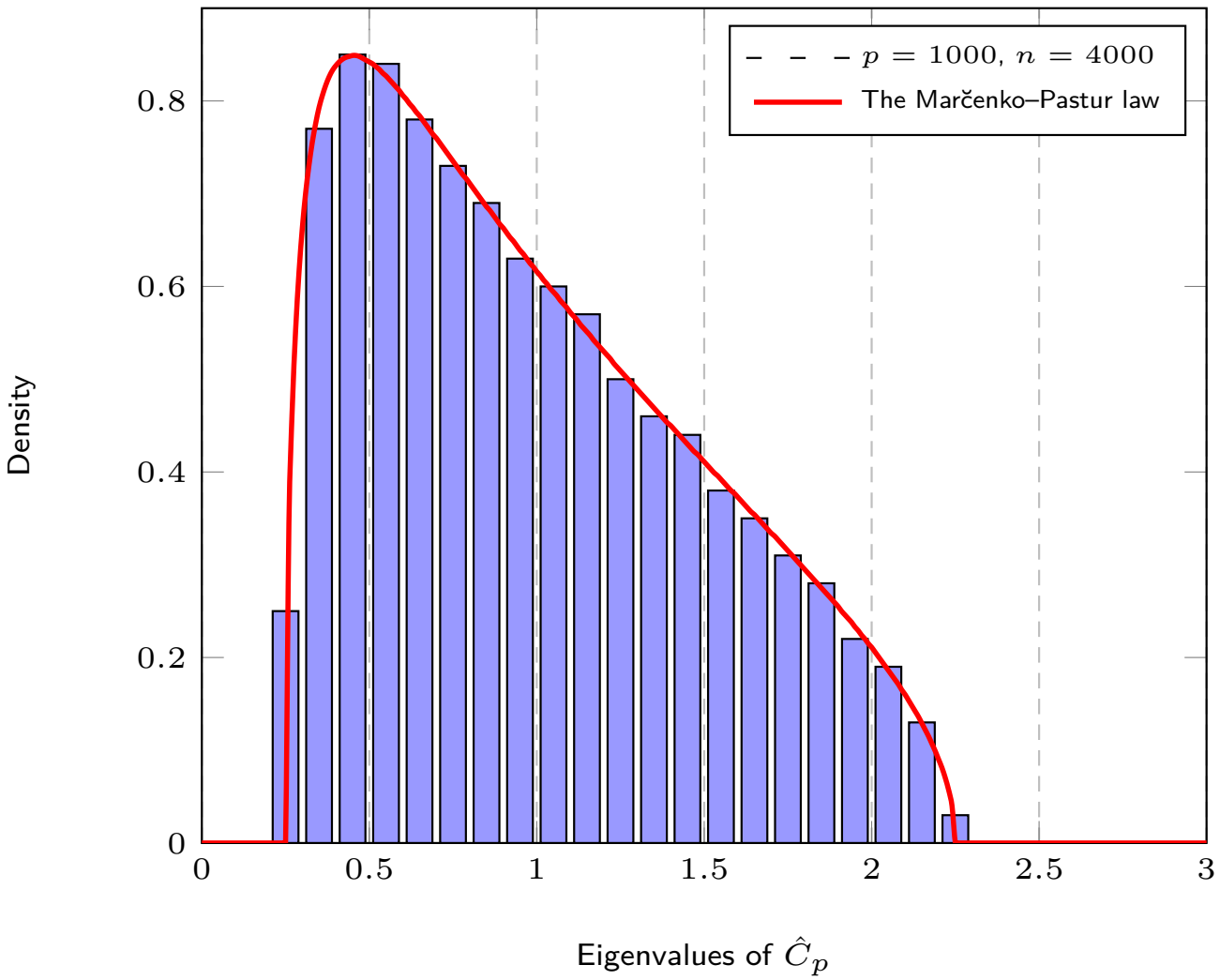


Figure: Histogram of the eigenvalues of $\hat{C}_p$ for $c = 1/4$, $C_p = I_p$.

Definition (Empirical Spectral Density)

Empirical spectral density (e.s.d.) μ_p of Hermitian matrix $A_p \in \mathbb{C}^{p \times p}$ is

$$\mu_p = \frac{1}{p} \sum_{i=1}^{p} \boldsymbol{\delta}_{\lambda_i(A_p)}.$$

The Marčenko–Pastur law

Definition (Empirical Spectral Density)

Empirical spectral density (e.s.d.) μ_p of Hermitian matrix $A_p \in \mathbb{C}^{p\times p}$ is

$$\mu_p = \frac{1}{p}\sum_{i=1}^{p} \boldsymbol{\delta}_{\lambda_i(A_p)}.$$

Theorem (Marčenko–Pastur Law **[Marčenko,Pastur'67]**)

$X_p \in \mathbb{C}^{p\times n}$ with i.i.d. zero mean, unit variance entries.
As $p, n \to \infty$ with $p/n \to c \in (0,\infty)$, e.s.d. μ_p of $\frac{1}{n}X_pX_p^*$ satisfies

$$\mu_p \xrightarrow{\text{a.s.}} \mu_c$$

weakly, where

- $\mu_c(\{0\}) = \max\{0, 1 - c^{-1}\}$

The Marčenko–Pastur law

Definition (Empirical Spectral Density)

Empirical spectral density (e.s.d.) μ_p of Hermitian matrix $A_p \in \mathbb{C}^{p\times p}$ is

$$\mu_p = \frac{1}{p}\sum_{i=1}^{p} \boldsymbol{\delta}_{\lambda_i(A_p)}.$$

Theorem (Marčenko–Pastur Law **[Marčenko,Pastur'67]**)

$X_p \in \mathbb{C}^{p\times n}$ with i.i.d. zero mean, unit variance entries.
As $p, n \to \infty$ with $p/n \to c \in (0,\infty)$, e.s.d. μ_p of $\frac{1}{n}X_pX_p^*$ satisfies

$$\mu_p \xrightarrow{\text{a.s.}} \mu_c$$

weakly, where

- $\mu_c(\{0\}) = \max\{0, 1 - c^{-1}\}$
- *on $(0,\infty)$, μ_c has continuous density f_c supported on $[(1-\sqrt{c})^2, (1+\sqrt{c})^2]$*

$$f_c(x) = \frac{1}{2\pi c x}\sqrt{(x-(1-\sqrt{c})^2)((1+\sqrt{c})^2 - x)}.$$

The Marčenko–Pastur law

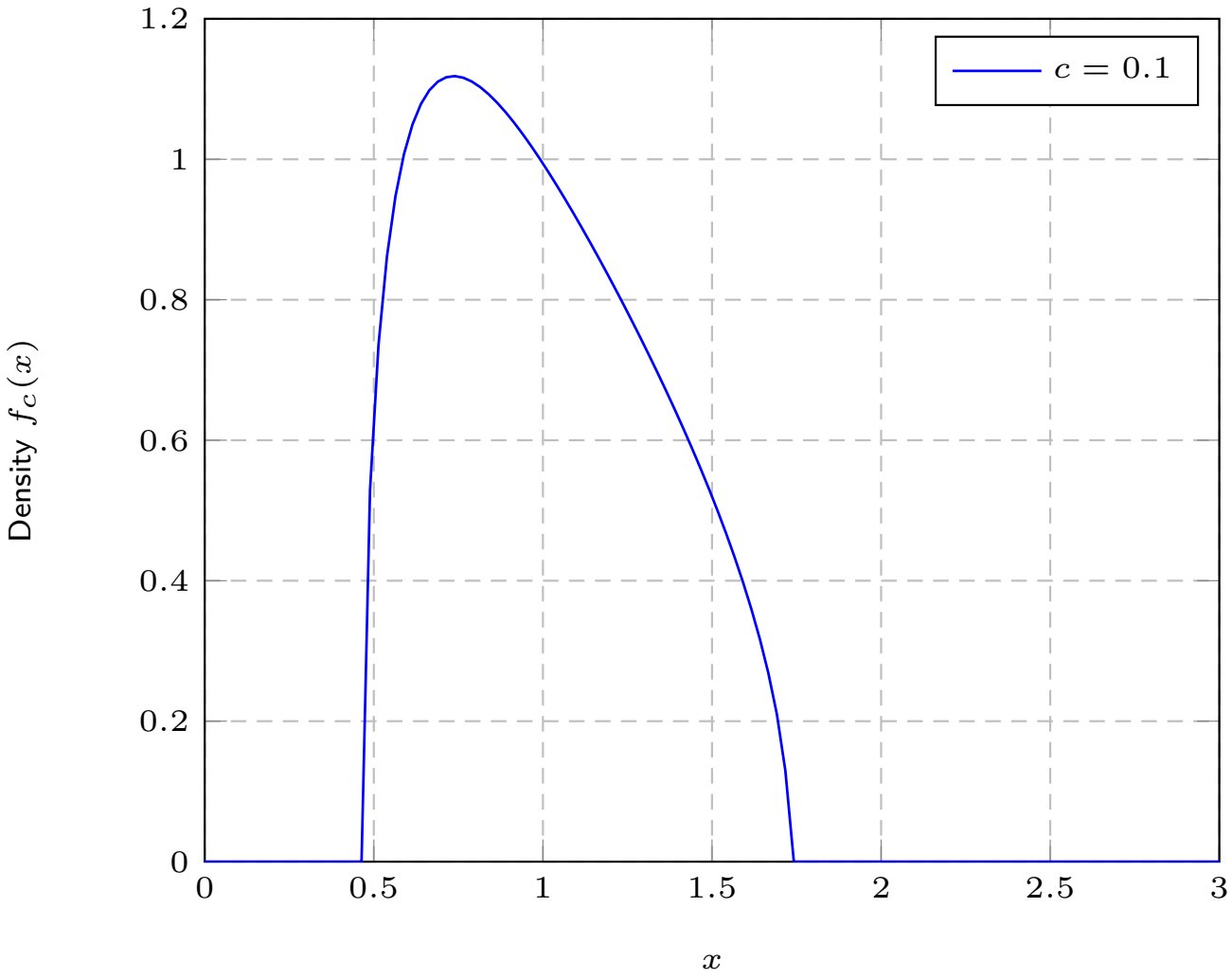


Figure: Marčenko-Pastur law for different limit ratios $c = \lim_{p\to\infty} p/n$.

The Marčenko–Pastur law

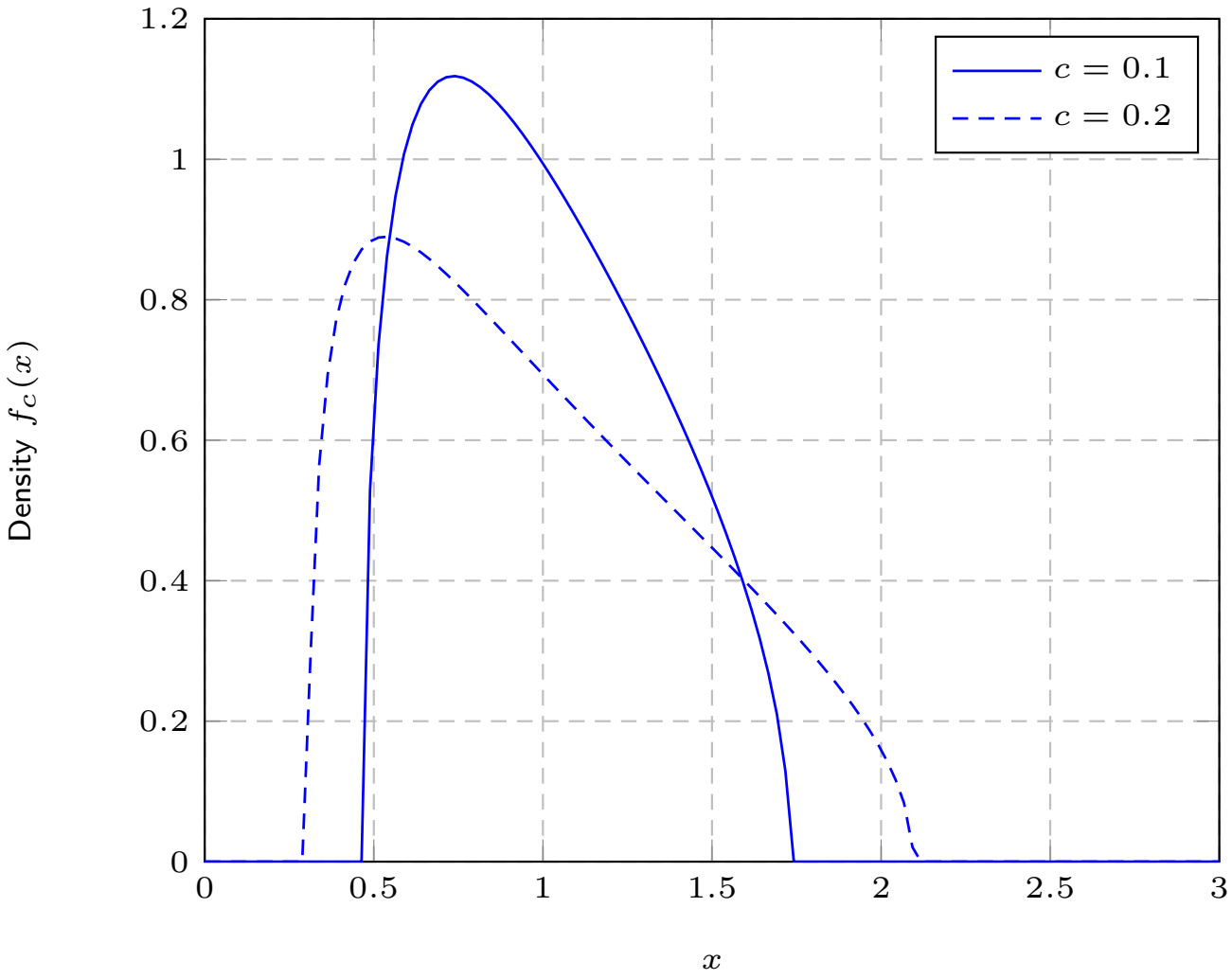


Figure: Marčenko-Pastur law for different limit ratios $c = \lim_{p\to\infty} p/n$.

The Marčenko–Pastur law

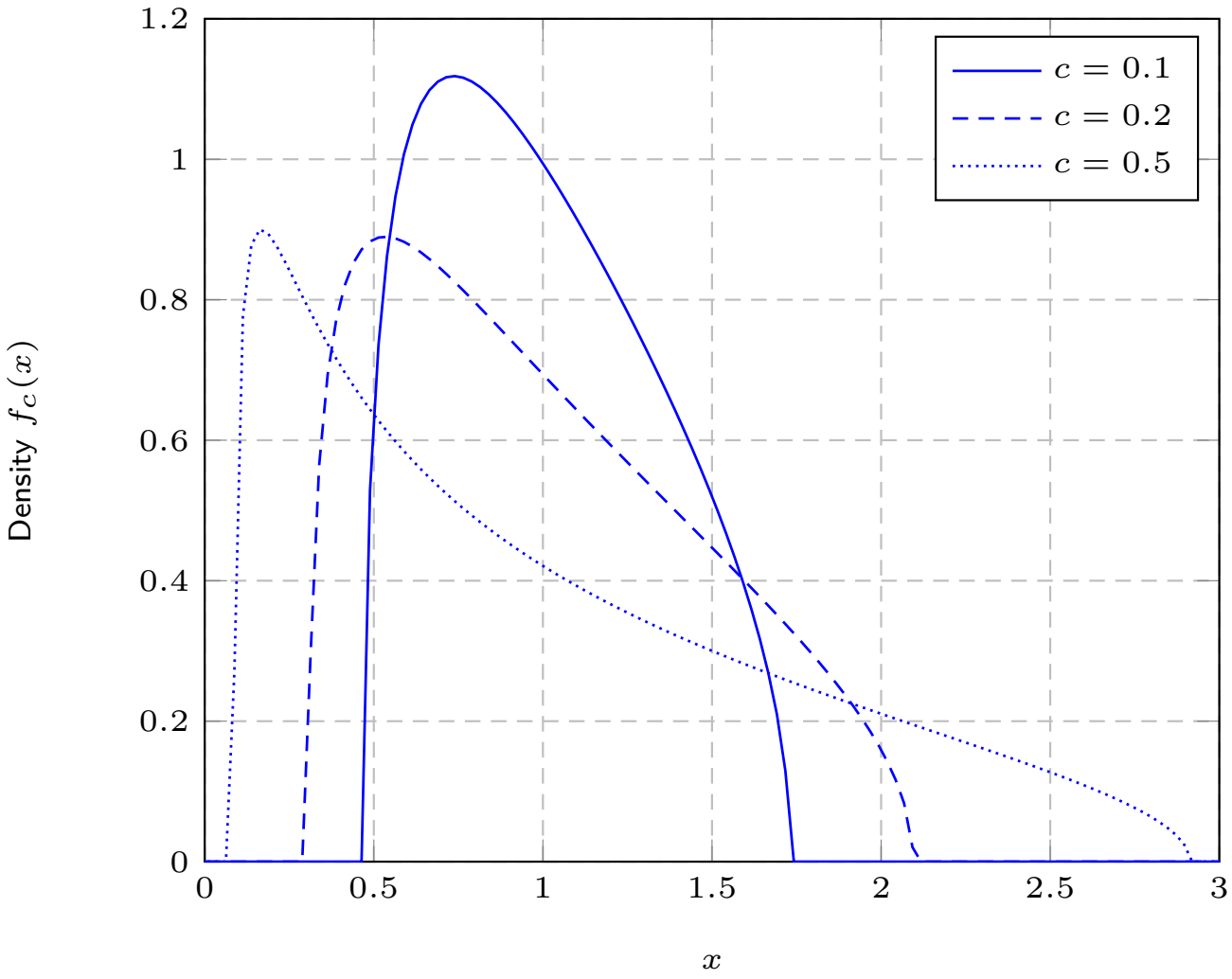


Figure: Marčenko-Pastur law for different limit ratios $c = \lim_{p\to\infty} p/n$.

Outline

Spiked Models

Small rank perturbation: $C_p = I_p + P$, P of low rank.

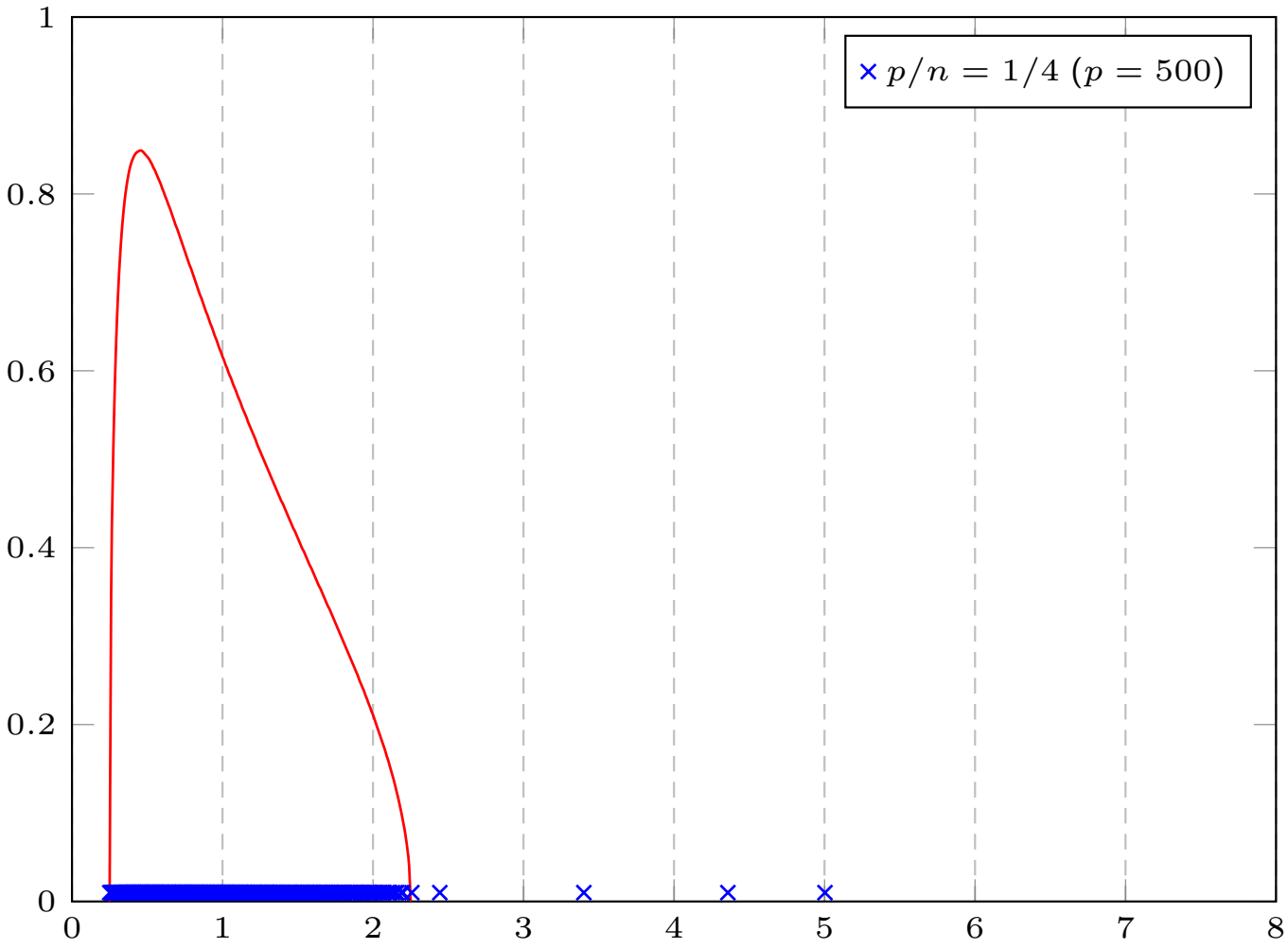


Figure: Eigenvalues of $\frac{1}{n} Y_p Y_p^\mathsf{T}$, $\mathrm{eig}(C_p) = \{\underbrace{1, \ldots, 1}_{p-4}, 2, 3, 4, 5\}$.

Spiked Models

Small rank perturbation: $C_p = I_p + P$, P of low rank.

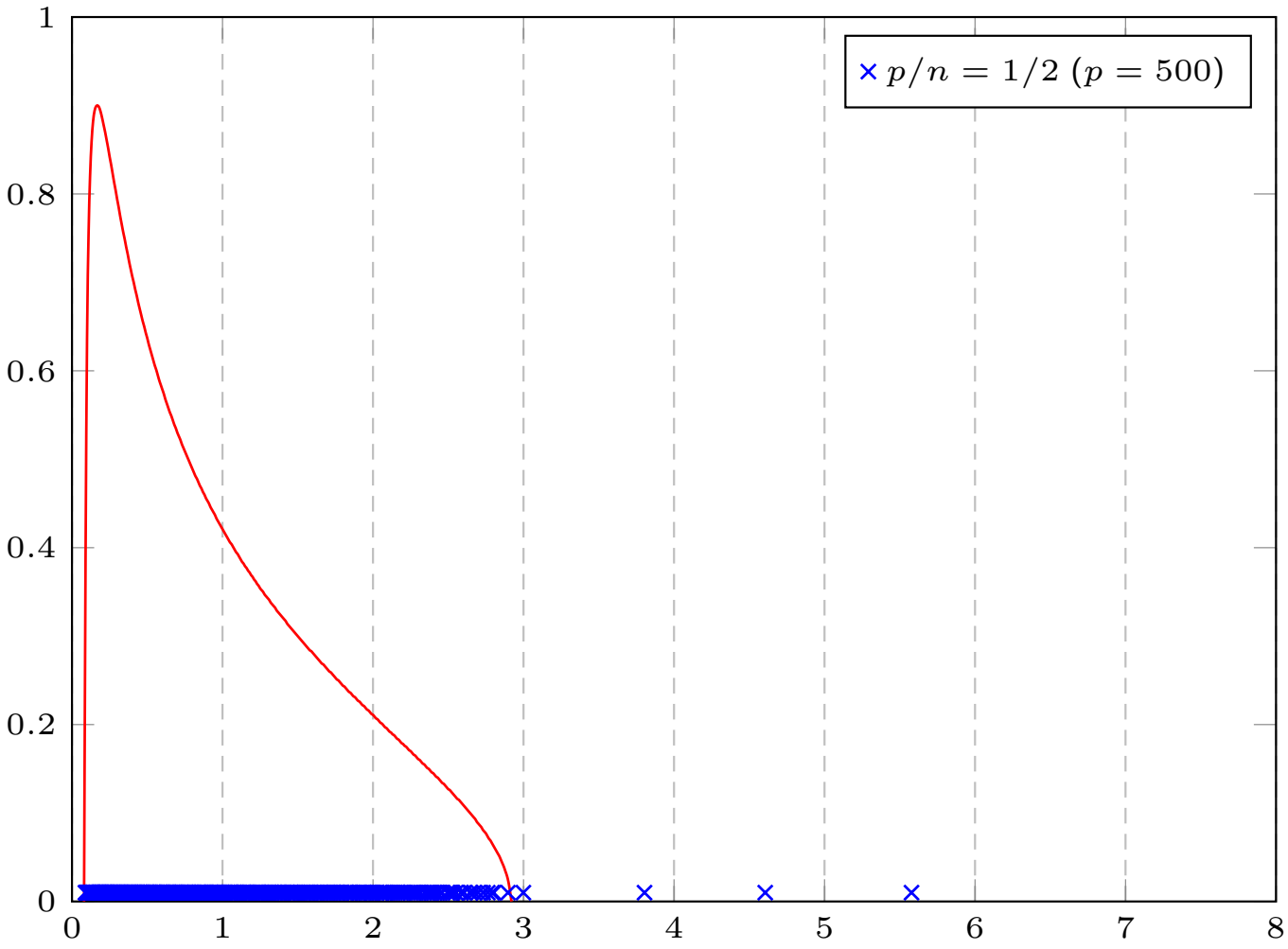


Figure: Eigenvalues of $\frac{1}{n} Y_p Y_p^\mathsf{T}$, $\mathrm{eig}(C_p) = \{\underbrace{1, \ldots, 1}_{p-4}, 2, 3, 4, 5\}$.

Spiked Models

Small rank perturbation: $C_p = I_p + P$, P of low rank.

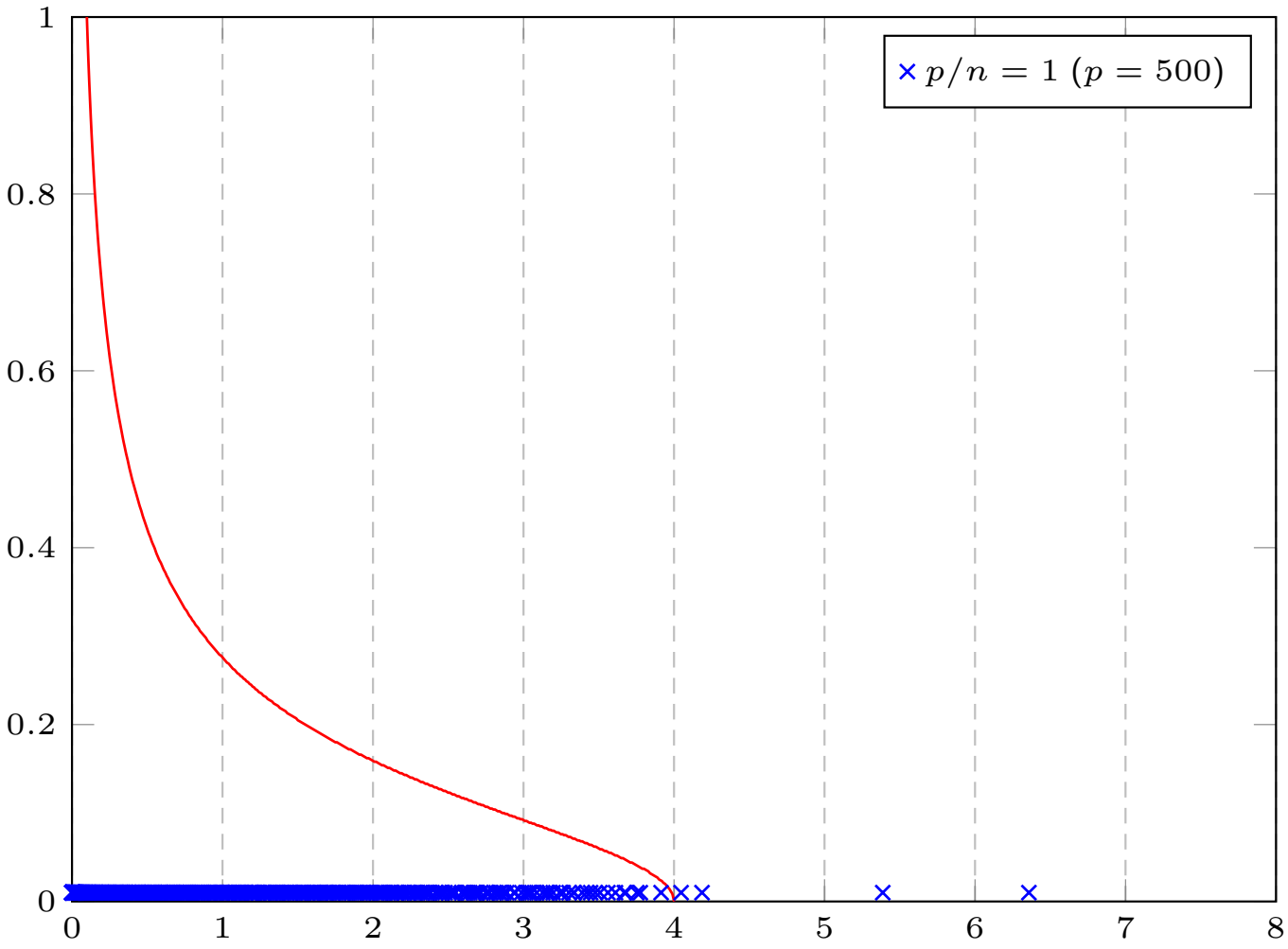


Figure: Eigenvalues of $\frac{1}{n} Y_p Y_p^{\mathsf{T}}$, $\mathrm{eig}(C_p) = \{\underbrace{1, \ldots, 1}_{p-4}, 2, 3, 4, 5\}$.

Spiked Models

Small rank perturbation: $C_p = I_p + P$, P of low rank.

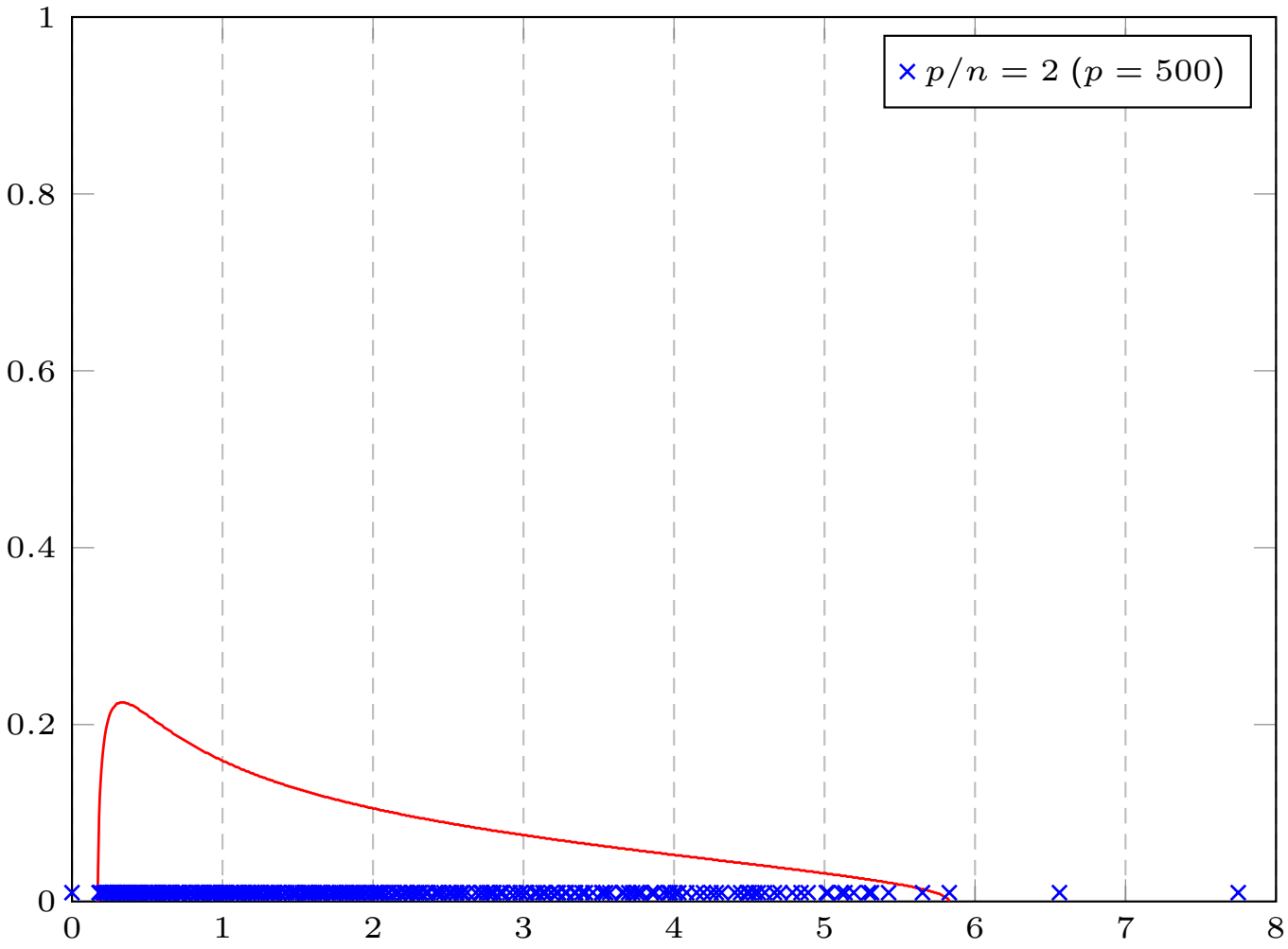


Figure: Eigenvalues of $\frac{1}{n} Y_p Y_p^{\mathsf{T}}$, $\mathrm{eig}(C_p) = \{\underbrace{1, \ldots, 1}_{p-4}, 2, 3, 4, 5\}$.

Spiked Models

Theorem (Eigenvalues [Baik,Silverstein'06])

Let $Y_p = C_p^{\frac{1}{2}} X_p$, *with*

- X_p *with i.i.d. zero mean, unit variance,* $E[|X_p|_{ij}^4] < \infty$.
- $C_p = I_p + P$, $P = U\Omega U^*$, *where, for* K *fixed,*

$$\Omega = \mathrm{diag}\,(\omega_1, \ldots, \omega_K) \in \mathbb{R}^{K \times K}, \text{ with } \omega_1 \geq \ldots \geq \omega_K > 0.$$

Theorem (Eigenvalues [Baik,Silverstein'06])

Let $Y_p = C_p^{\frac{1}{2}} X_p$, *with*

- X_p *with i.i.d. zero mean, unit variance,* $E[|X_p|_{ij}^4] < \infty$.
- $C_p = I_p + P$, $P = U\Omega U^*$, *where, for* K *fixed,*

$$\Omega = \operatorname{diag}(\omega_1, \ldots, \omega_K) \in \mathbb{R}^{K \times K}, \text{ with } \omega_1 \geq \ldots \geq \omega_K > 0.$$

Then, as $p, n \to \infty$, $p/n \to c \in (0, \infty)$, *denoting* $\lambda_m = \lambda_m(\frac{1}{n} Y_p Y_p^*)$ $(\lambda_m > \lambda_{m+1})$,

$$\lambda_m \xrightarrow{\text{a.s.}} \begin{cases} 1 + \omega_m + c\frac{1+\omega_m}{\omega_m} > (1+\sqrt{c})^2 & , \ \omega_m > \sqrt{c} \\ (1+\sqrt{c})^2 & , \ \omega_m \in (0, \sqrt{c}]. \end{cases}$$

Theorem (Eigenvectors [Paul'07])

Let $Y_p = C_p^{\frac{1}{2}} X_p$, *with*

- X_p *with i.i.d. zero mean, unit variance,* $E[|X_p|_{ij}^4] < \infty$.
- $C_p = I_p + P$, $P = U\Omega U^* = \sum_{i=1}^{K} \omega_i u_i u_i^*$, $\omega_1 > \ldots > \omega_M > 0$.

Theorem (Eigenvectors **[Paul'07]**)

Let $Y_p = C_p^{\frac{1}{2}} X_p$, *with*

- X_p *with i.i.d. zero mean, unit variance,* $E[|X_p|_{ij}^4] < \infty$.
- $C_p = I_p + P$, $P = U\Omega U^* = \sum_{i=1}^K \omega_i u_i u_i^*$, $\omega_1 > \ldots > \omega_M > 0$.

Then, as $p, n \to \infty$, $p/n \to c \in (0, \infty)$, *for* $a, b \in \mathbb{C}^p$ *deterministic and* $\hat{u}_i$ *eigenvector of* $\lambda_i(\frac{1}{n} Y_p Y_p^*)$,

$$a^* \hat{u}_i \hat{u}_i^* b - \frac{1 - c\omega_i^{-2}}{1 + c\omega_i^{-1}} a^* u_i u_i^* b \cdot 1_{\omega_i > \sqrt{c}} \xrightarrow{\text{a.s.}} 0$$

In particular,

$$|\hat{u}_i^* u_i|^2 \xrightarrow{\text{a.s.}} \frac{1 - c\omega_i^{-2}}{1 + c\omega_i^{-1}} \cdot 1_{\omega_i > \sqrt{c}}.$$

Spiked Models

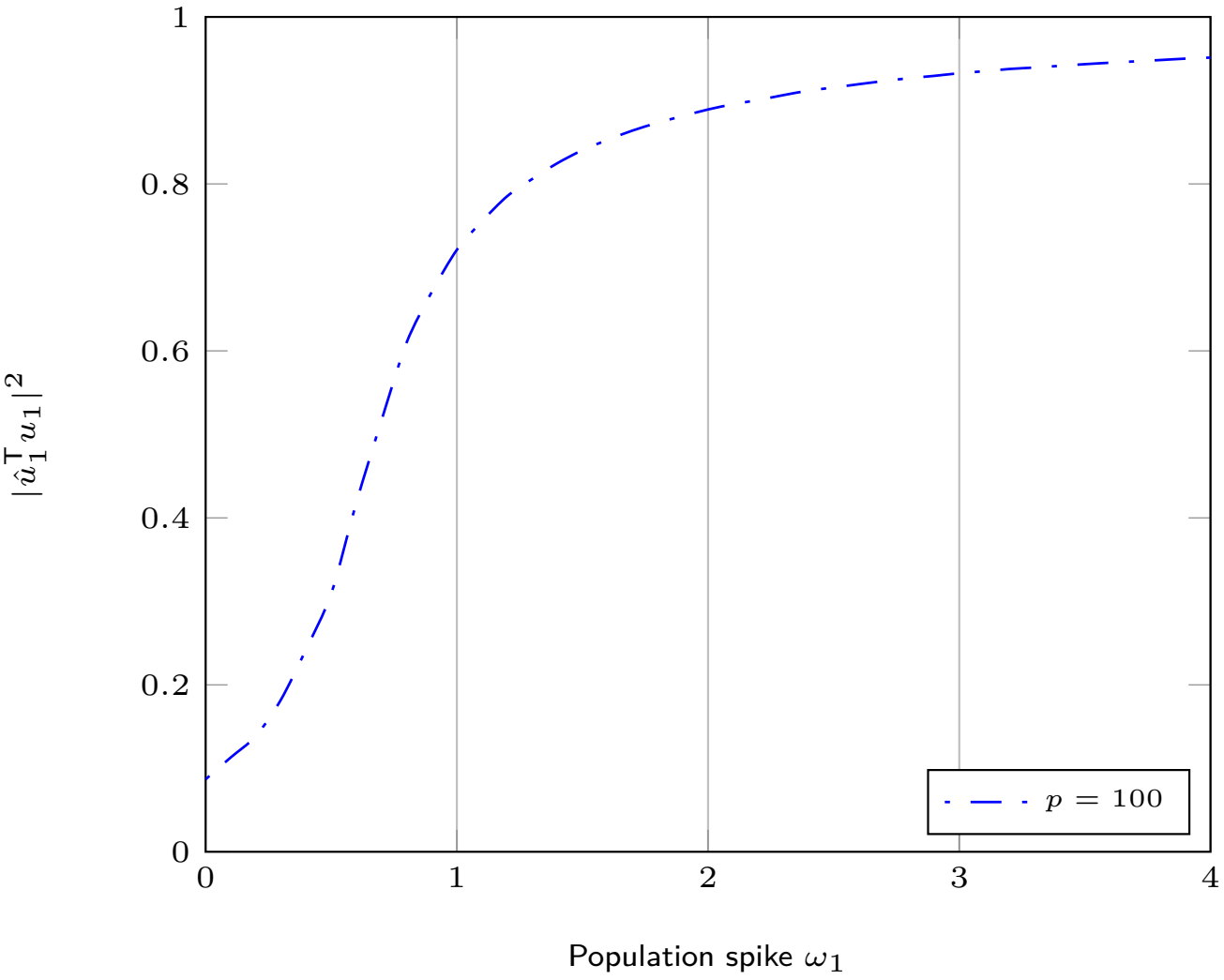


Figure: Simulated versus limiting $|\hat{u}_1^\mathsf{T} u_1|^2$ for $Y_p = C_p^{\frac{1}{2}} X_p$, $C_p = I_p + \omega_1 u_1 u_1^\mathsf{T}$, $p/n = 1/3$, varying ω_1.

Spiked Models

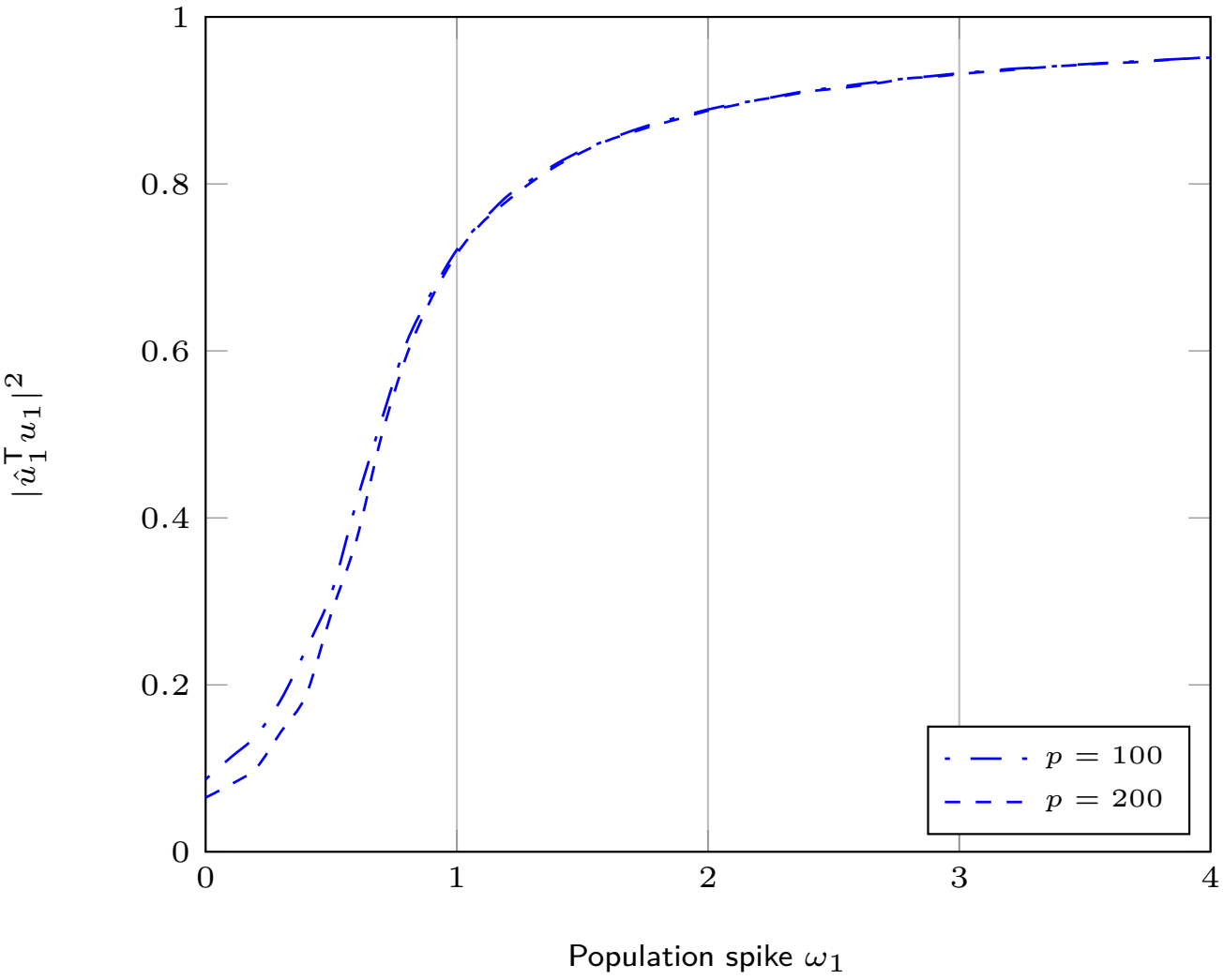


Figure: Simulated versus limiting $|\hat{u}_1^\mathsf{T} u_1|^2$ for $Y_p = C_p^{\frac{1}{2}} X_p$, $C_p = I_p + \omega_1 u_1 u_1^\mathsf{T}$, $p/n = 1/3$, varying ω_1.

Spiked Models

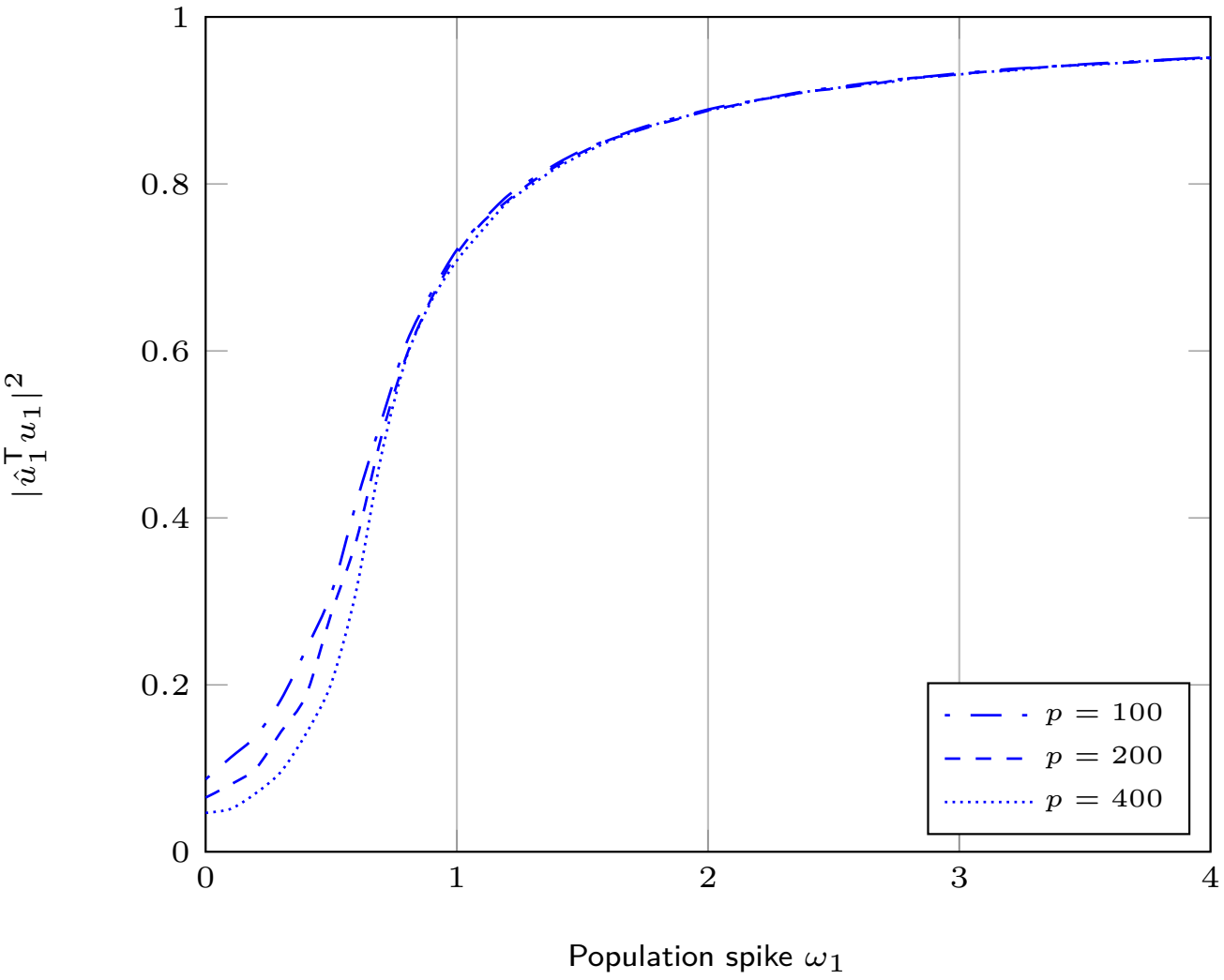


Figure: Simulated versus limiting $|\hat{u}_1^\mathsf{T} u_1|^2$ for $Y_p = C_p^{\frac{1}{2}} X_p$, $C_p = I_p + \omega_1 u_1 u_1^\mathsf{T}$, $p/n = 1/3$, varying ω_1.

Spiked Models

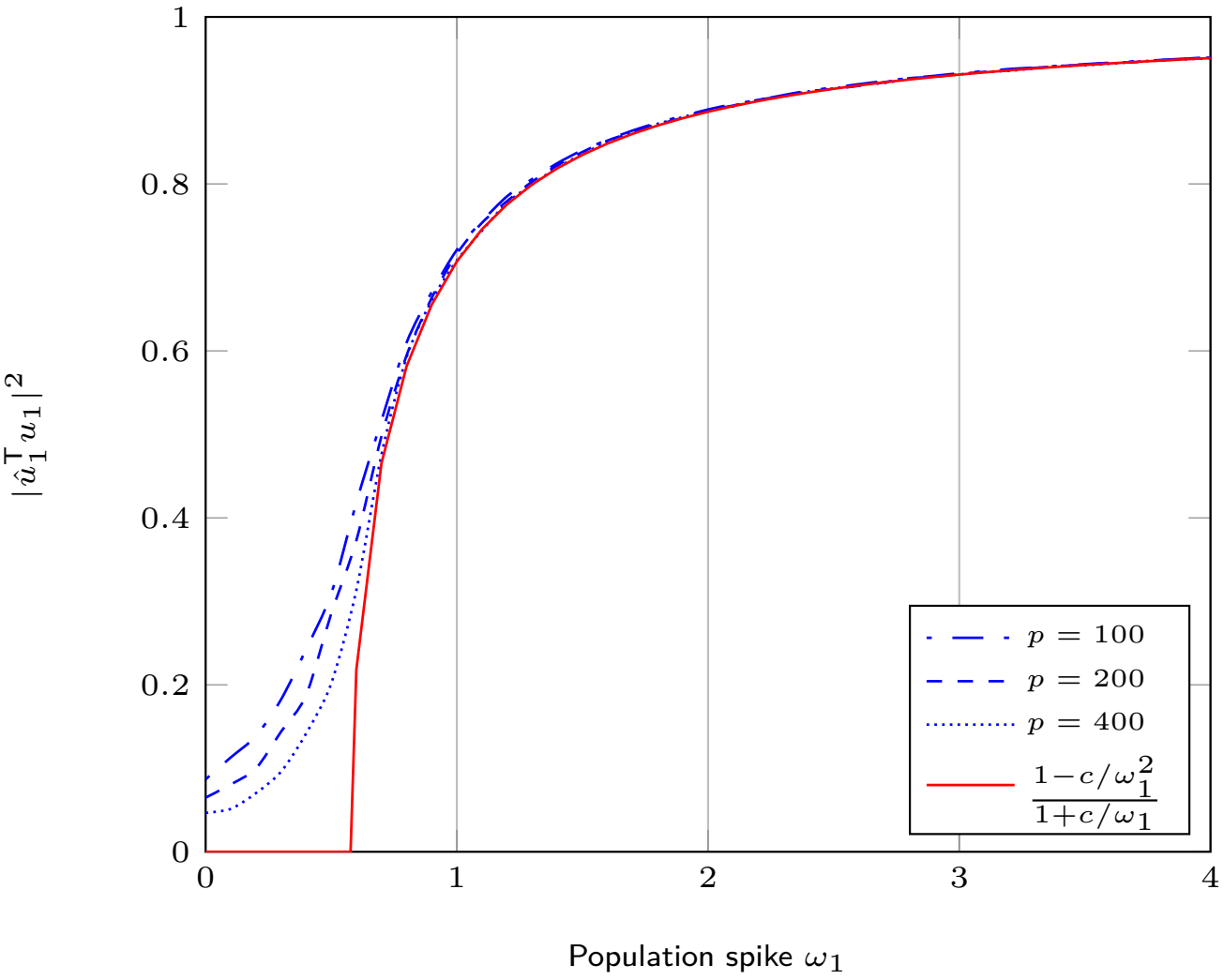


Figure: Simulated versus limiting $|\hat{u}_1^\mathsf{T} u_1|^2$ for $Y_p = C_p^{\frac{1}{2}} X_p$, $C_p = I_p + \omega_1 u_1 u_1^\mathsf{T}$, $p/n = 1/3$, varying ω_1.

Other Spiked Models

Similar results for multiple matrix models:

- $Y_p = \frac{1}{n}(I+P)^{\frac{1}{2}} X_p X_p^* (I+P)^{\frac{1}{2}}$
- $Y_p = \frac{1}{n} X_p X_p^* + P$
- $Y_p = \frac{1}{n} X_p^* (I+P) X$
- $Y_p = \frac{1}{n}(X_p + P)^*(X_p + P)$
- etc.

Outline

Takeaway Message 1

"RMT Explains Why Machine Learning Intuitions Collapse in Large Dimensions"

The curse of dimensionality and its consequences

Clustering setting in (not so) large n, p:

The curse of dimensionality and its consequences

Clustering setting in (not so) large n, p:

- GMM setting: $x_1^{(a)}, \ldots, x_{n_a}^{(a)} \sim \mathcal{N}(\mu_a, C_a)$, $a = 1, \ldots, k$

The curse of dimensionality and its consequences

Clustering setting in (not so) large n, p:

- GMM setting: $x_1^{(a)}, \ldots, x_{n_a}^{(a)} \sim \mathcal{N}(\mu_a, C_a)$, $a = 1, \ldots, k$
- Non-trivial task:

$$\|\mu_a - \mu_b\| = O(1), \quad \mathsf{tr}\,(C_a - C_b) = O(\sqrt{p}), \quad \mathsf{tr}\,[(C_a - C_b)^2] = O(p)$$

The curse of dimensionality and its consequences

Clustering setting in (not so) large n, p:

- GMM setting: $x_1^{(a)}, \ldots, x_{n_a}^{(a)} \sim \mathcal{N}(\mu_a, C_a)$, $a = 1, \ldots, k$
- Non-trivial task:

$$\|\mu_a - \mu_b\| = O(1), \quad \mathsf{tr}\,(C_a - C_b) = O(\sqrt{p}), \quad \mathsf{tr}\,[(C_a - C_b)^2] = O(p)$$

Classical method: spectral clustering

The curse of dimensionality and its consequences

Clustering setting in (not so) large n, p:

- GMM setting: $x_1^{(a)}, \ldots, x_{n_a}^{(a)} \sim \mathcal{N}(\mu_a, C_a)$, $a = 1, \ldots, k$
- Non-trivial task:

$$\|\mu_a - \mu_b\| = O(1), \quad \mathsf{tr}\,(C_a - C_b) = O(\sqrt{p}), \quad \mathsf{tr}\,[(C_a - C_b)^2] = O(p)$$

Classical method: spectral clustering

- Extract and cluster the dominant eigenvectors of

$$K = \{\kappa(x_i, x_j)\}_{i,j=1}^n$$

The curse of dimensionality and its consequences

Clustering setting in (not so) large n, p:

- GMM setting: $x_1^{(a)}, \ldots, x_{n_a}^{(a)} \sim \mathcal{N}(\mu_a, C_a)$, $a = 1, \ldots, k$
- Non-trivial task:

$$\|\mu_a - \mu_b\| = O(1), \quad \mathsf{tr}\,(C_a - C_b) = O(\sqrt{p}), \quad \mathsf{tr}\,[(C_a - C_b)^2] = O(p)$$

Classical method: spectral clustering

- Extract and cluster the dominant eigenvectors of

$$K = \{\kappa(x_i, x_j)\}_{i,j=1}^n, \quad \kappa(x_i, x_j) = f\left(\frac{1}{p}\|x_i - x_j\|^2\right).$$

The curse of dimensionality and its consequences

Clustering setting in (not so) large n, p:

- GMM setting: $x_1^{(a)}, \ldots, x_{n_a}^{(a)} \sim \mathcal{N}(\mu_a, C_a)$, $a = 1, \ldots, k$
- Non-trivial task:

$$\|\mu_a - \mu_b\| = O(1), \quad \mathsf{tr}\,(C_a - C_b) = O(\sqrt{p}), \quad \mathsf{tr}\,[(C_a - C_b)^2] = O(p)$$

Classical method: spectral clustering

- Extract and cluster the dominant eigenvectors of

$$K = \{\kappa(x_i, x_j)\}_{i,j=1}^n, \quad \kappa(x_i, x_j) = f\left(\frac{1}{p}\|x_i - x_j\|^2\right).$$

- Why? **Finite-dimensional intuition**

$$K = \begin{pmatrix} \kappa(x_i,x_j) \gg 1 & \kappa(x_i,x_j) \ll 1 & \kappa(x_i,x_j) \ll 1 \\ \kappa(x_i,x_j) \ll 1 & \kappa(x_i,x_j) \gg 1 & \kappa(x_i,x_j) \ll 1 \\ \kappa(x_i,x_j) \ll 1 & \kappa(x_i,x_j) \ll 1 & \kappa(x_i,x_j) \gg 1 \end{pmatrix} \begin{array}{c} \mathcal{C}_1 \\ \mathcal{C}_2 \\ \mathcal{C}_3 \end{array}$$

The curse of dimensionality and its consequences (2)

In reality, here is what happens...

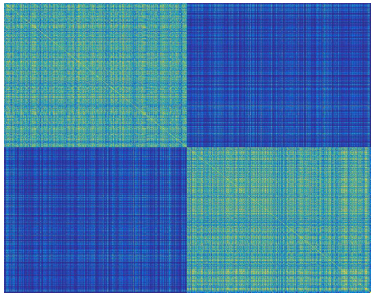
Kernel $K_{ij} = \exp(-\frac{1}{2p}\|x_i - x_j\|^2)$ and second eigenvector v_2 ($x_i \sim \mathcal{N}(\pm\mu, I_p)$, $\mu = (2, 0, \ldots, 0)^\mathsf{T} \in \mathbb{R}^p$).

The curse of dimensionality and its consequences (2)

In reality, here is what happens...

Kernel $K_{ij} = \exp(-\frac{1}{2p}\|x_i - x_j\|^2)$ and second eigenvector v_2 ($x_i \sim \mathcal{N}(\pm\mu, I_p)$, $\mu = (2, 0, \ldots, 0)^\mathsf{T} \in \mathbb{R}^p$).

$p = 4$, $n = 1000$

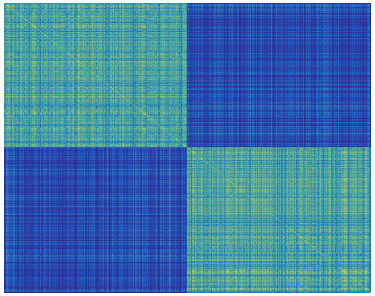
$K =$  $v_2 =$ 

The curse of dimensionality and its consequences (2)

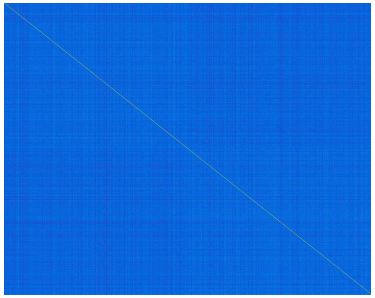
In reality, here is what happens...

Kernel $K_{ij} = \exp(-\frac{1}{2p}\|x_i - x_j\|^2)$ and second eigenvector v_2
($x_i \sim \mathcal{N}(\pm\mu, I_p)$, $\mu = (2, 0, \ldots, 0)^\mathsf{T} \in \mathbb{R}^p$).

$p = 4$, $n = 1000$

$K =$  $v_2 =$ 

$p = 400$, $n = 1000$

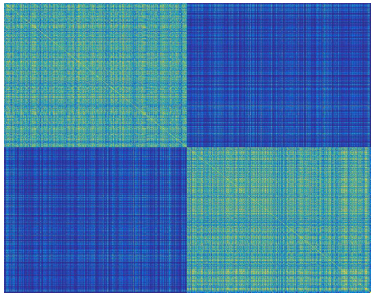
$K =$  $v_2 =$ 

The curse of dimensionality and its consequences (2)

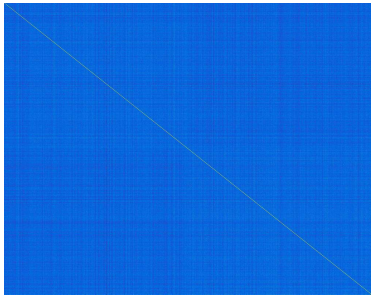
In reality, here is what happens...

Kernel $K_{ij} = \exp(-\frac{1}{2p}\|x_i - x_j\|^2)$ and second eigenvector v_2
($x_i \sim \mathcal{N}(\pm\mu, I_p)$, $\mu = (2, 0, \ldots, 0)^\mathsf{T} \in \mathbb{R}^p$).

$p = 4$, $n = 1000$

$K =$  $v_2 =$ 

$p = 400$, $n = 1000$

$K =$  $v_2 =$ 

Key observation: Under growth rate assumptions,

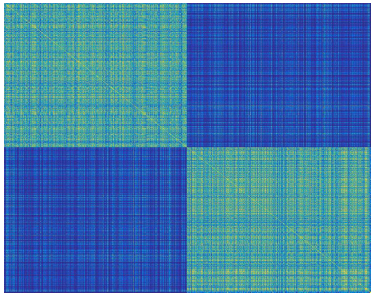
$$\boxed{\max_{1 \le i \ne j \le n} \left\{ \left| \frac{1}{p} \|x_i - x_j\|^2 - \tau \right| \right\} \xrightarrow{\text{a.s.}} 0}, \quad \tau = \frac{2}{p} \sum_{i=1}^{k} \mathsf{tr}\, \frac{n_a}{n} C_a.$$

The curse of dimensionality and its consequences (2)

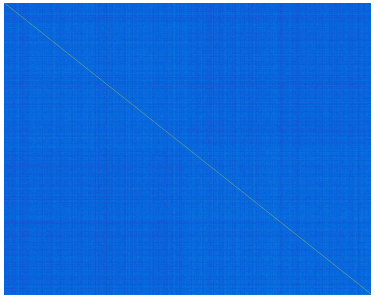
In reality, here is what happens...

Kernel $K_{ij} = \exp(-\frac{1}{2p}\|x_i - x_j\|^2)$ and second eigenvector v_2
($x_i \sim \mathcal{N}(\pm\mu, I_p)$, $\mu = (2, 0, \ldots, 0)^\mathsf{T} \in \mathbb{R}^p$).

$p = 4$, $n = 1000$

$K =$  $v_2 =$ 

$p = 400$, $n = 1000$

$K =$  $v_2 =$ 

Key observation: Under growth rate assumptions,

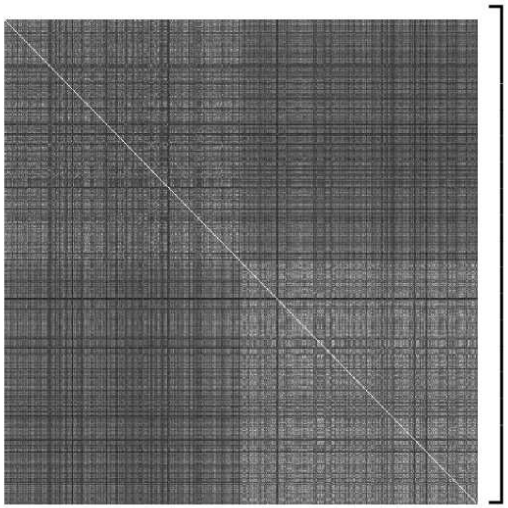
$$\boxed{\max_{1 \le i \ne j \le n} \left\{ \left| \frac{1}{p} \|x_i - x_j\|^2 - \tau \right| \right\} \xrightarrow{\text{a.s.}} 0}, \quad \tau = \frac{2}{p} \sum_{i=1}^{k} \mathsf{tr}\, \frac{n_a}{n} C_a.$$

- this suggests $K \simeq f(\tau) 1_n 1_n^\mathsf{T}$!

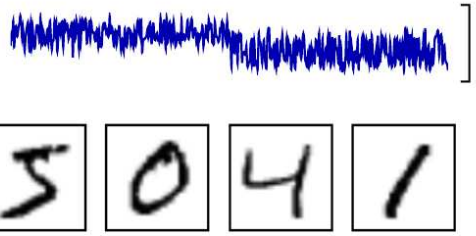
The curse of dimensionality and its consequences (3)

MNIST
raw
$p = 784,\ n = 500$
$\downarrow$

$\mathbf{K} =$



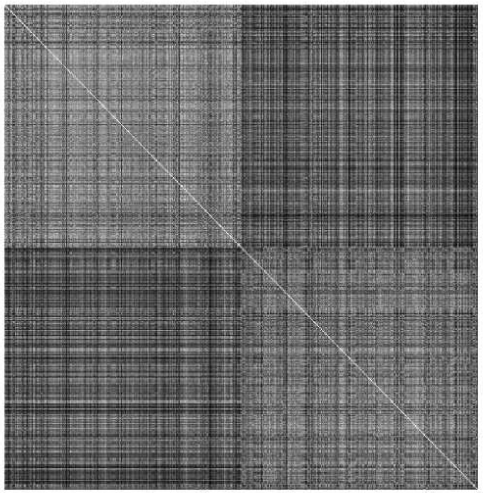
$\mathbf{v}_2 =$



(ici, classes “5” et “0”)

ImageNet
VGG-features
$p = 3084,\ n = 500$
$\downarrow$

$\mathbf{K} =$

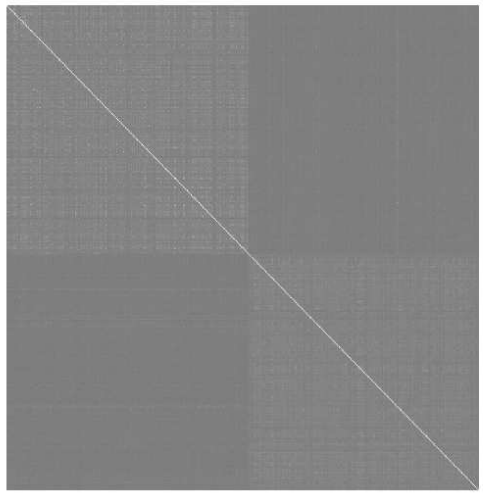


$\mathbf{v}_2 =$

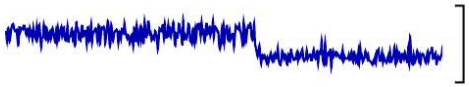
(ici, classes “bird” et “plane”)

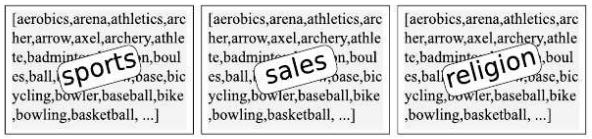
20NewsGroup
BERT embedding
$p = 300,\ n = 500$
$\downarrow$

$\mathbf{K} =$



$\mathbf{v}_2 =$





(ici, classes “sports” et “sales”)

The curse of dimensionality and its consequences (4)

(Major) consequences:

- Most **machine learning intuitions collapse**

The curse of dimensionality and its consequences (4)

(Major) consequences:

- Most **machine learning intuitions collapse**
- **But luckily**, concentration of distances allows for Taylor expansion, linearization...

The curse of dimensionality and its consequences (4)

(Major) consequences:

- Most **machine learning intuitions collapse**
- **But luckily**, concentration of distances allows for Taylor expansion, linearization...

Theorem (**[C-Benaych'16]** Asymptotic Kernel Behavior)

Under growth rate assumptions, as $p, n \to \infty$,

$$\left\| K - \hat{K} \right\| \xrightarrow{\text{a.s.}} 0, \quad \hat{K} \simeq \underbrace{f(\tau) 1_n 1_n^\mathsf{T}}_{O_{\|\cdot\|}(n)}$$

The curse of dimensionality and its consequences (4)

(Major) consequences:

- Most **machine learning intuitions collapse**
- **But luckily**, concentration of distances allows for Taylor expansion, linearization...

Theorem (**[C-Benaych'16]** Asymptotic Kernel Behavior)

Under growth rate assumptions, as $p, n \to \infty$,

$$\left\| K - \hat{K} \right\| \xrightarrow{\text{a.s.}} 0, \quad \hat{K} \simeq \underbrace{f(\tau) 1_n 1_n^\mathsf{T}}_{O_{\|\cdot\|}(n)} + \frac{1}{p} Z Z^\mathsf{T} + J A J^\mathsf{T} + *$$

The curse of dimensionality and its consequences (4)

(Major) consequences:

- Most **machine learning intuitions collapse**
- **But luckily**, concentration of distances allows for Taylor expansion, linearization...

Theorem (**[C-Benaych'16]** Asymptotic Kernel Behavior)

Under growth rate assumptions, as $p, n \to \infty$,

$$\left\| K - \hat{K} \right\| \xrightarrow{\text{a.s.}} 0, \quad \hat{K} \simeq \underbrace{f(\tau) 1_n 1_n^\mathsf{T}}_{O_{\|\cdot\|}(n)} + \frac{1}{p} Z Z^\mathsf{T} + J A J^\mathsf{T} + *$$

with $J = [j_1, \ldots, j_k] \in \mathbb{R}^{n \times k}$, $j_a = (0, 1_{n_a}, 0)^\mathsf{T}$ *(the clusters!)*

The curse of dimensionality and its consequences (4)

(Major) consequences:

- Most **machine learning intuitions collapse**
- **But luckily**, concentration of distances allows for Taylor expansion, linearization...

Theorem ([C-Benaych'16] Asymptotic Kernel Behavior)

Under growth rate assumptions, as $p, n \to \infty$,

$$\left\| K - \hat{K} \right\| \xrightarrow{\text{a.s.}} 0, \quad \hat{K} \simeq \underbrace{f(\tau) 1_n 1_n^\mathsf{T}}_{O_{\|\cdot\|}(n)} + \frac{1}{p} Z Z^\mathsf{T} + J A J^\mathsf{T} + *$$

with $J = [j_1, \ldots, j_k] \in \mathbb{R}^{n \times k}$, $j_a = (0, 1_{n_a}, 0)^\mathsf{T}$ *(the clusters!) and* $A \in \mathbb{R}^{k \times k}$ *function of:*

- $f(\tau)$, $f'(\tau)$, $f''(\tau)$
- $\|\mu_a - \mu_b\|$, $\mathrm{tr}(C_a - C_b)$, $\mathrm{tr}((C_a - C_b)^2)$, *for* $a, b \in \{1, \ldots, k\}$.

The curse of dimensionality and its consequences (4)

(Major) consequences:

- Most **machine learning intuitions collapse**
- **But luckily**, concentration of distances allows for Taylor expansion, linearization...

Theorem ([C-Benaych'16] Asymptotic Kernel Behavior)

Under growth rate assumptions, as $p, n \to \infty$,

$$\left\| K - \hat{K} \right\| \xrightarrow{\text{a.s.}} 0, \quad \hat{K} \simeq \underbrace{f(\tau) 1_n 1_n^\mathsf{T}}_{O_{\|\cdot\|}(n)} + \frac{1}{p} Z Z^\mathsf{T} + J A J^\mathsf{T} + *$$

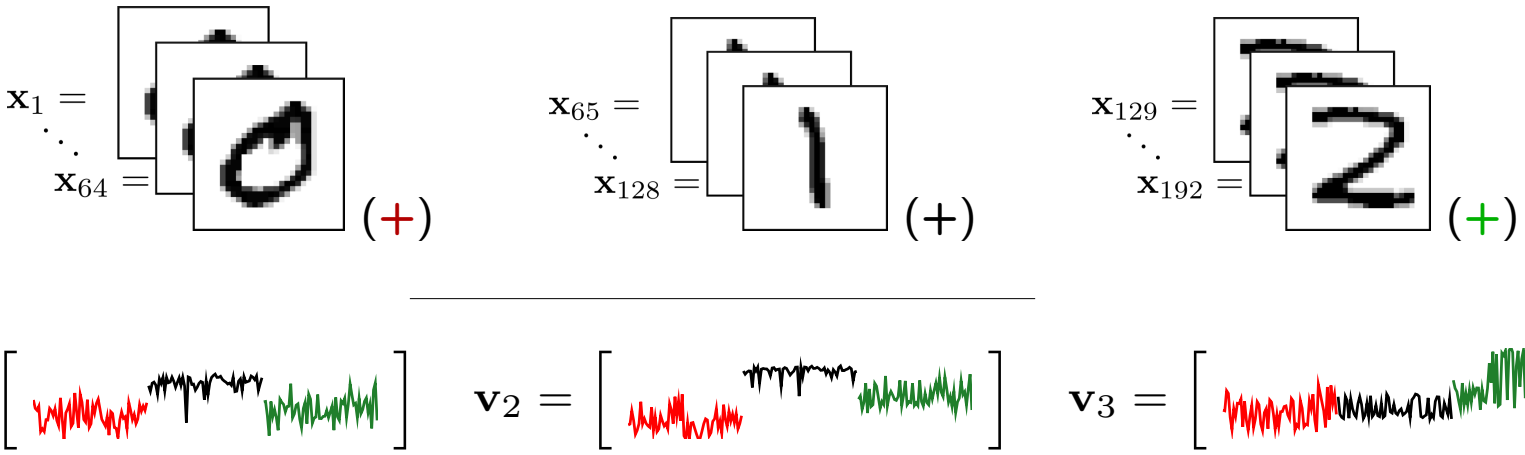
with $J = [j_1, \ldots, j_k] \in \mathbb{R}^{n \times k}$, $j_a = (0, 1_{n_a}, 0)^\mathsf{T}$ (the clusters!) and $A \in \mathbb{R}^{k \times k}$ function of:

- $f(\tau)$, $f'(\tau)$, $f''(\tau)$
- $\|\mu_a - \mu_b\|$, $\mathrm{tr}(C_a - C_b)$, $\mathrm{tr}((C_a - C_b)^2)$, *for* $a, b \in \{1, \ldots, k\}$.

➪ **This is a spiked model! We can study it fully!**

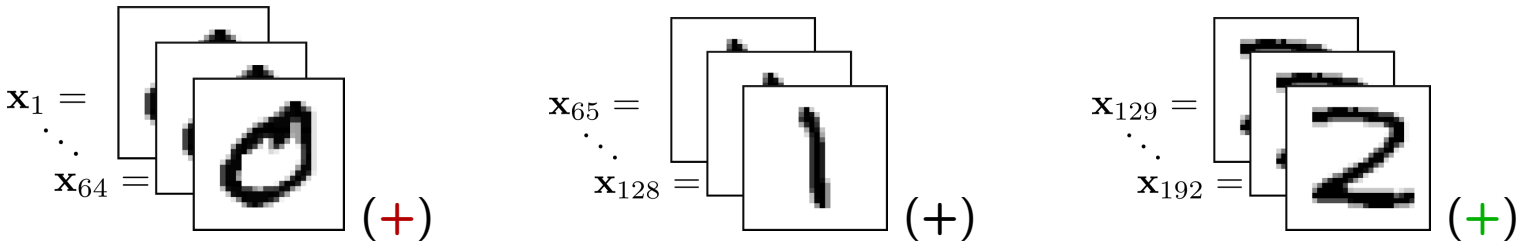
Performance prediction: spectral clustering

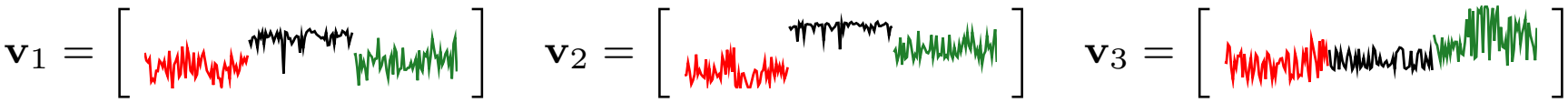
- **Asymptotic analysis of eigenvectors of** K: (MNIST, $p = 28 \times 28 (= 784)$)

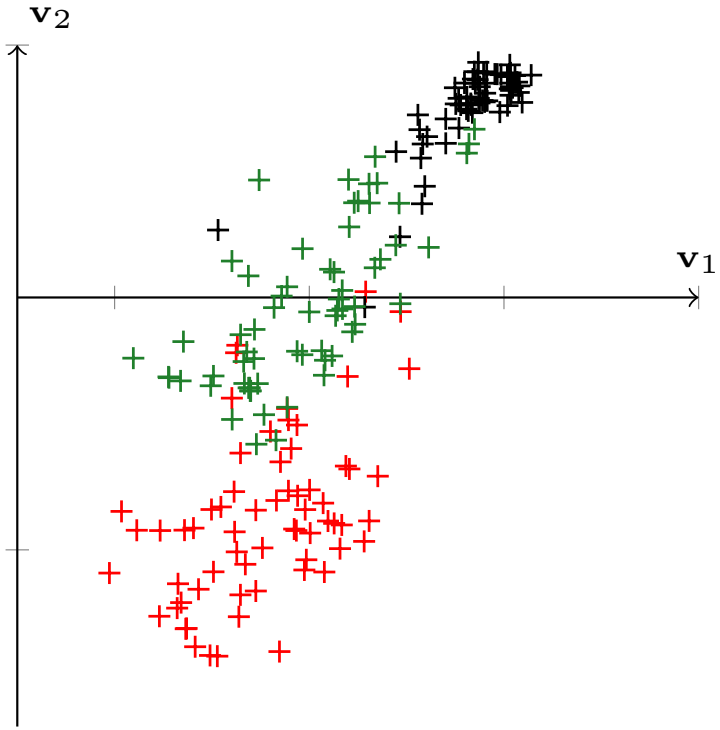


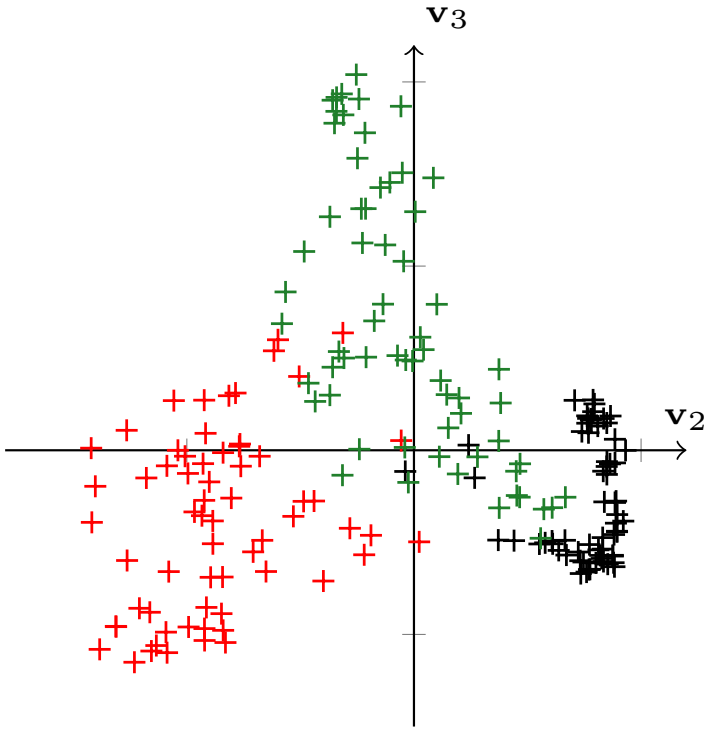
Performance prediction: spectral clustering

- **Asymptotic analysis of eigenvectors of** K: (MNIST, $p = 28 \times 28 (= 784)$)



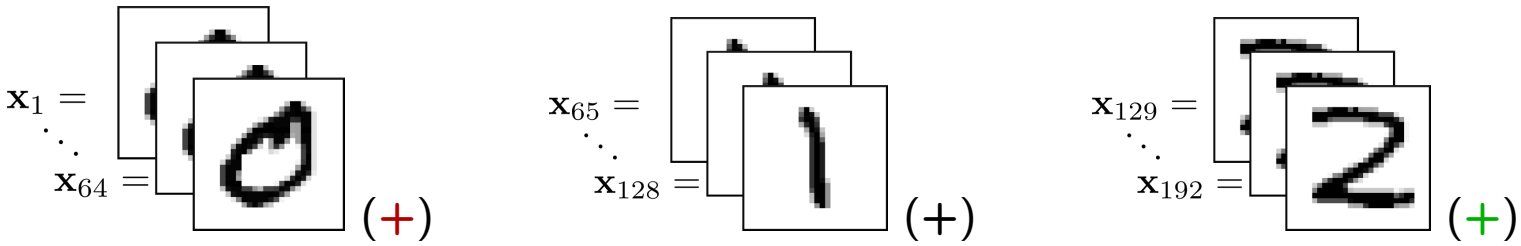


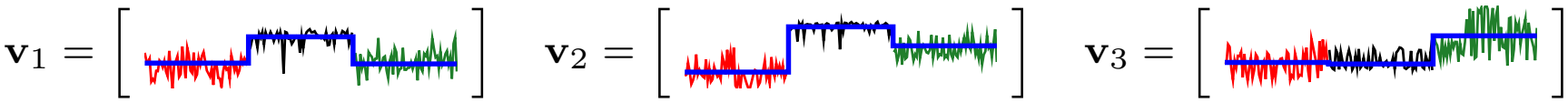


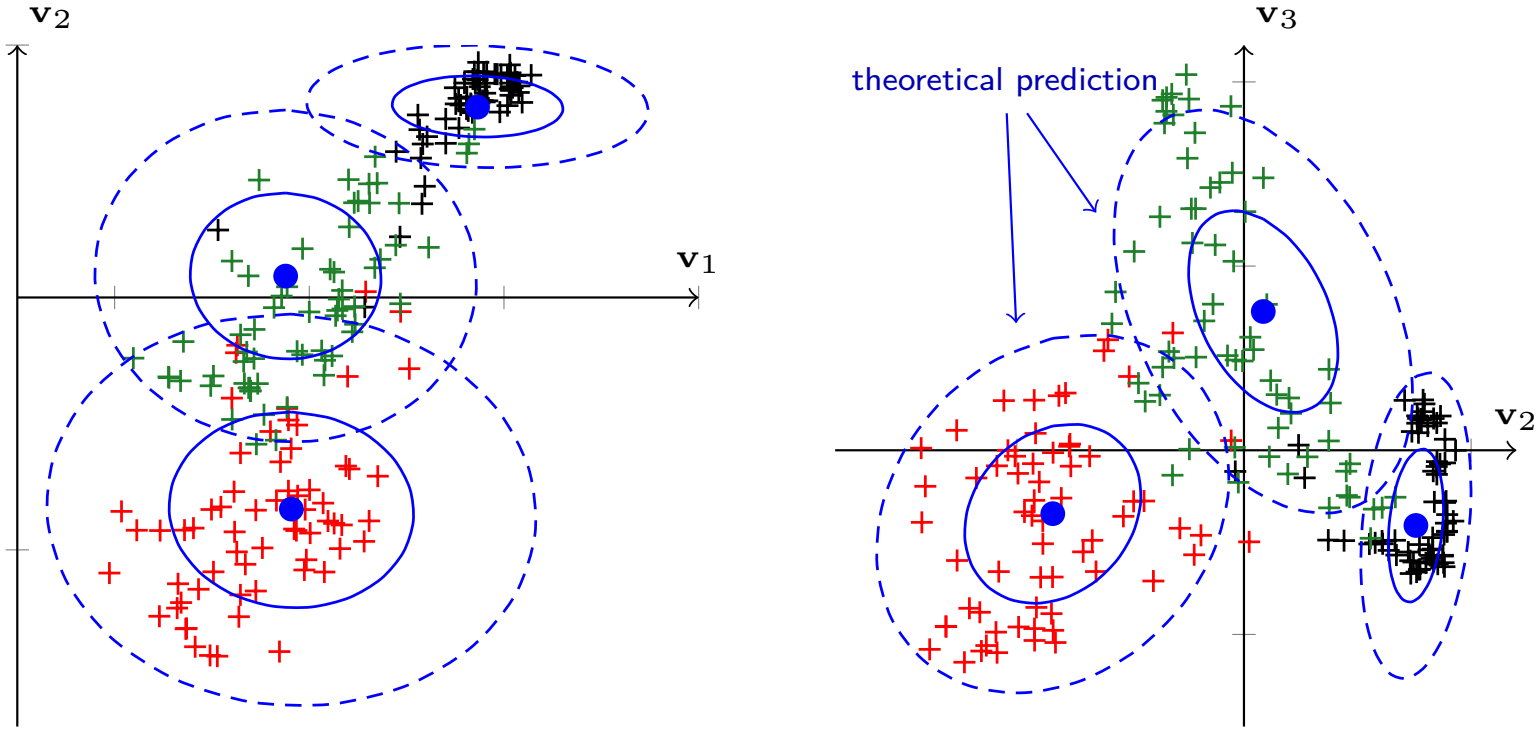


Performance prediction: spectral clustering

- **Asymptotic analysis of eigenvectors of** K: (MNIST, $p = 28 \times 28 (= 784)$)







Takeaway Message 2

"RMT Reassesses and Improves Data Processing"

Improving Kernel Spectral Clustering

- **Going further than ([Kammoun,Couillet'17])**,

$$K \simeq \underbrace{f(\tau)1_n 1_n^\mathsf{T}}_{O_{\|\cdot\|}(n)} + f'(\tau)\frac{1}{p}ZZ^\mathsf{T} + JAJ^\mathsf{T}, \text{ avec } A = F\left(\begin{array}{c} f(\tau), f'(\tau), f''(\tau) \\ \|\mu_a - \mu_b\|, \mathsf{tr}\,(C_a - C_b), \ldots \end{array}\right).$$

Improving Kernel Spectral Clustering

- **Going further than ([Kammoun,Couillet'17])**, if $\underline{f'(\tau) = 0}$,

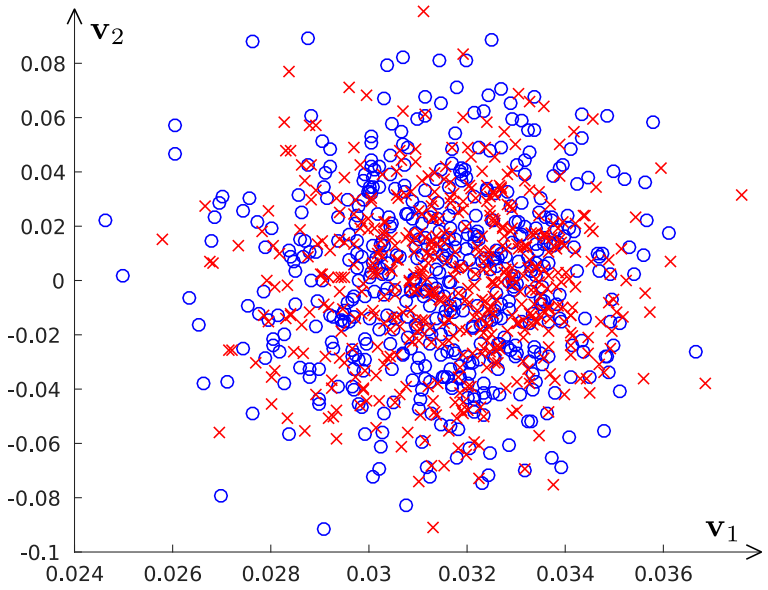
$$K \simeq \underbrace{f(\tau) 1_n 1_n^\mathsf{T}}_{O_{\|\cdot\|}(n)} + \cancel{f'(\tau) \frac{1}{p} Z Z^\mathsf{T}} + J A J^\mathsf{T}, \text{ avec } A = F\left(\begin{array}{c} f(\tau), f'(\tau), f''(\tau) \\ \cancel{\|\mu_a - \mu_b\|}, \mathrm{tr}\,(C_a - C_b), \dots \end{array} \right).$$

Improving Kernel Spectral Clustering

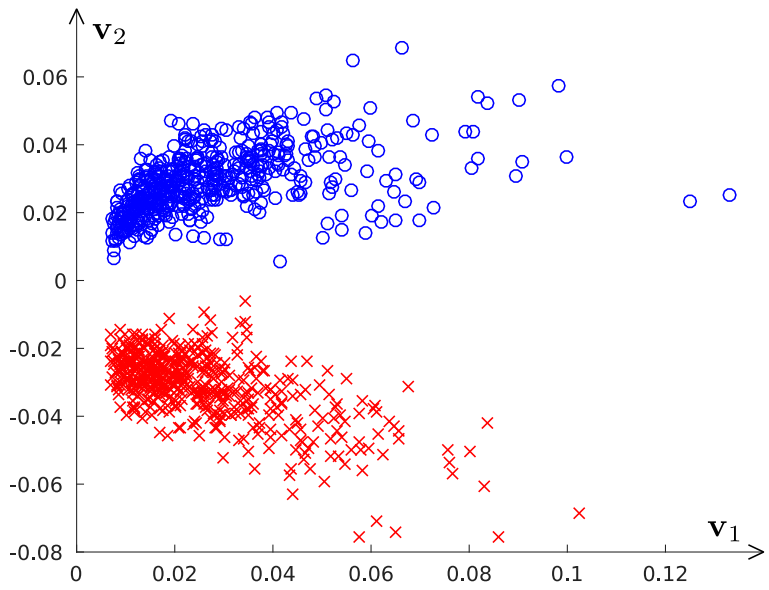
- **Going further than ([Kammoun,Couillet'17])**, if $\underline{f'(\tau) = 0}$,

$$K \simeq \underbrace{f(\tau) 1_n 1_n^\mathsf{T}}_{O_{\|\cdot\|}(n)} + \cancel{f'(\tau)\frac{1}{p} Z Z^\mathsf{T}} + J A J^\mathsf{T}, \text{ avec } A = F\left(\begin{array}{c} f(\tau), f'(\tau), f''(\tau) \\ \cancel{\|\mu_a - \mu_b\|}, \mathrm{tr}\,(C_a - C_b), \ldots \end{array} \right).$$

- **Gaussian case**: $\mathcal{N}(0, \mathbf{C}_1)$ vs. $\mathcal{N}(0, \mathbf{C}_2)$

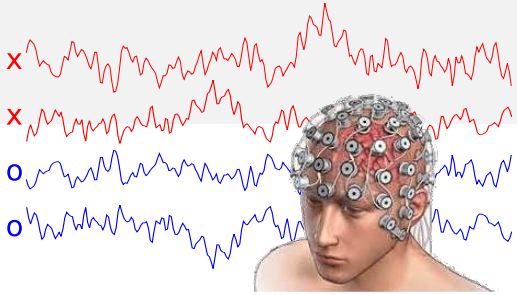


Kernel $K_{ij} = \exp(-\frac{1}{2p}\|x_i - x_j\|^2)$

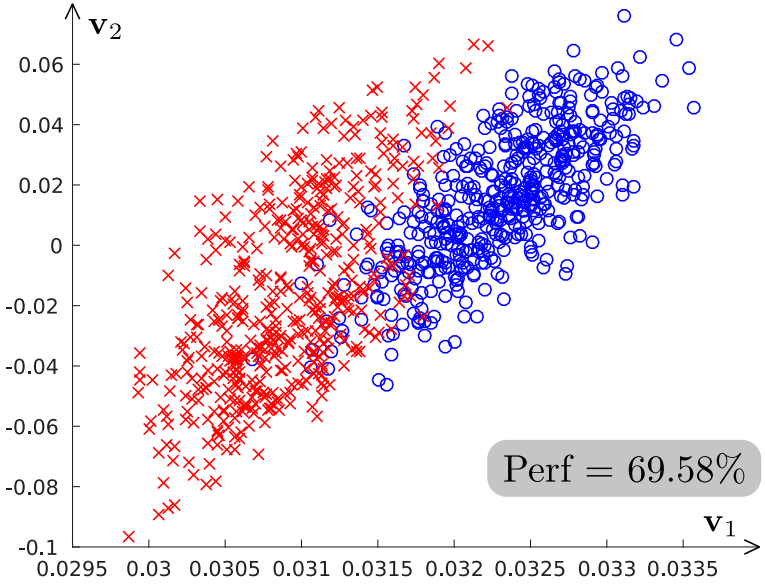


Kernel $K_{ij} = (\frac{1}{p}\|x_i - x_j\|^2 - \tau)^2$

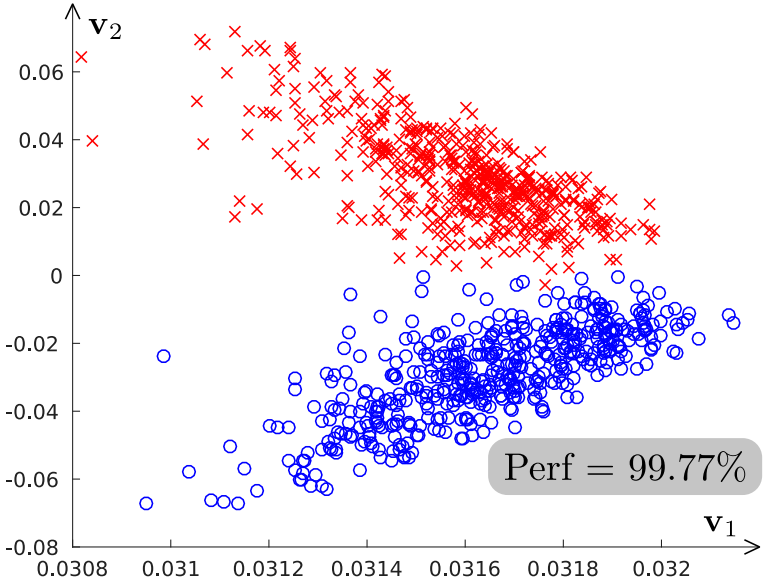
Improving Kernel Spectral Clustering



- **EEG data**: sane vs. epileptic patients

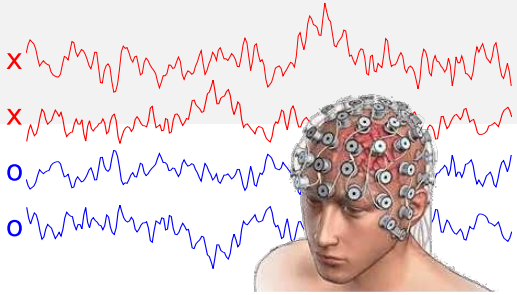


Kernel $K_{ij} = \exp(-\frac{1}{2p}\|x_i - x_j\|^2)$

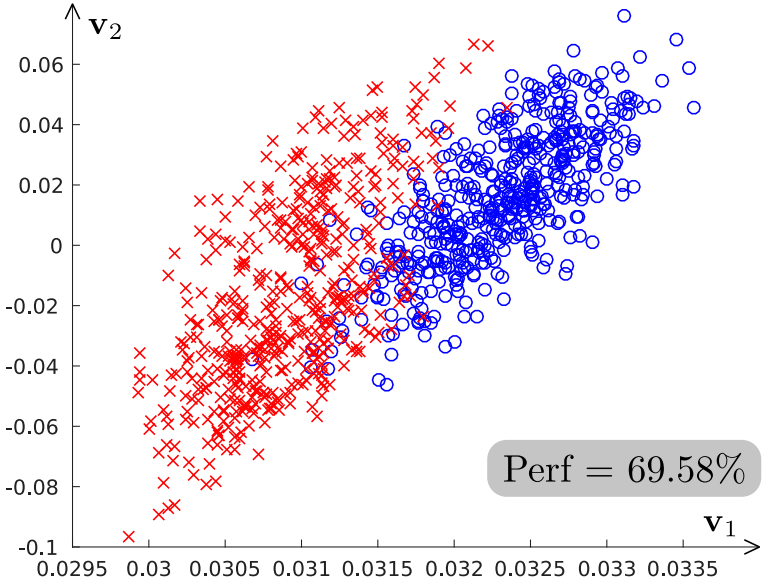


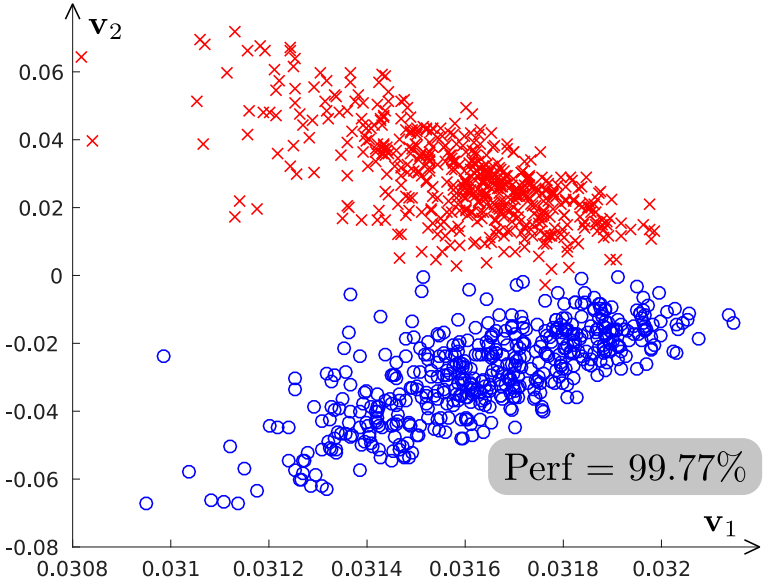
Kernel $K_{ij} = (\frac{1}{p}\|x_i - x_j\|^2 - \tau)^2$

Improving Kernel Spectral Clustering



- **EEG data**: sane vs. epileptic patients

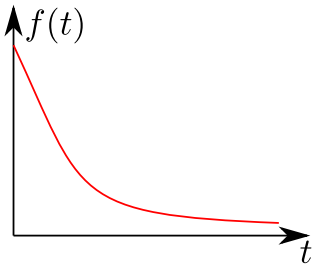


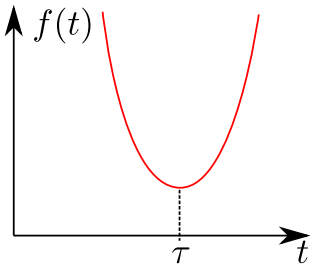


Kernel $K_{ij} = \exp(-\frac{1}{2p}\|x_i - x_j\|^2)$

Kernel $K_{ij} = (\frac{1}{p}\|x_i - x_j\|^2 - \tau)^2$

→ ***Remark***: *highly counter-intuitive kernel!*





Another, more striking, example: Semi-supervised Learning

Semi-supervised learning: a great idea that never worked!

Another, more striking, example: Semi-supervised Learning

Semi-supervised learning: a great idea that never worked!

- **Setting**: assume now
  - $x_1^{(a)}, \ldots, x_{n_{a,[l]}}^{(a)}$ already labelled (few),
  - $x_{n_{a,[l]}+1}^{(a)}, \ldots, x_{n_a}^{(a)}$ unlabelled (a lot).

Another, more striking, example: Semi-supervised Learning

Semi-supervised learning: a great idea that never worked!

- **Setting**: assume now
  - $x_1^{(a)}, \ldots, x_{n_{a,[l]}}^{(a)}$ already labelled (few),
  - $x_{n_{a,[l]}+1}^{(a)}, \ldots, x_{n_a}^{(a)}$ unlabelled (a lot).

- **Machine Learning original idea**: find "scores" F_{ia} for x_i to belong to class a

$$F = \mathrm{argmin}_{F \in \mathbb{R}^{n \times k}} \sum_{a=1}^{k} \sum_{i,j} K_{ij} \left(F_{ia} \quad - F_{ja} \quad \right)^2, \quad F_{ia}^{[l]} = \boldsymbol{\delta}_{\{x_i \in \mathcal{C}_a\}}.$$

Another, more striking, example: Semi-supervised Learning

Semi-supervised learning: a great idea that never worked!

- **Setting**: assume now
  - $x_1^{(a)}, \ldots, x_{n_{a,[l]}}^{(a)}$ already labelled (few),
  - $x_{n_{a,[l]}+1}^{(a)}, \ldots, x_{n_a}^{(a)}$ unlabelled (a lot).

- **Machine Learning original idea**: find "scores" F_{ia} for x_i to belong to class a

$$F = \mathrm{argmin}_{F \in \mathbb{R}^{n \times k}} \sum_{a=1}^{k} \sum_{i,j} K_{ij} \left(F_{ia} D_{ii}^{\alpha} - F_{ja} D_{jj}^{\alpha} \right)^2, \quad F_{ia}^{[l]} = \boldsymbol{\delta}_{\{x_i \in \mathcal{C}_a\}}.$$

Another, more striking, example: Semi-supervised Learning

Semi-supervised learning: a great idea that never worked!

- **Setting**: assume now
  - $x_1^{(a)}, \ldots, x_{n_{a,[l]}}^{(a)}$ already labelled (few),
  - $x_{n_{a,[l]}+1}^{(a)}, \ldots, x_{n_a}^{(a)}$ unlabelled (a lot).

- **Machine Learning original idea**: find "scores" F_{ia} for x_i to belong to class a

$$F = \operatorname{argmin}_{F \in \mathbb{R}^{n \times k}} \sum_{a=1}^{k} \sum_{i,j} K_{ij} \left(F_{ia} D_{ii}^{\alpha} - F_{ja} D_{jj}^{\alpha} \right)^2, \quad F_{ia}^{[l]} = \boldsymbol{\delta}_{\{x_i \in \mathcal{C}_a\}}.$$

- **Explicit solution**:

$$F^{[u]} = \left(I_{n_{[u]}} - D_{[u]}^{-1-\alpha} K_{[uu]} D^{\alpha}{}_{[u]} \right)^{-1} D_{[u]}^{-1-\alpha} K_{[ul]} D^{\alpha}{}_{[l]} F^{[l]}$$

where $D = \operatorname{diag}(K 1_n)$ (degree matrix) and $[ul]$, $[uu]$, ... blocks of **l**abeled/**u**nlabeled data.

The finite-dimensional case: What we expect

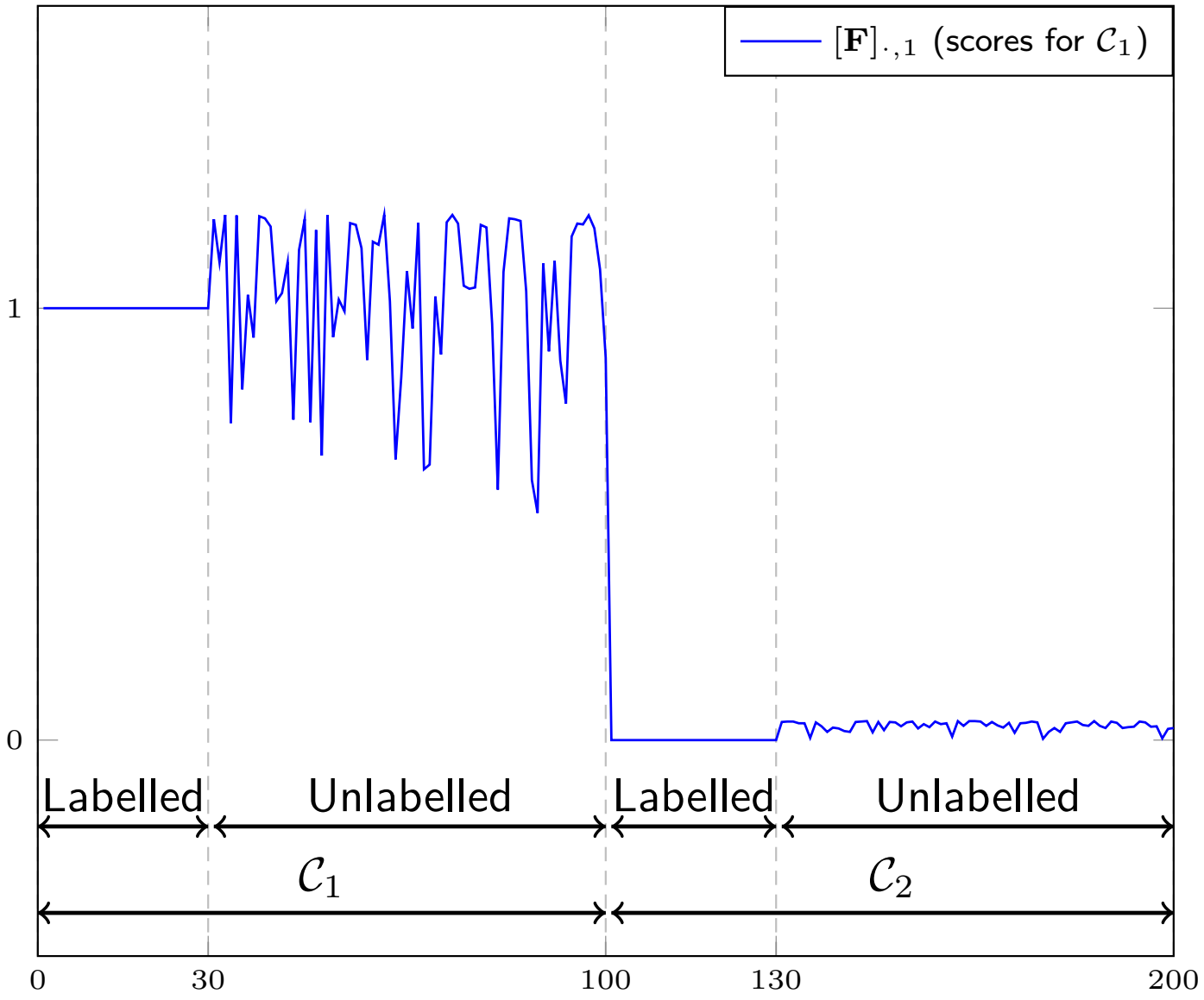


Figure: Outcome $\mathbf{F}$ of Laplacian algorithms ($\alpha = -1$) for $\mathcal{N}(\pm\mu, I_p)$ with $p = 1$.

The finite-dimensional case: What we expect

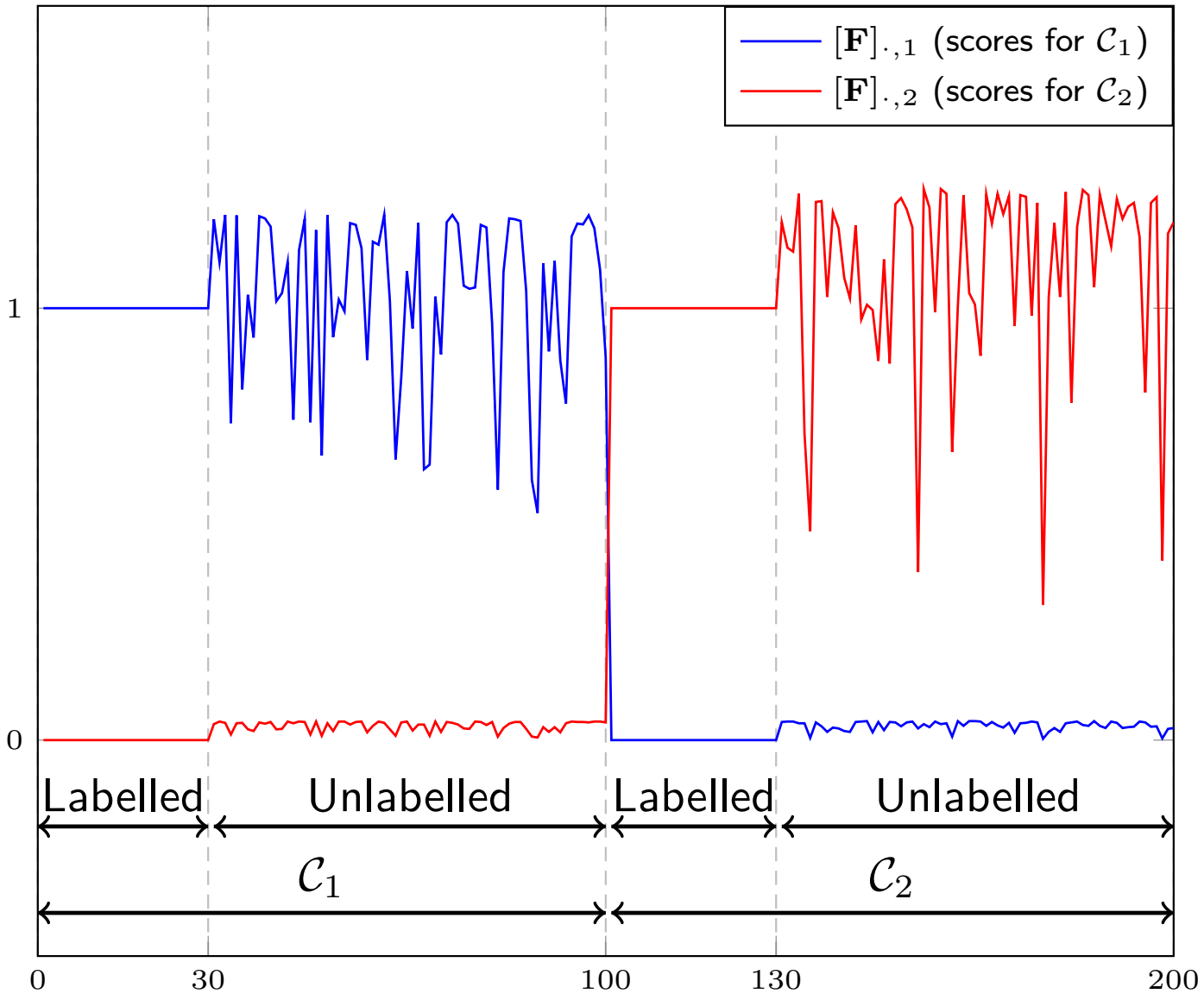


Figure: Outcome $\mathbf{F}$ of Laplacian algorithms ($\alpha = -1$) for $\mathcal{N}(\pm\mu, I_p)$ with $p = 1$.

The reality: What we see!

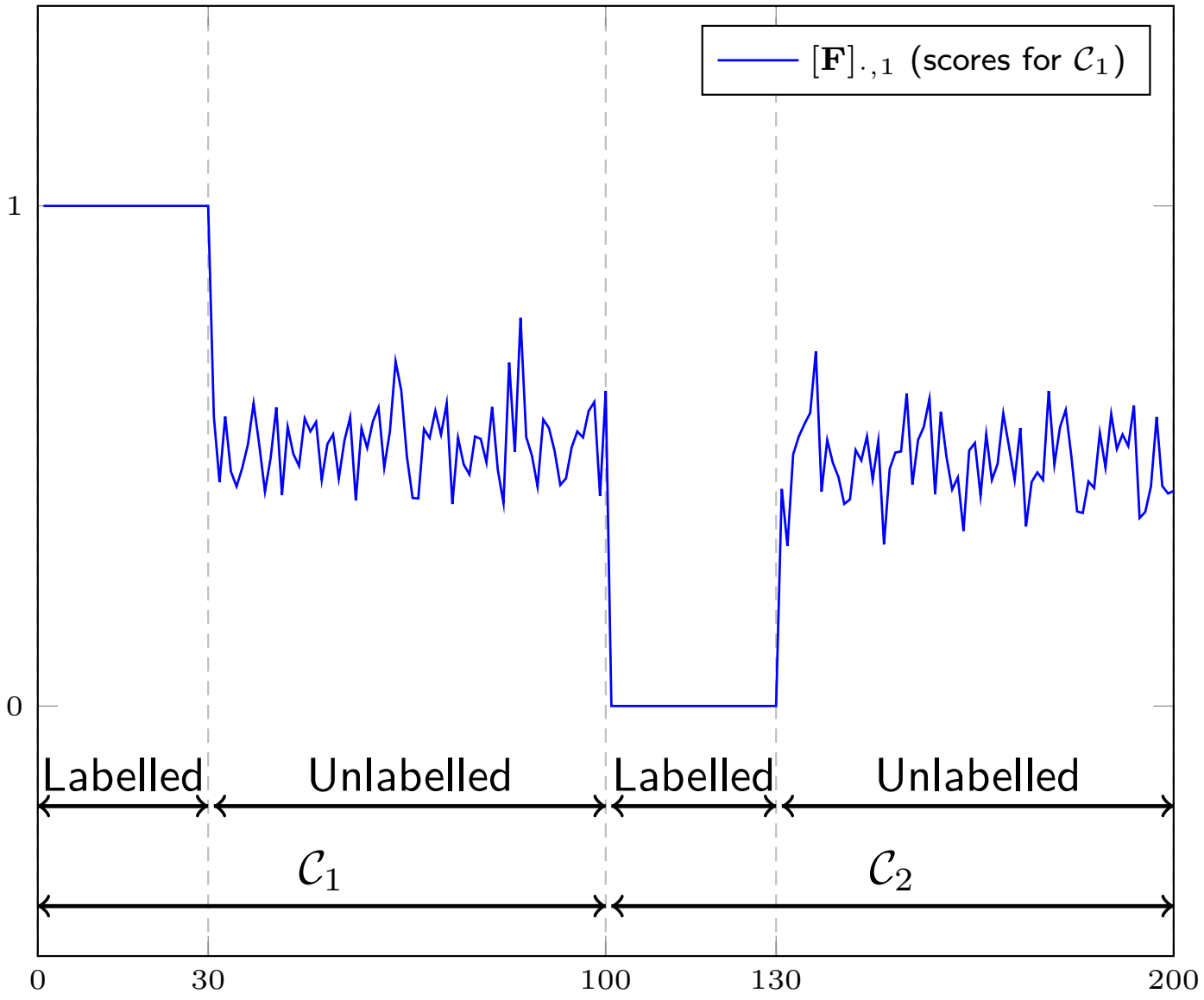


Figure: Outcome $\mathbf{F}$ of Laplacian algorithms ($\alpha = -1$) for $\mathcal{N}(\pm\mu, I_p)$ with $p = 80$.

The reality: What we see!

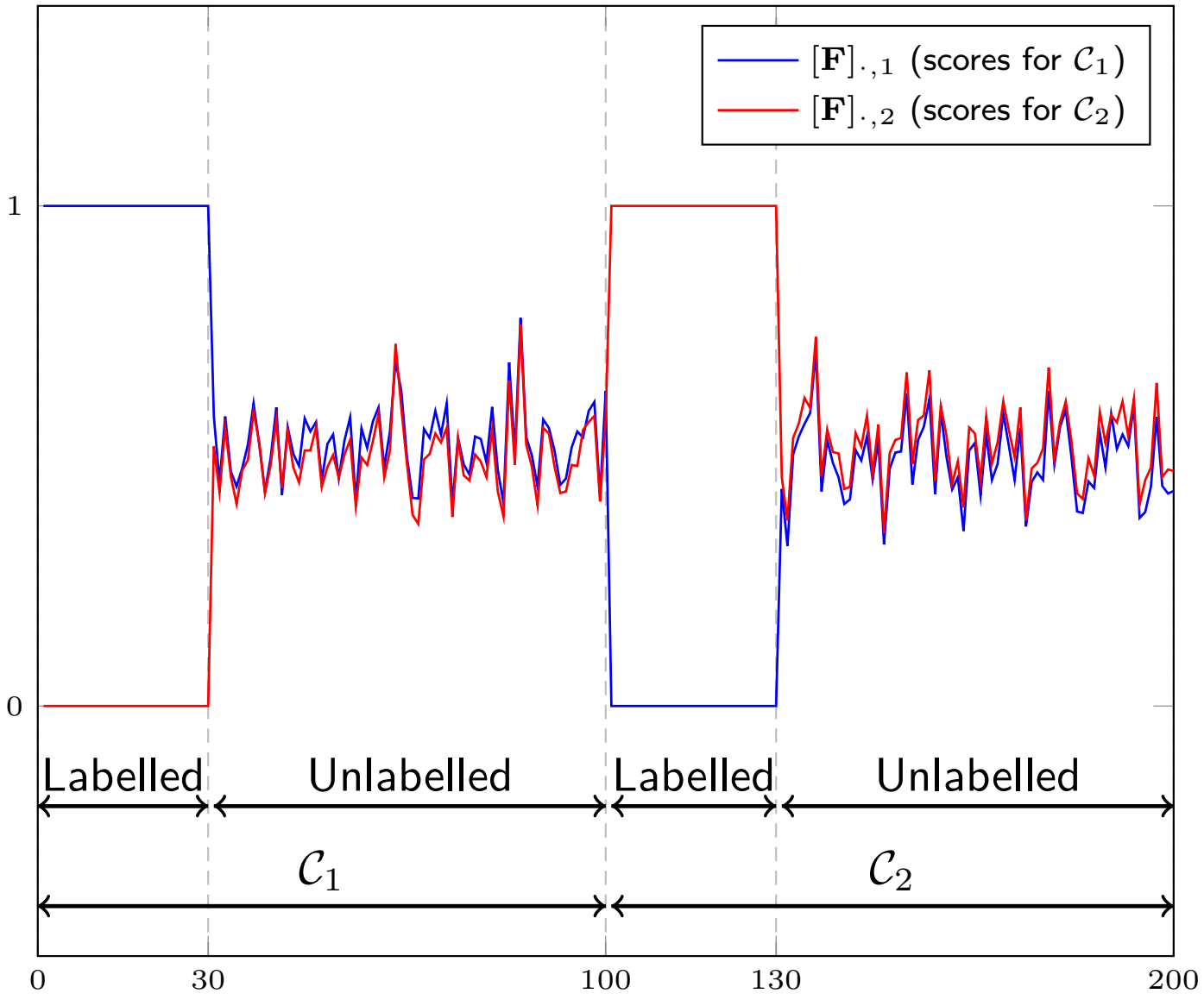
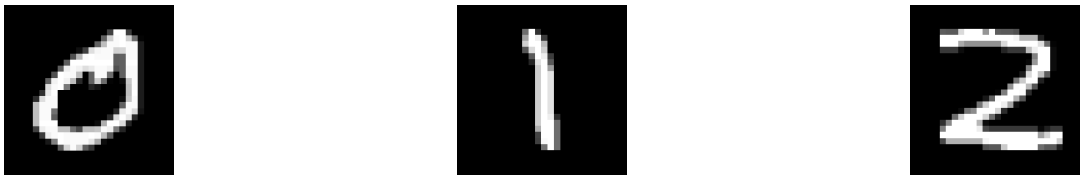


Figure: Outcome $\mathbf{F}$ of Laplacian algorithms ($\alpha = -1$) for $\mathcal{N}(\pm\mu, I_p)$ with $p = 80$.

The reality: What we see! (on MNIST)



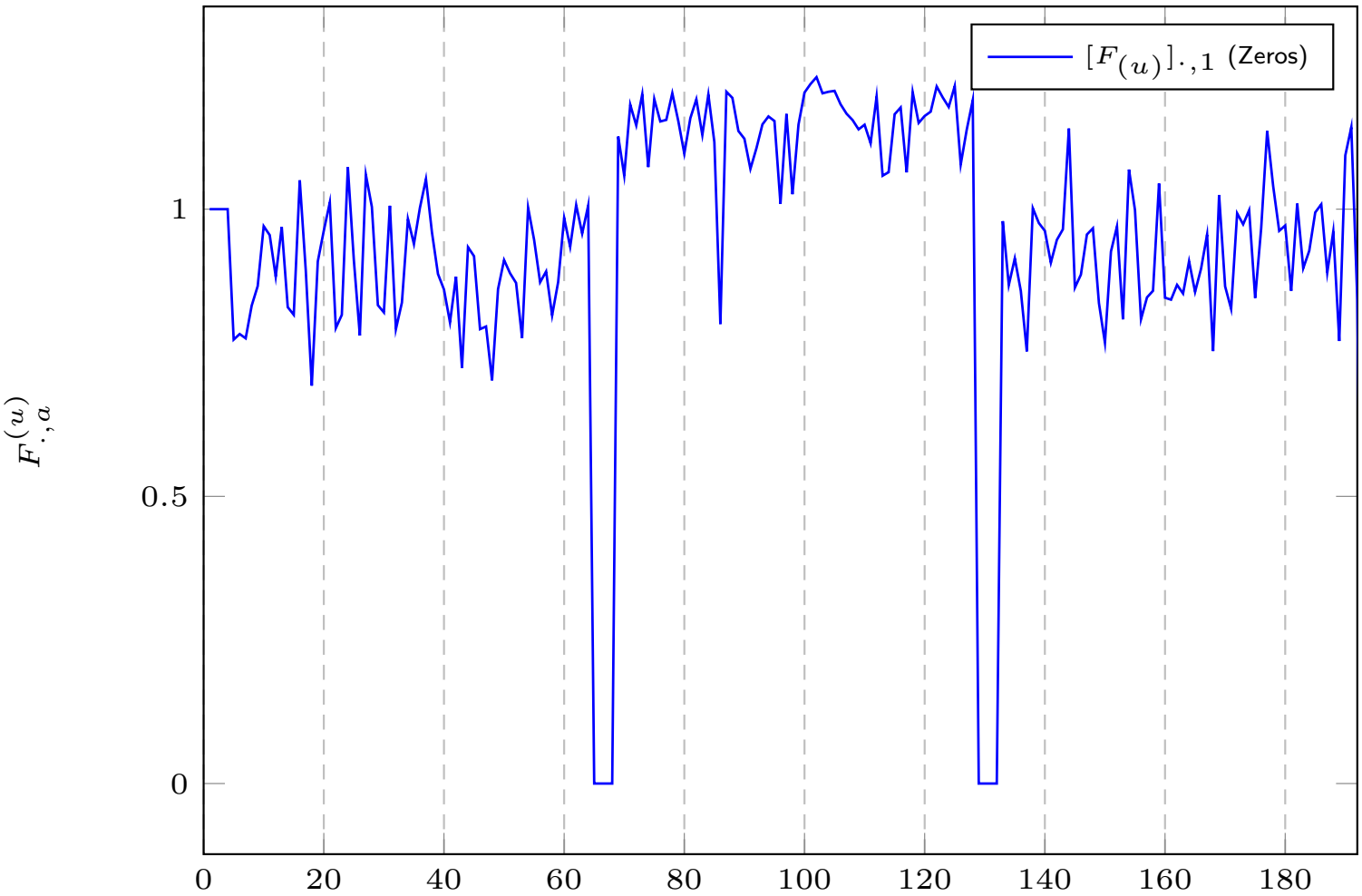


Figure: Vectors $[F^{(u)}]_{\cdot,a}$, $a = 1, 2, 3$, for 3-class MNIST data (zeros, ones, twos), $n = 192$, $p = 784$, $n_l/n = 1/16$, Gaussian kernel.

The reality: What we see! (on MNIST)

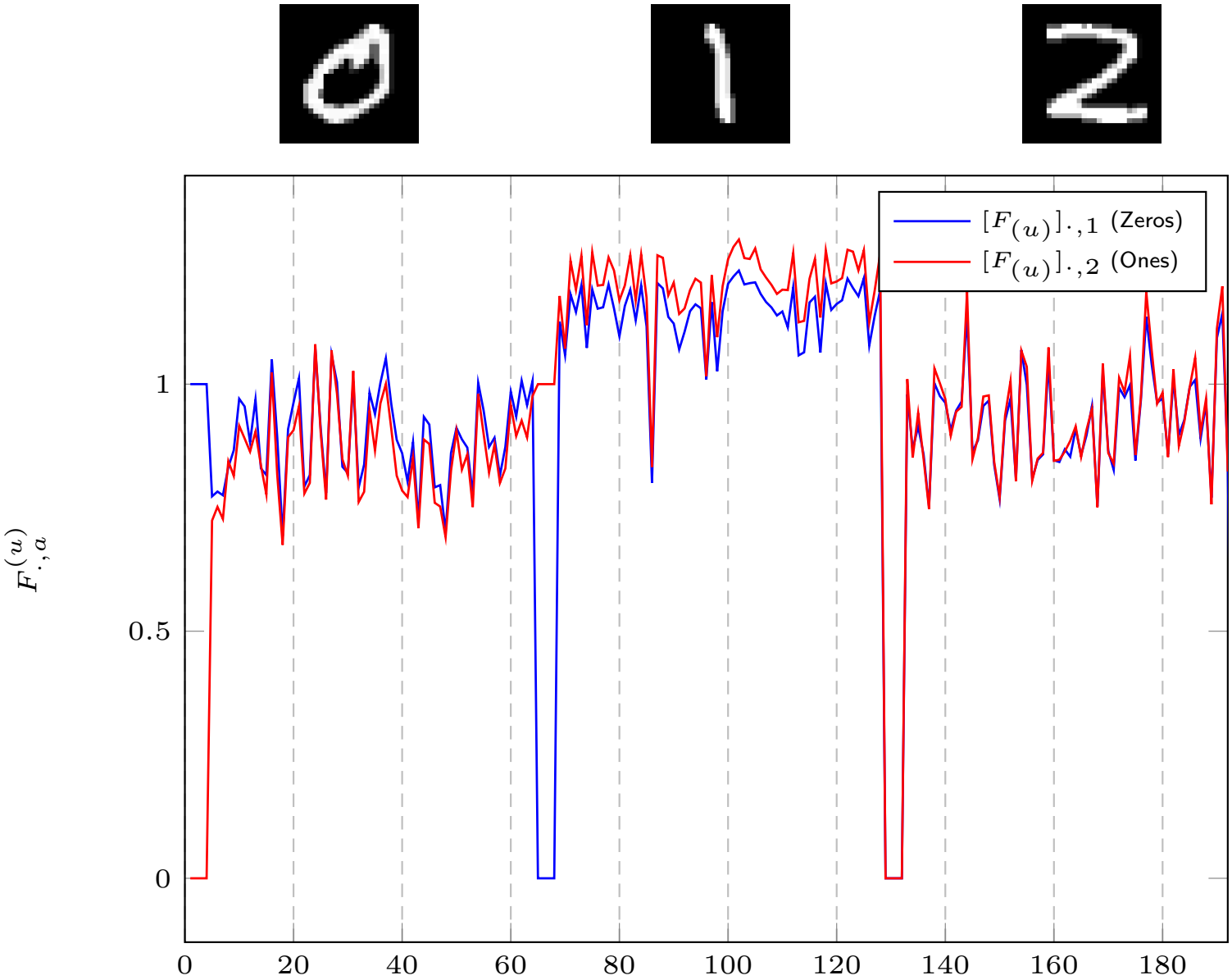


Figure: Vectors $[F^{(u)}]_{\cdot,a}$, $a = 1, 2, 3$, for 3-class MNIST data (zeros, ones, twos), $n = 192$, $p = 784$, $n_l/n = 1/16$, Gaussian kernel.

The reality: What we see! (on MNIST)

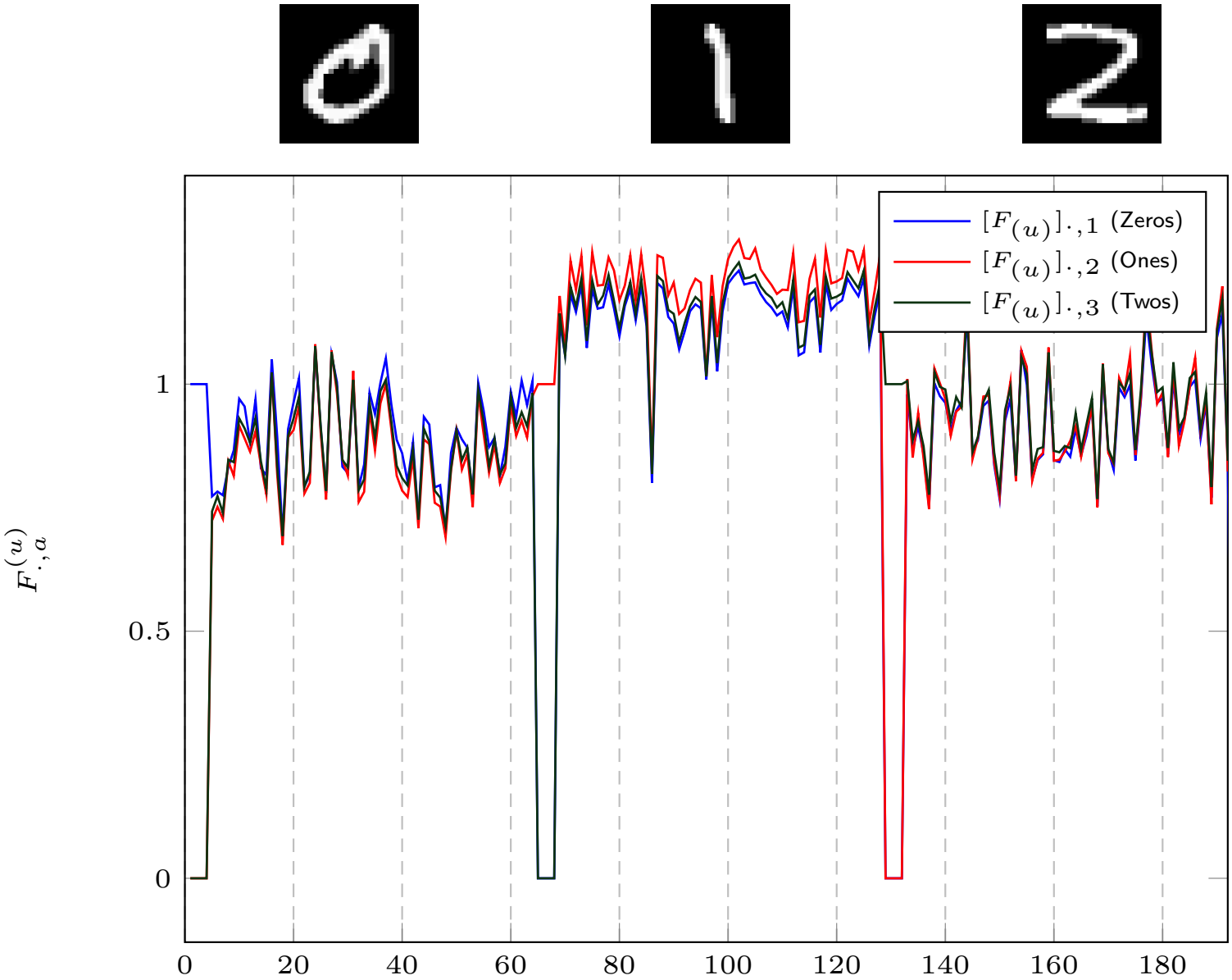


Figure: Vectors $[F^{(u)}]_{\cdot,a}$, $a = 1, 2, 3$, for 3-class MNIST data (zeros, ones, twos), $n = 192$, $p = 784$, $n_l/n = 1/16$, Gaussian kernel.

Consequences of the finite-dimensional "mismatch"

Exploiting RMT to resurrect SSL

Consequences of the finite-dimensional "mismatch"

- A priori, **the algorithm should not work**

Exploiting RMT to resurrect SSL

Consequences of the finite-dimensional "mismatch"

- A priori, **the algorithm should not work**
- Indeed "in general" it does not!

Exploiting RMT to resurrect SSL

Consequences of the finite-dimensional "mismatch"

- A priori, **the algorithm should not work**
- Indeed "in general" it does not!
- But, luckily, after some (not clearly motivated) renormalization (e.g., $\alpha = -1$, $F_{i\cdot} \leftarrow F_{i\cdot}/n_{[l],i}$), it works again...

Exploiting RMT to resurrect SSL

Consequences of the finite-dimensional "mismatch"

- A priori, **the algorithm should not work**
- Indeed "in general" it does not!
- But, luckily, after some (not clearly motivated) renormalization (e.g., $\alpha = -1$, $F_{i\cdot} \leftarrow F_{i\cdot}/n_{[l],i}$), it works again...

- **BUT** it does not use efficiently unlabelled data!

Exploiting RMT to resurrect SSL

Consequences of the finite-dimensional "mismatch"

- A priori, **the algorithm should not work**
- Indeed "in general" it does not!
- But, luckily, after some (not clearly motivated) renormalization (e.g., $\alpha = -1$, $F_{i\cdot} \leftarrow F_{i\cdot}/n_{[l],i}$), it works again...

- **BUT** it does not use efficiently unlabelled data!

Chapelle, Schölkopf, Zien, "**Semi-Supervised Learning**", Chapter 4, 2009.

Our concern is this: it is frequently the case that we would be better off just discarding the unlabeled data and employing a supervised method, rather than taking a semi-supervised route. Thus ***we worry about the embarrassing situation where the addition of unlabeled data degrades the performance*** *of a classifier.*

Asymptotic Performance Analysis

Theorem ([Mai,C'18] Asymptotic Performance of SSL)

For $x_i \in \mathcal{C}_b$ unlabelled, score vector $F_{i,\cdot} \in \mathbb{R}^k$ satisfies:

$$F_{i,\cdot} - G_b \to 0, \; G_b \sim \mathcal{N}(m_b, \Sigma_b)$$

with $m_b \in \mathbb{R}^k$, $\Sigma_b \in \mathbb{R}^{k \times k}$ function of

- $f(\tau), f'(\tau), f''(\tau)$, $\mu_1, \ldots, \mu_k$, $C_1, \ldots, C_k$
- *only n_l.*

Asymptotic Performance Analysis

Theorem ([Mai,C'18] Asymptotic Performance of SSL)

For $x_i \in \mathcal{C}_b$ unlabelled, score vector $F_{i,\cdot} \in \mathbb{R}^k$ satisfies:

$$F_{i,\cdot} - G_b \to 0, \; G_b \sim \mathcal{N}(m_b, \Sigma_b)$$

with $m_b \in \mathbb{R}^k$, $\Sigma_b \in \mathbb{R}^{k \times k}$ function of

- $f(\tau), f'(\tau), f''(\tau)$, $\mu_1, \ldots, \mu_k$, $C_1, \ldots, C_k$
- *only n_l.*

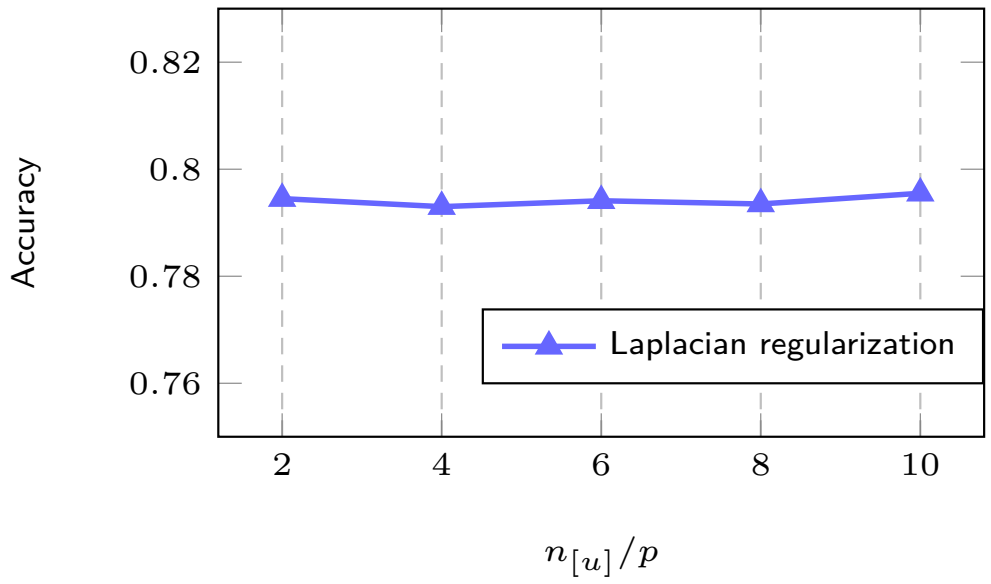


Figure: Accuracy as a function of $n_{[u]}/p$ with $n_{[l]}/p = 2$, $c_1 = c_2$, $p = 100$, $-\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = [1; \mathbf{0}_{p-1}]$, $\{\mathbf{C}\}_{i,j} = .1^{|i-j|}$. Graph constructed with $K_{ij} = e^{-\|x_i - x_j\|^2/p}$.

Asymptotic Performance Analysis

Theorem ([Mai,C'18] Asymptotic Performance of SSL)

For $x_i \in \mathcal{C}_b$ unlabelled, score vector $F_{i,\cdot} \in \mathbb{R}^k$ satisfies:

$$F_{i,\cdot} - G_b \to 0, \; G_b \sim \mathcal{N}(m_b, \Sigma_b)$$

with $m_b \in \mathbb{R}^k$, $\Sigma_b \in \mathbb{R}^{k \times k}$ function of

- $f(\tau), f'(\tau), f''(\tau)$, $\mu_1, \ldots, \mu_k$, $C_1, \ldots, C_k$
- *only n_l.*

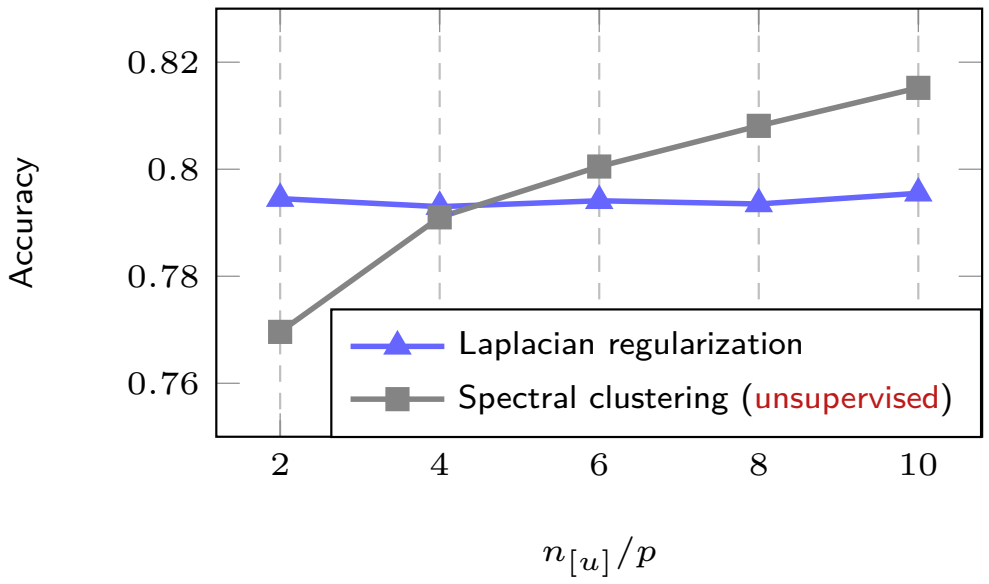


Figure: Accuracy as a function of $n_{[u]}/p$ with $n_{[l]}/p = 2$, $c_1 = c_2$, $p = 100$, $-\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = [1; \mathbf{0}_{p-1}]$, $\{\mathbf{C}\}_{i,j} = .1^{|i-j|}$. Graph constructed with $K_{ij} = e^{-\|x_i - x_j\|^2/p}$.

Improved SSL

Solution: From RMT calculus (but not from ML intuition!), solution is to replace K by

$$\tilde{K} \equiv PKP, \quad P = I_n - \frac{1}{n} 1_n 1_n^\mathsf{T}.$$

Improved SSL

Solution: From RMT calculus (but not from ML intuition!), solution is to replace K by

$$\tilde{K} \equiv PKP, \quad P = I_n - \frac{1}{n} 1_n 1_n^\mathsf{T}.$$

Theorem ([Mai,C'19] Asymptotic Performance of **Improved** SSL)

For $x_i \in \mathcal{C}_b$ unlabelled, score vector $\tilde{F}_{i,\cdot} \in \mathbb{R}^k$ satisfies:

$$\tilde{F}_{i,\cdot} - \tilde{G}_b \to 0, \ \tilde{G}_b \sim \mathcal{N}(\tilde{m}_b, \tilde{\Sigma}_b)$$

with $\tilde{m}_b \in \mathbb{R}^k$, $\tilde{\Sigma}_b \in \mathbb{R}^{k \times k}$ function of

- $f(\tau), f'(\tau), f''(\tau)$, $\mu_1, \ldots, \mu_k$, $C_1, \ldots, C_k$
- n_l **and** n_u.

Improved SSL

Solution: From RMT calculus (but not from ML intuition!), solution is to replace K by

$$\tilde{K} \equiv PKP, \quad P = I_n - \frac{1}{n} 1_n 1_n^\mathsf{T}.$$

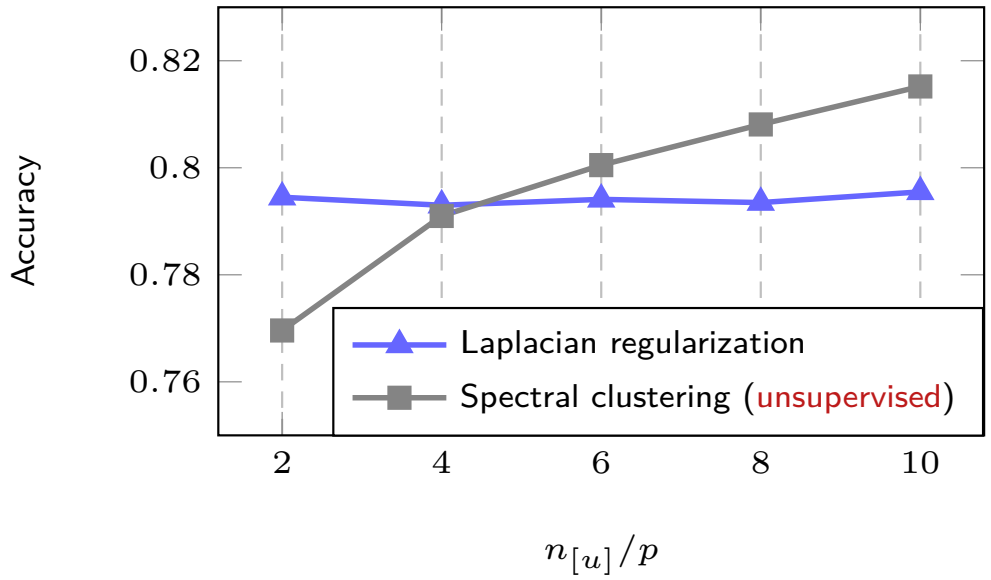
Theorem ([Mai,C'19] Asymptotic Performance of Improved SSL)

For $x_i \in \mathcal{C}_b$ unlabelled, score vector $\tilde{F}_{i,\cdot} \in \mathbb{R}^k$ satisfies:

$$\tilde{F}_{i,\cdot} - \tilde{G}_b \to 0, \; \tilde{G}_b \sim \mathcal{N}(\tilde{m}_b, \tilde{\Sigma}_b)$$

with $\tilde{m}_b \in \mathbb{R}^k$, $\tilde{\Sigma}_b \in \mathbb{R}^{k \times k}$ function of

- $f(\tau), f'(\tau), f''(\tau)$, $\mu_1, \ldots, \mu_k$, $C_1, \ldots, C_k$
- n_l **and** n_u.



Improved SSL

Solution: From RMT calculus (but not from ML intuition!), solution is to replace K by

$$\tilde{K} \equiv PKP, \quad P = I_n - \frac{1}{n} 1_n 1_n^\mathsf{T}.$$

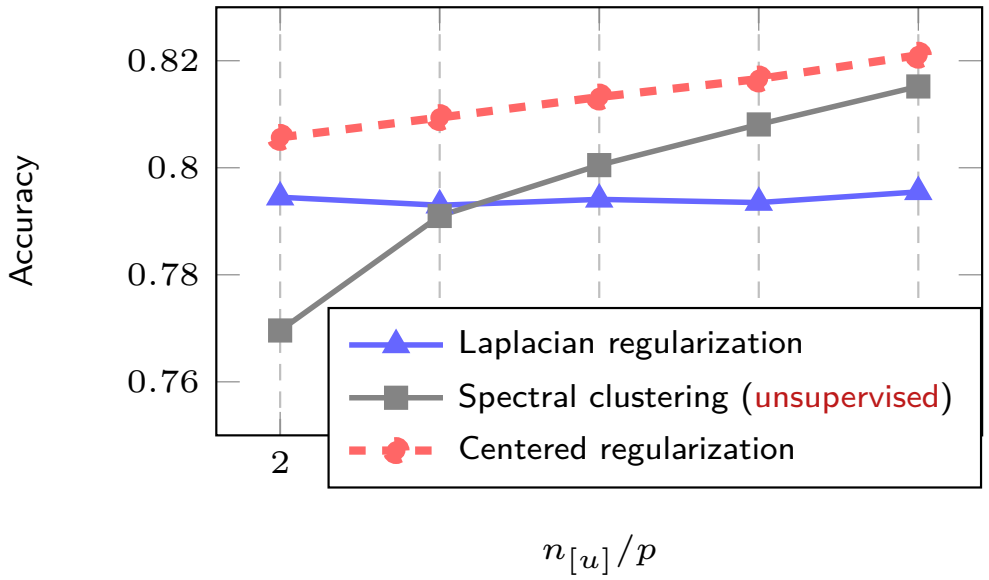
Theorem ([Mai,C'19] Asymptotic Performance of Improved SSL)

For $x_i \in \mathcal{C}_b$ unlabelled, score vector $\tilde{F}_{i,\cdot} \in \mathbb{R}^k$ satisfies:

$$\tilde{F}_{i,\cdot} - \tilde{G}_b \to 0, \; \tilde{G}_b \sim \mathcal{N}(\tilde{m}_b, \tilde{\Sigma}_b)$$

with $\tilde{m}_b \in \mathbb{R}^k$, $\tilde{\Sigma}_b \in \mathbb{R}^{k \times k}$ function of

- $f(\tau), f'(\tau), f''(\tau)$, $\mu_1, \ldots, \mu_k$, $C_1, \ldots, C_k$
- n_l **and** n_u.



What about real data?

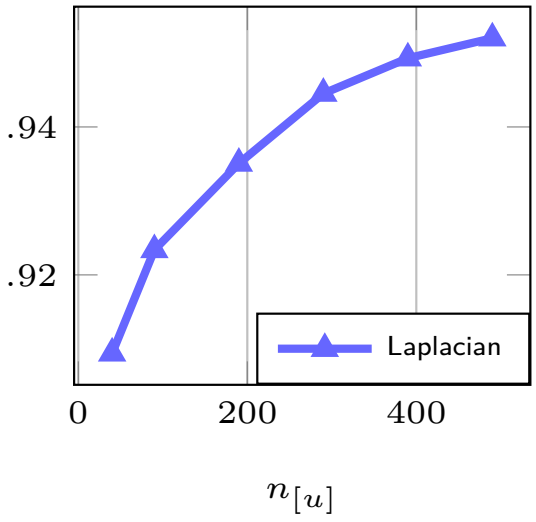


Figure: **Top**: distribution of normalized pairwise distances for noisy MNIST data (8,9). **Bottom**: average accuracy as a function of $n_{[u]}$ with $n_{[l]} = 10$, computed over 1000 random realizations.

What about real data?

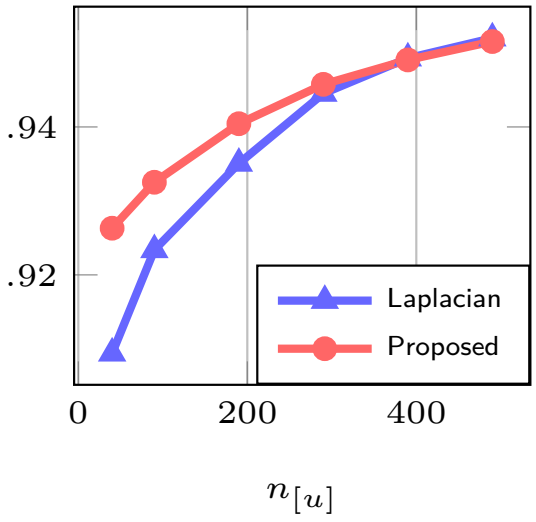


Figure: **Top**: distribution of normalized pairwise distances for noisy MNIST data (8,9). **Bottom**: average accuracy as a function of $n_{[u]}$ with $n_{[l]} = 10$, computed over 1000 random realizations.

What about real data?

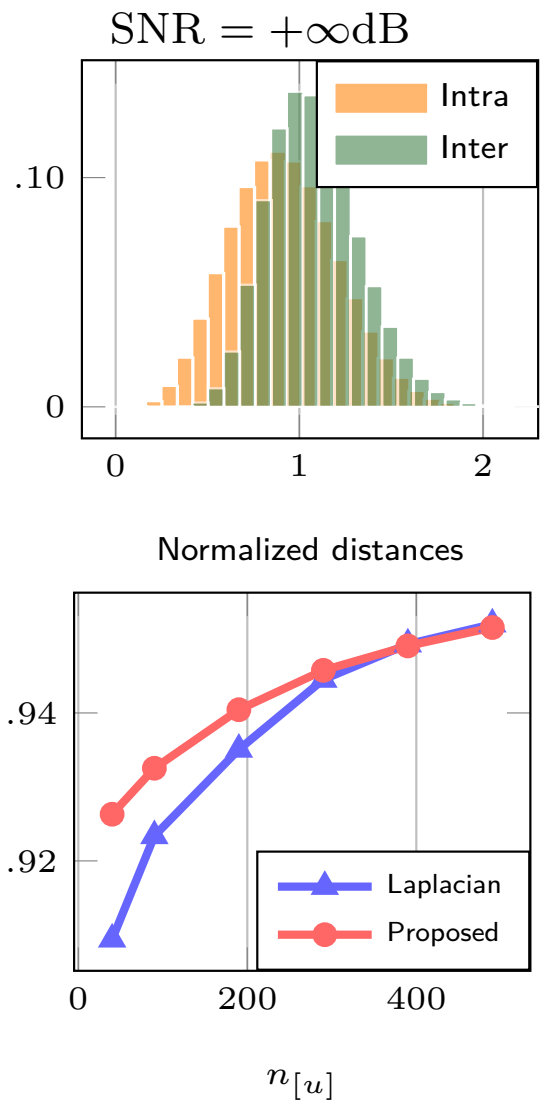


Figure: **Top**: distribution of normalized pairwise distances for noisy MNIST data (8,9). **Bottom**: average accuracy as a function of $n_{[u]}$ with $n_{[l]} = 10$, computed over 1000 random realizations.

What about real data?

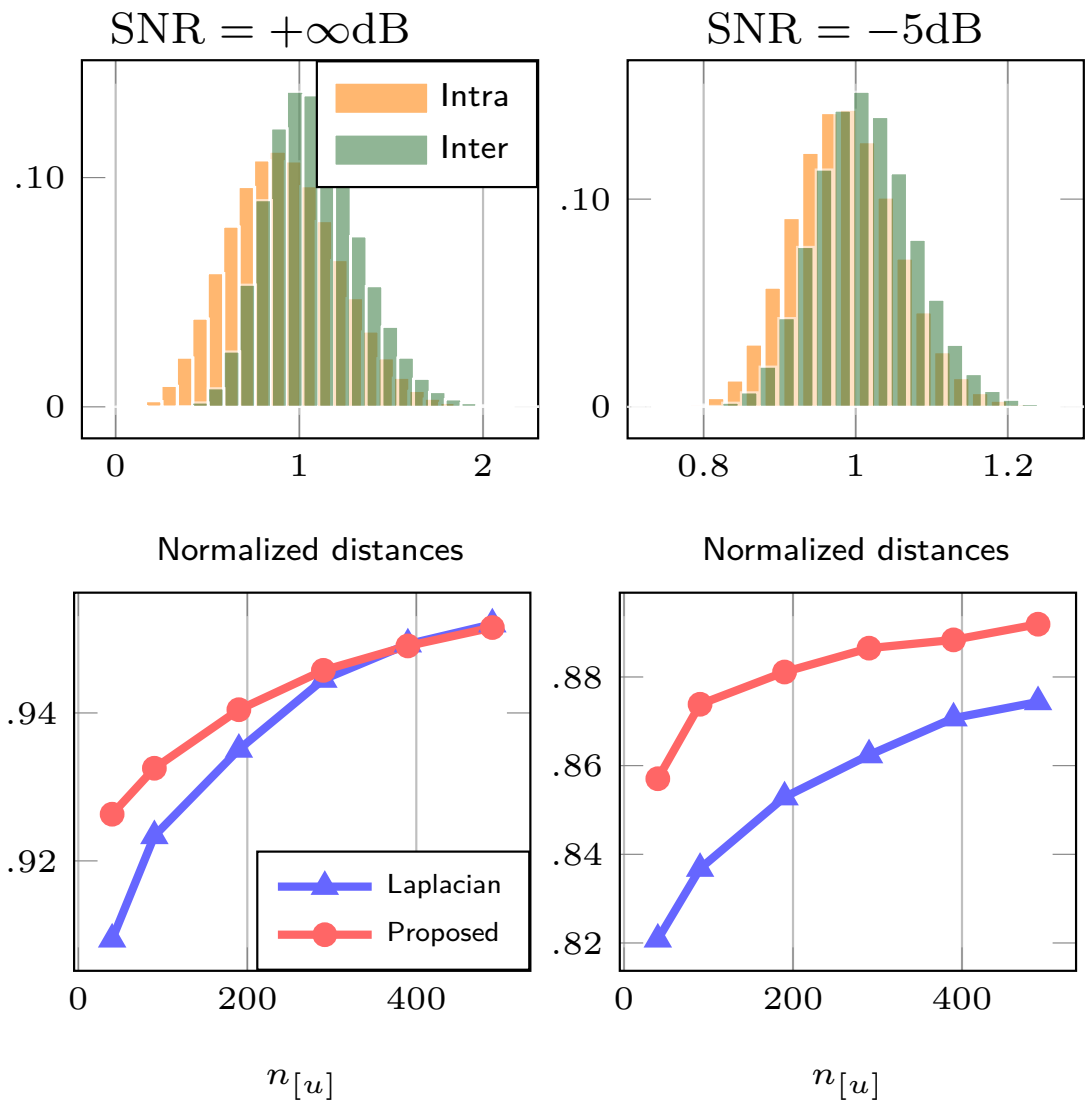


Figure: **Top**: distribution of normalized pairwise distances for noisy MNIST data (8,9). **Bottom**: average accuracy as a function of $n_{[u]}$ with $n_{[l]} = 10$, computed over 1000 random realizations.

What about real data?

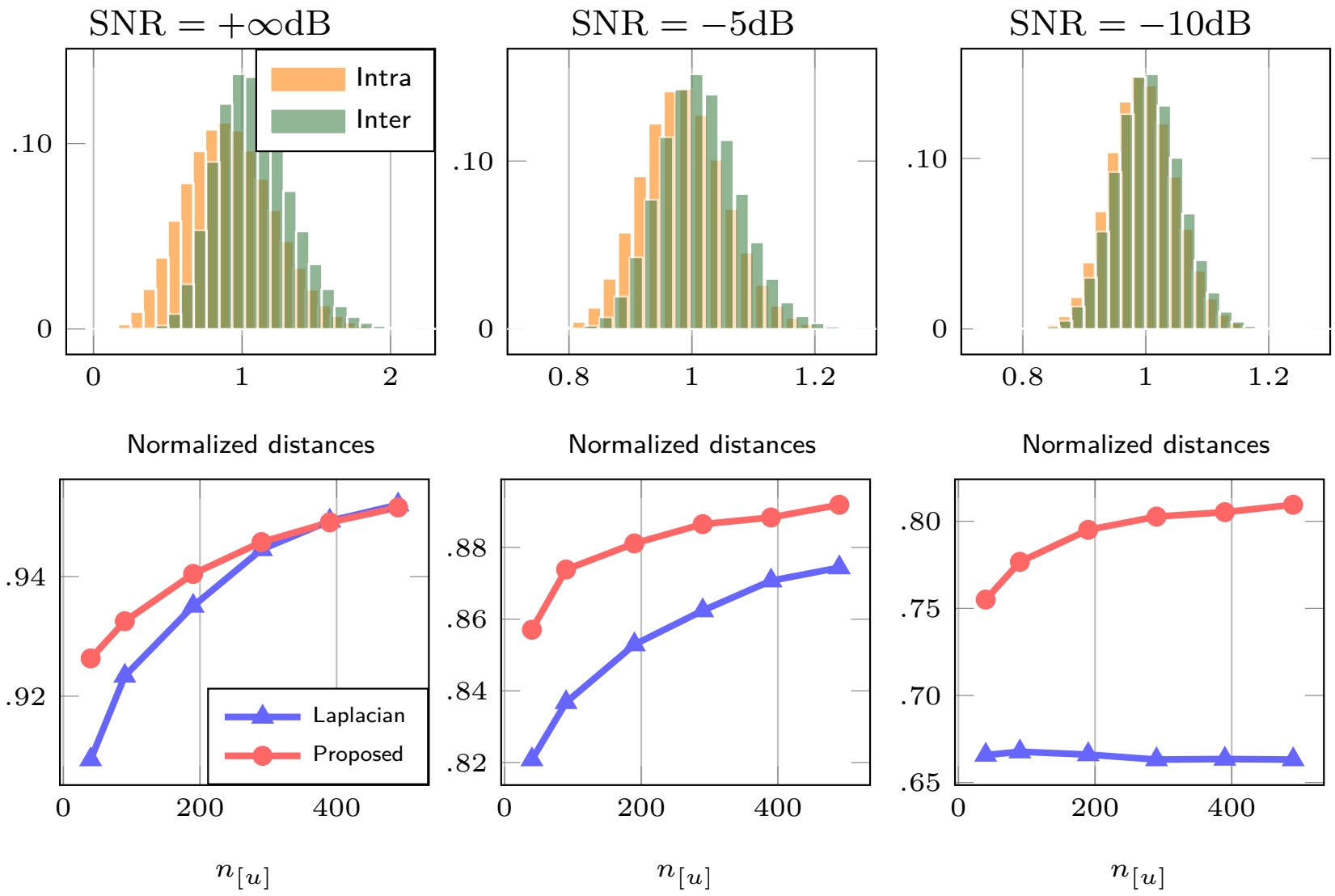
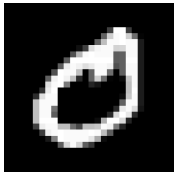
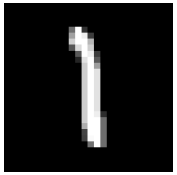


Figure: **Top**: distribution of normalized pairwise distances for noisy MNIST data (8,9). **Bottom**: average accuracy as a function of $n_{[u]}$ with $n_{[l]} = 10$, computed over 1000 random realizations.

Experimental evidence: MNIST


Digits	(0,8)	(2,7)	(6,9)
$n_u = 100$			
Centered kernel **(RMT)**	**89.5±3.6**	**89.5±3.4**	**85.3±5.9**
Iterated centered kernel **(RMT)**	**89.5±3.6**	**89.5±3.4**	**85.3±5.9**
Laplacian	75.5±5.6	74.2±5.8	70.0±5.5
Iterated Laplacian	87.2±4.7	86.0±5.2	81.4±6.8
Manifold	88.0±4.7	88.4±3.9	82.8±6.5
$n_u = 1000$			
Centered kernel **(RMT)**	92.2±0.9	92.5±0.8	92.6±1.6
Iterated centered kernel **(RMT)**	**92.3±0.9**	**92.5± 0.8**	**92.9±1.4**
Laplacian	65.6±4.1	74.4±4.0	69.5±3.7
Iterated Laplacian	**92.2±0.9**	92.4±0.9	92.0±1.6
Manifold	91.1±1.7	91.4±1.9	91.4±2.0

Table: Comparison of classification accuracy (%) on MNIST datasets with $n_l = 10$. Computed over 1000 random iterations for $n_u = 100$ and 100 for $n_u = 1000$.

Experimental evidence: Traffic signs (HOG features)

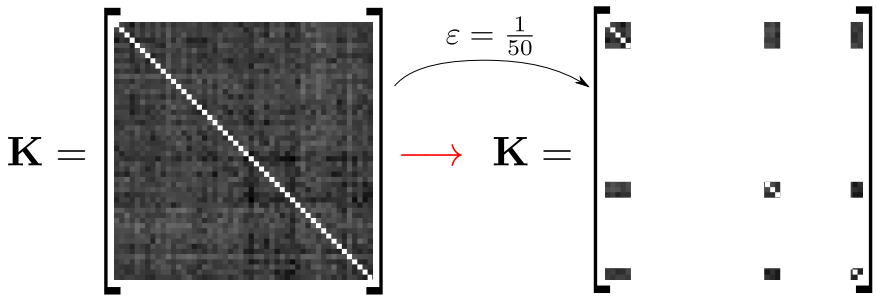
Class ID	(2,7)	(9,10)	(11,18)
$n_u = 100$			
Centered kernel **(RMT)**	79.0±10.4	77.5±9.2	78.5±7.1
Iterated centered kernel **(RMT)**	**85.3±5.9**	**89.2±5.6**	**90.1±6.7**
Laplacian	73.8±9.8	77.3±9.5	78.6±7.2
Iterated Laplacian	83.7±7.2	88.0±6.8	87.1±8.8
Manifold	77.6±8.9	81.4±10.4	82.3±10.8
$n_u = 1000$			
Centered kernel **(RMT)**	83.6±2.4	84.6±2.4	88.7±9.4
Iterated centered kernel **(RMT)**	**84.8±3.8**	**88.0±5.5**	**96.4±3.0**
Laplacian	72.7±4.2	88.9±5.7	95.8±3.2
Iterated Laplacian	83.0±5.5	88.2±6.0	92.7±6.1
Manifold	77.7±5.8	85.0±9.0	90.6±8.1

Table: Comparison of classification accuracy (%) on German Traffic Sign datasets with $n_l = 10$. Computed over 1000 random iterations for $n_u = 100$ and 100 for $n_u = 1000$.

Even more striking: new intuitions and cheap algorithms

- **Computation cost reduction**: $(p, n \gg 1)$

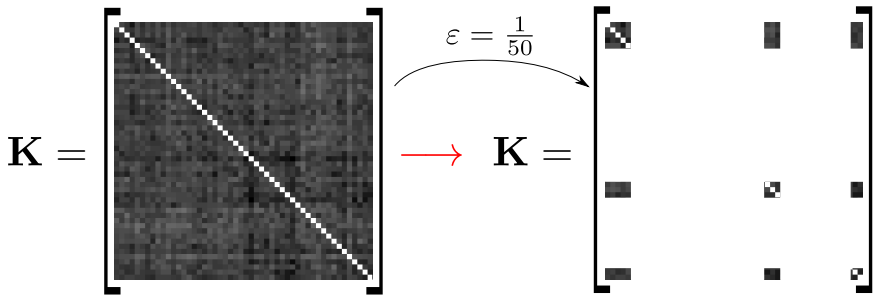
  $\rightarrow$ ε-subsampling $K \in \mathbb{R}^{n\varepsilon \times n\varepsilon}$



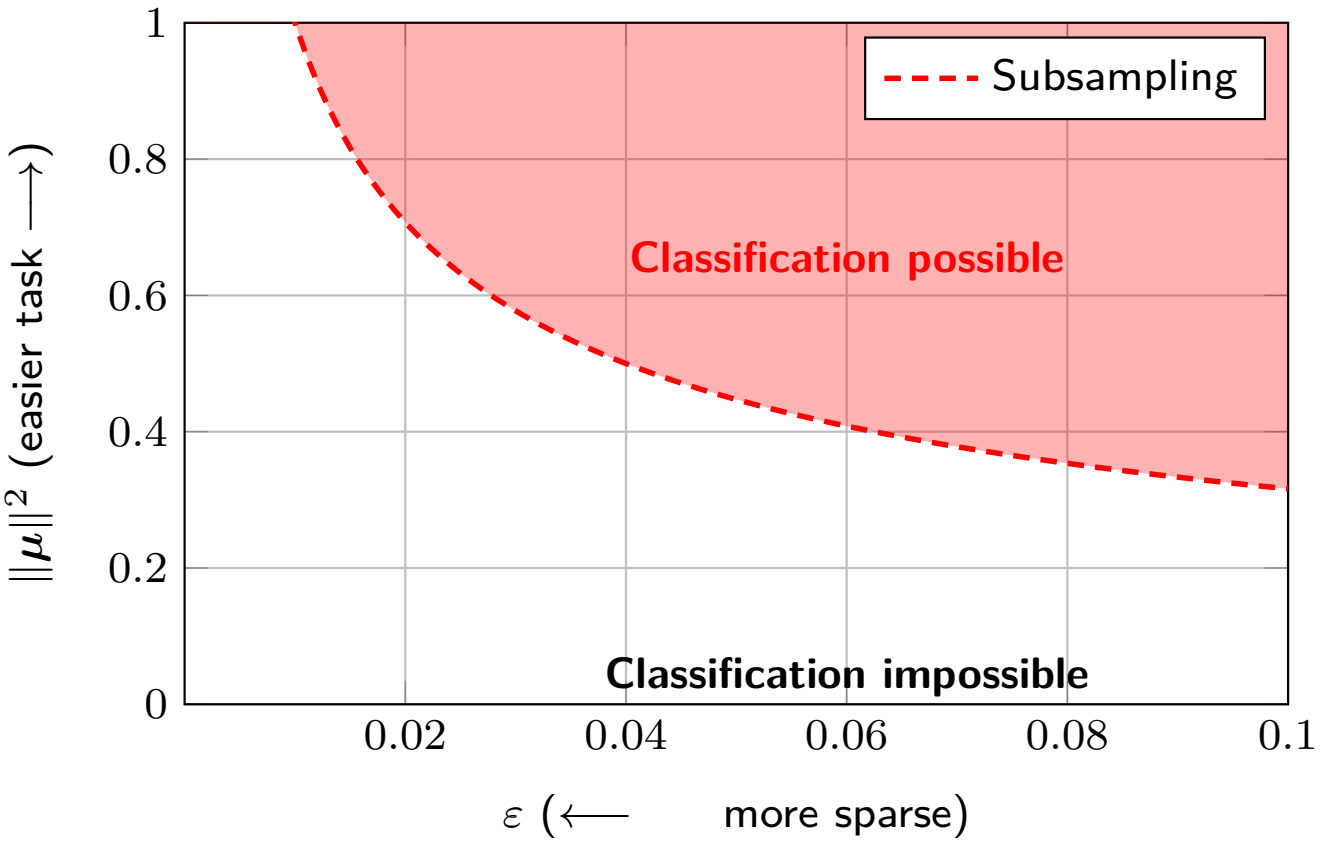
Even more striking: new intuitions and cheap algorithms

- **Computation cost reduction**: $(p, n \gg 1)$

$\rightarrow$ ε-subsampling $K \in \mathbb{R}^{n\varepsilon \times n\varepsilon}$



- **Phase transition of spectral clustering**: $(x_i \sim \mathcal{N}(\pm\mu, I_p),\ n/p = 100)$,

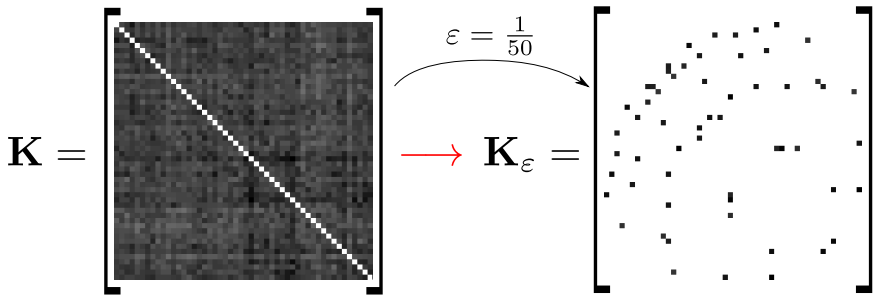


Even more striking: new intuitions and cheap algorithms

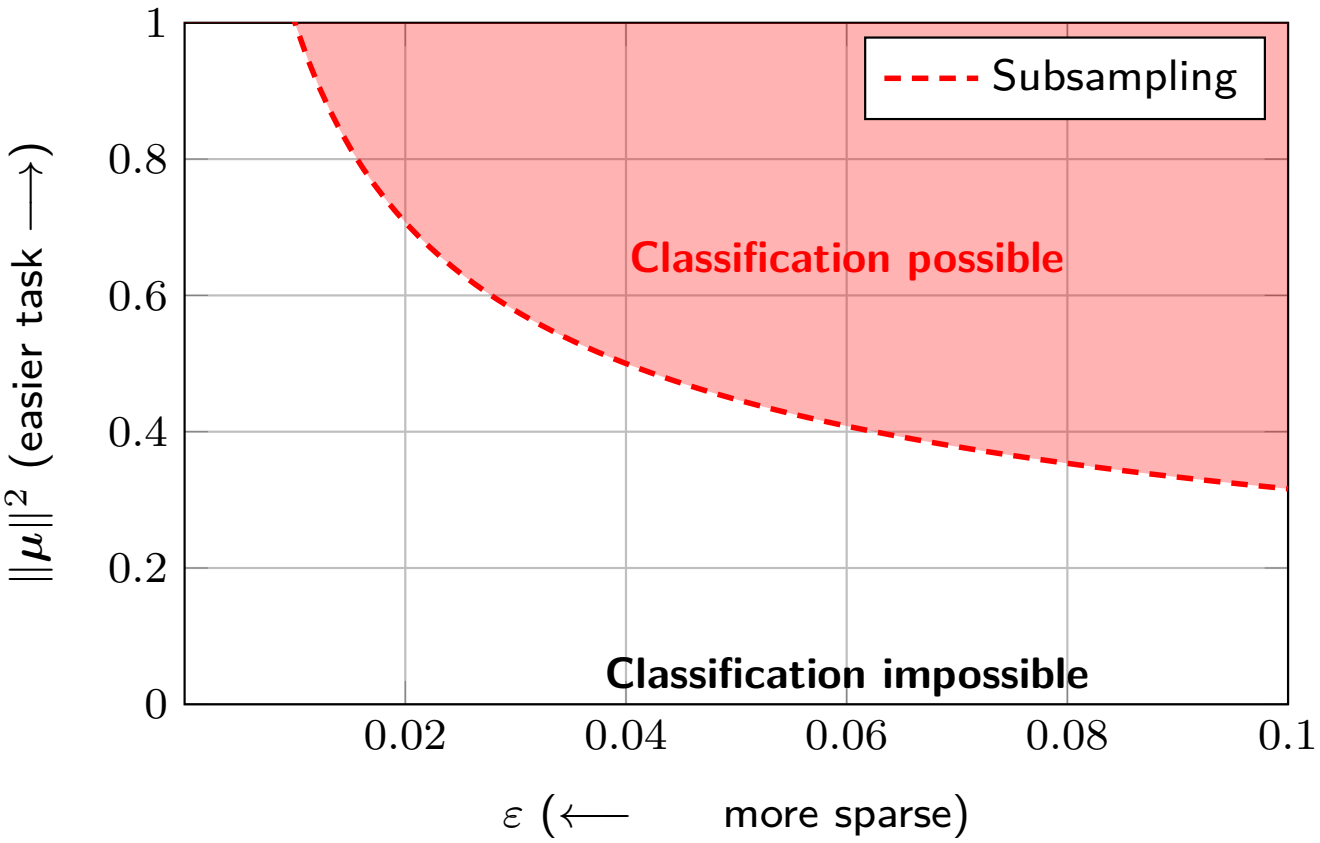
- **Computation cost reduction**: $(p, n \gg 1)$

  $\rightarrow$ ε-subsampling $K \in \mathbb{R}^{n\varepsilon \times n\varepsilon}$

  $\rightarrow$ $K_\varepsilon \equiv K \odot B$ with $B_{ij} \sim \mathrm{Bern}(\varepsilon)$ i.i.d.



- **Phase transition of spectral clustering**: $(x_i \sim \mathcal{N}(\pm\mu, I_p),\ n/p = 100)$,

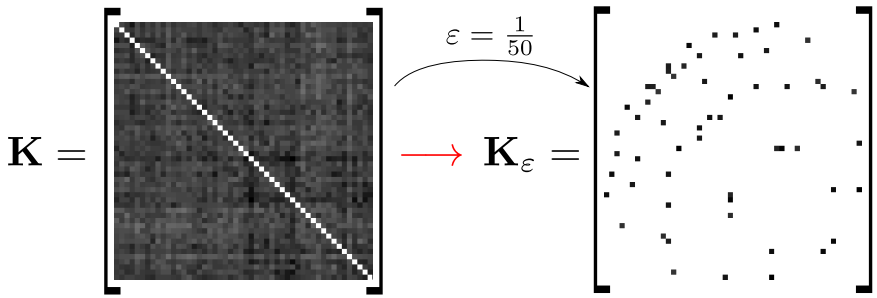


Even more striking: new intuitions and cheap algorithms

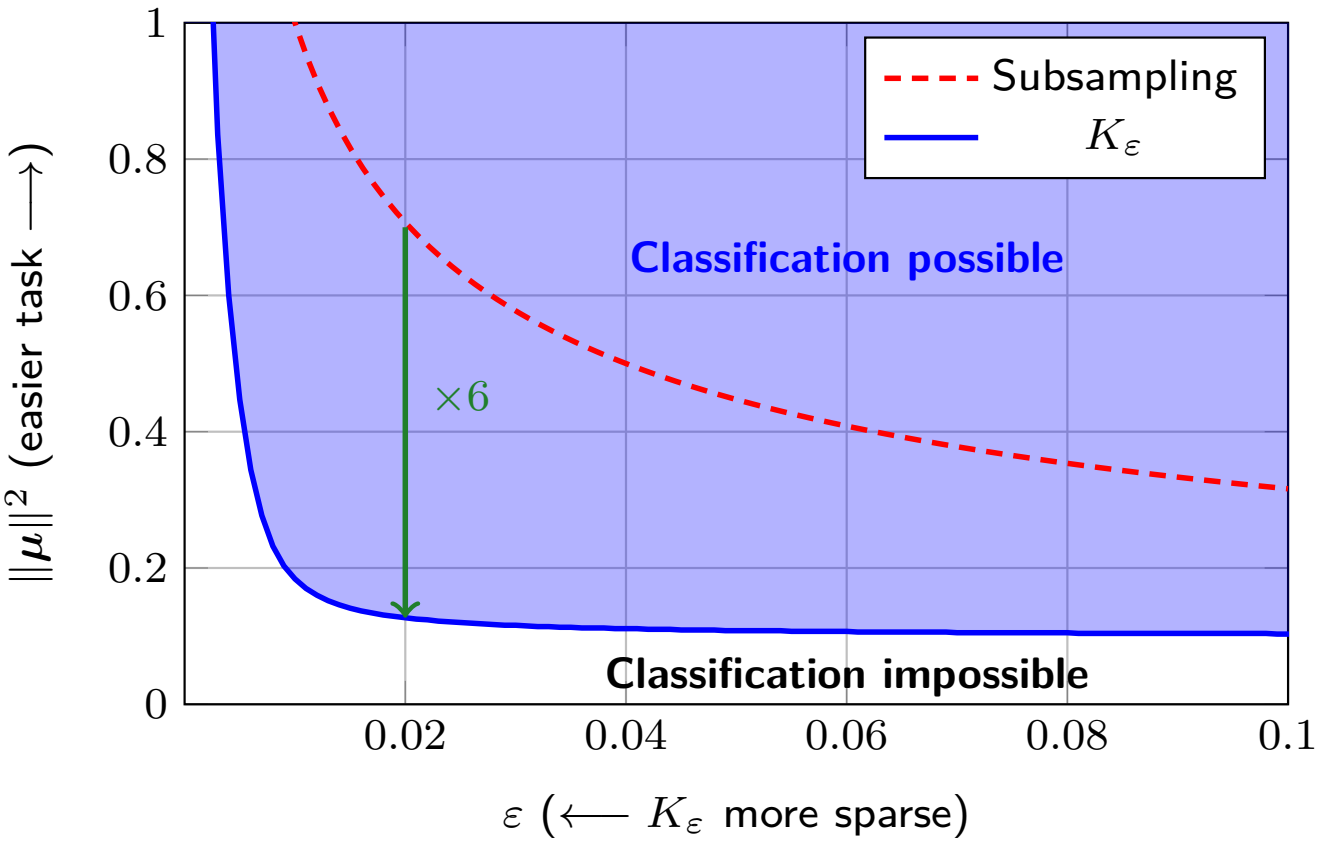
- **Computation cost reduction**: $(p, n \gg 1)$

  $\rightarrow$ ε-subsampling $K \in \mathbb{R}^{n\varepsilon \times n\varepsilon}$

  $\rightarrow$ $K_\varepsilon \equiv K \odot B$ with $B_{ij} \sim \mathrm{Bern}(\varepsilon)$ i.i.d.



- **Phase transition of spectral clustering**: $(x_i \sim \mathcal{N}(\pm\mu, I_p),\ n/p = 100)$,

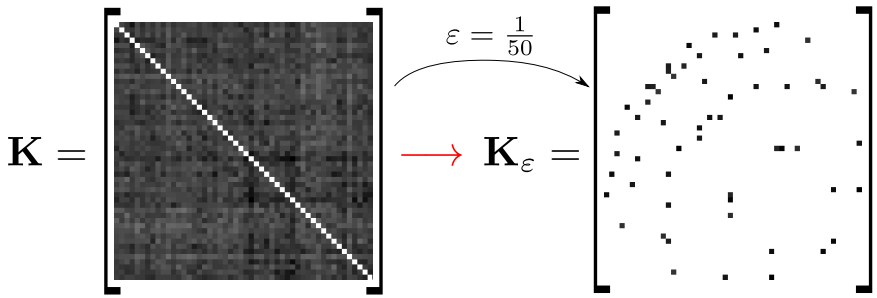


Even more striking: new intuitions and cheap algorithms

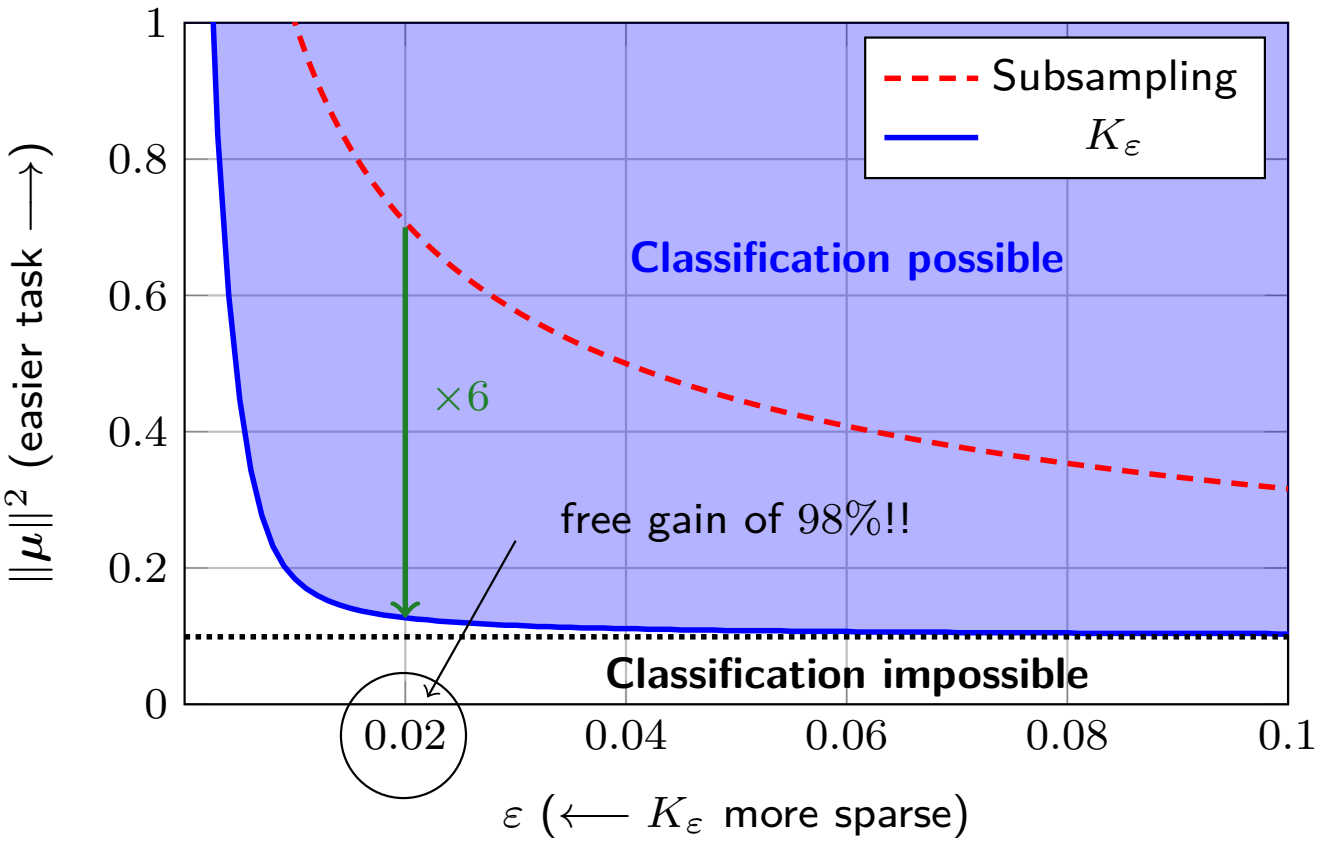
- **Computation cost reduction**: $(p, n \gg 1)$

  $\rightarrow$ ε-subsampling $K \in \mathbb{R}^{n\varepsilon \times n\varepsilon}$

  $\rightarrow$ $K_\varepsilon \equiv K \odot B$ with $B_{ij} \sim \mathrm{Bern}(\varepsilon)$ i.i.d.



- **Phase transition of spectral clustering**: $(x_i \sim \mathcal{N}(\pm\mu, I_p),\ n/p = 100)$,



Takeaway Message 3

"RMT Also Grasps 'Real Data' Processing"

From i.i.d. to concentrated random vectors

Beyond Gaussian Mixtures: results still valid for **concentrated random vectors.**

From i.i.d. to concentrated random vectors

Beyond Gaussian Mixtures: results still valid for **concentrated random vectors.**

Definition (Concentrated Random Vector)

$x \in \mathbb{R}^p$ is concentrated if, for all Lipschitz $f : \mathbb{R}^p \to \mathbb{R}$, there exists $m_f \in \mathbb{R}$, such that

$$P\left(|f(x) - m_f| > \varepsilon\right) \leq e^{-g(\varepsilon)}, \quad g \text{ increasing function.}$$

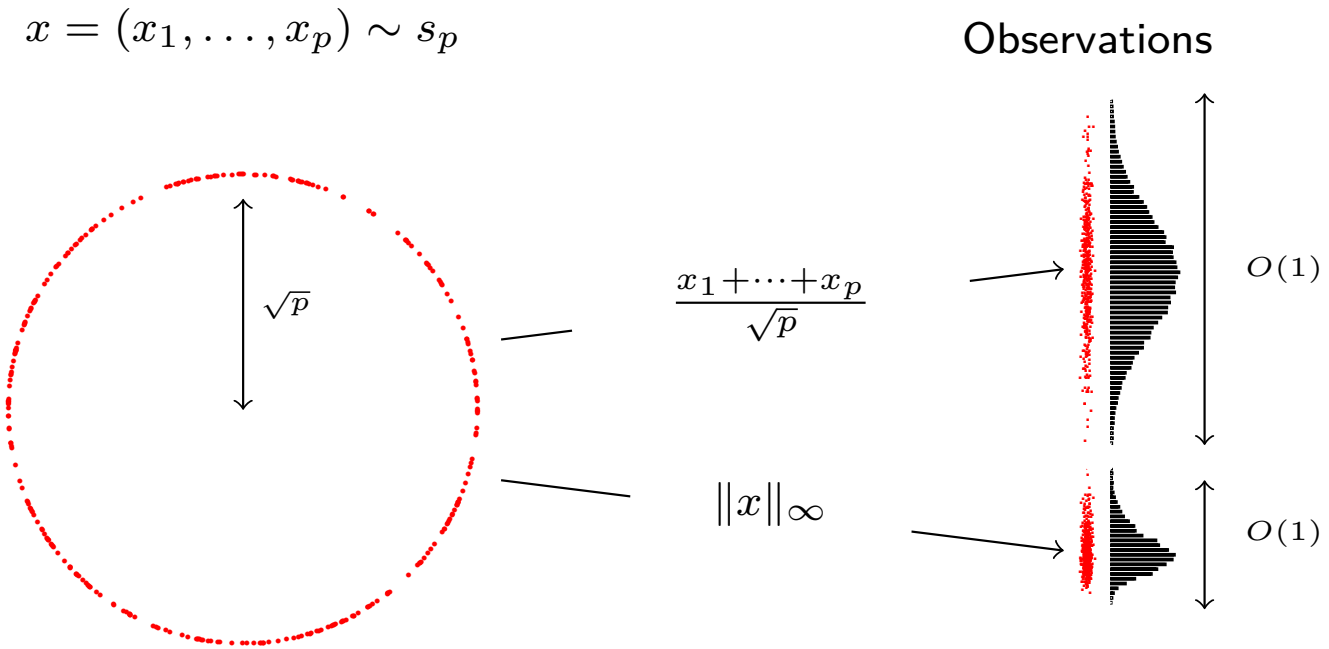
From i.i.d. to concentrated random vectors

Beyond Gaussian Mixtures: results still valid for **concentrated random vectors.**

Definition (Concentrated Random Vector)

$x \in \mathbb{R}^p$ is concentrated if, for all Lipschitz $f : \mathbb{R}^p \to \mathbb{R}$, there exists $m_f \in \mathbb{R}$, such that

$$P\left(|f(x) - m_f| > \varepsilon\right) \leq e^{-g(\varepsilon)}, \quad g \text{ increasing function.}$$



From i.i.d. to concentrated random vectors

Theorem ([Louart,C'18] [Seddik,C'19] Kernel Universality)

For $x_i \sim \mathcal{L}(\mu_a, C_a)$ **concentrated random vector**, *under the conditions of [C-Benaych'16],*

$$\|K - \hat{K}\| \xrightarrow{\text{a.s.}} 0, \quad \hat{K} = f(\tau) 1_n 1_n^\mathsf{T} + \frac{1}{p} Z Z^\mathsf{T} + JAJ^\mathsf{T} + *$$

with A only dependent on $f(\tau), f'(\tau), f''(\tau)$, $\mu_1, \ldots, \mu_k$, $C_1, \ldots, C_k$.

From i.i.d. to concentrated random vectors

Theorem ([Louart,C'18] [Seddik,C'19] Kernel Universality)

For $x_i \sim \mathcal{L}(\mu_a, C_a)$ **concentrated random vector**, *under the conditions of [C-Benaych'16],*

$$\|K - \hat{K}\| \xrightarrow{\text{a.s.}} 0, \quad \hat{K} = f(\tau) 1_n 1_n^\mathsf{T} + \frac{1}{p} Z Z^\mathsf{T} + J A J^\mathsf{T} + *$$

with A only dependent on $f(\tau), f'(\tau), f''(\tau)$, $\mu_1, \ldots, \mu_k$, $C_1, \ldots, C_k$.

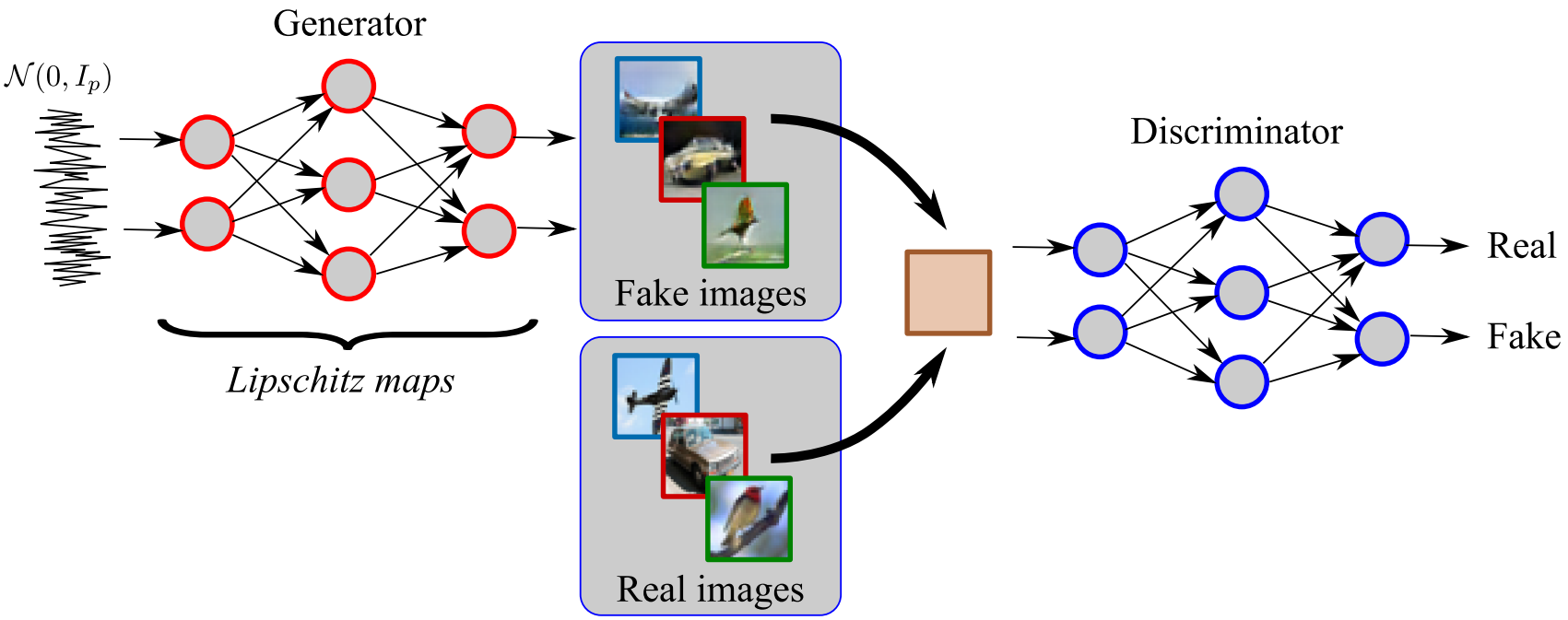
➪ **Same result as [C-Benaych'16]... Universality of first two moments!**

Ok. . . so what?

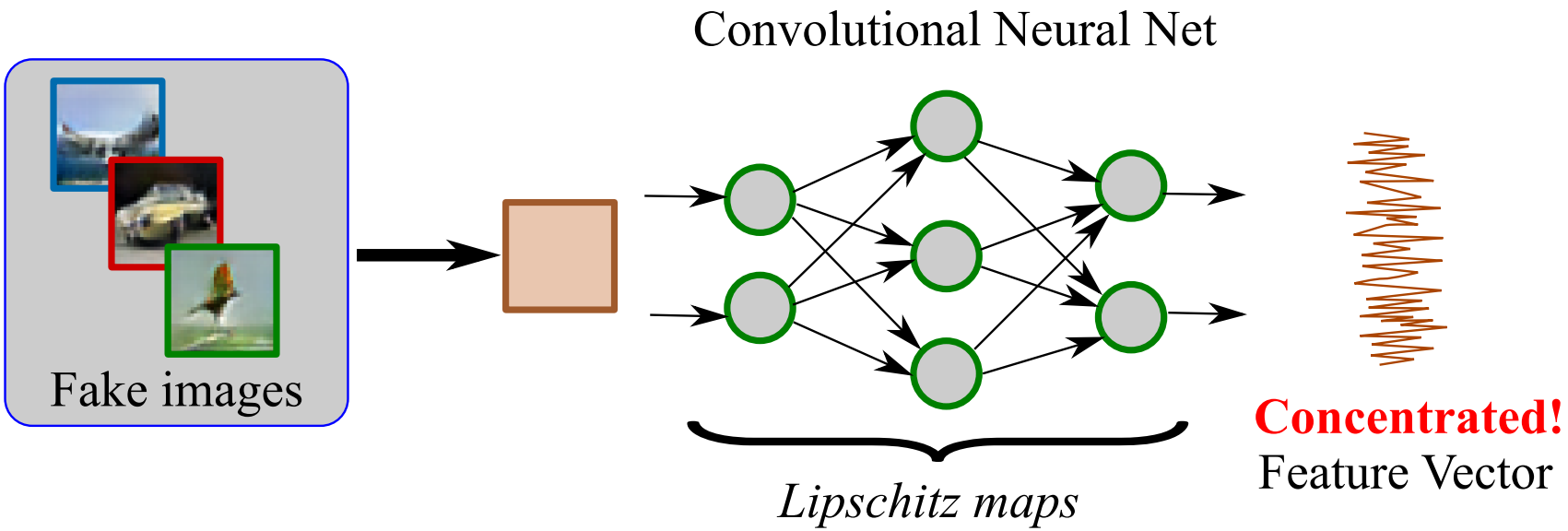
Key Finding. GAN-generated data **are concentrated random vectors!**

Ok. . . so what?

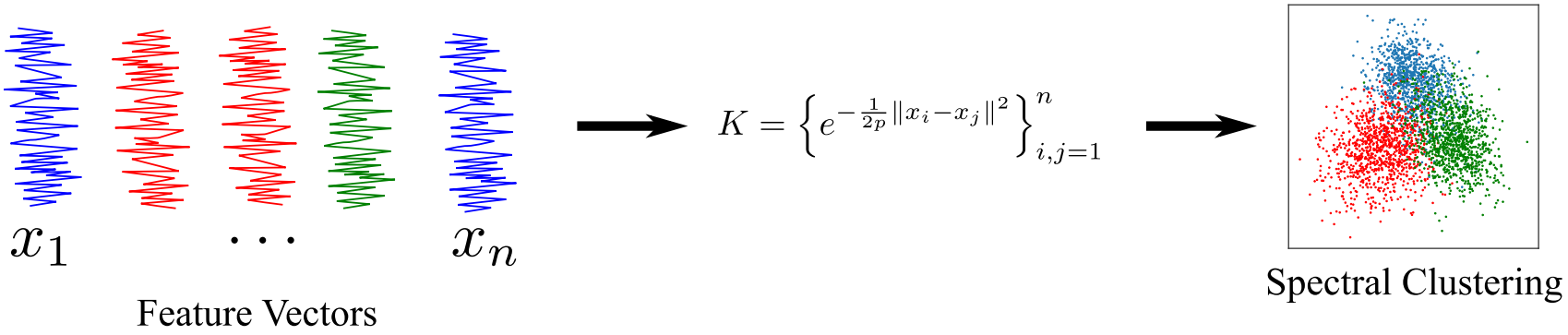
Key Finding. GAN-generated data **are concentrated random vectors!**



Ok. . . so what?


Convolutional Neural Net
Fake images
Lipschitz maps
Concentrated!
Feature Vector

Ok. . . so what?

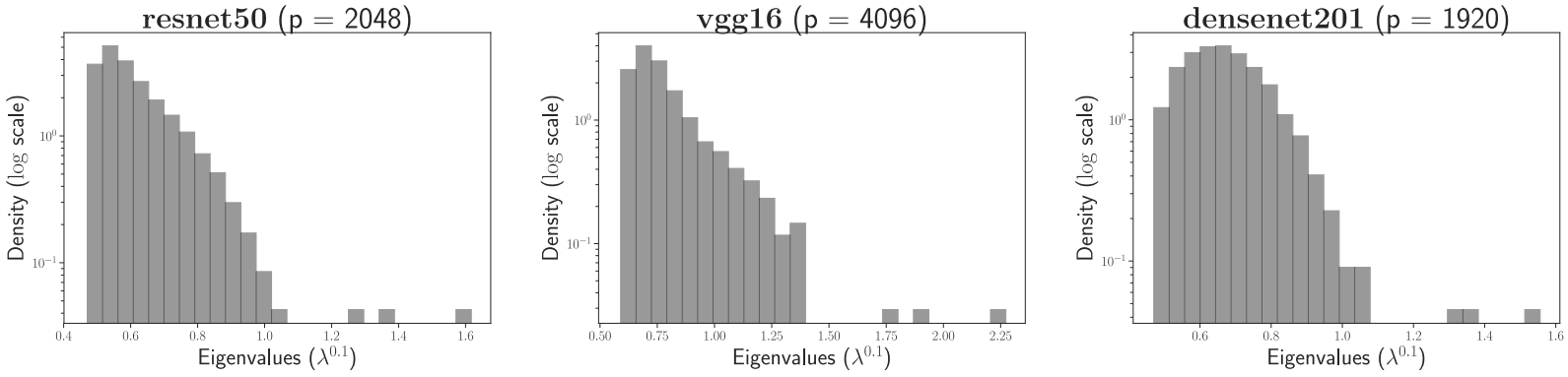

x_1
$\cdots$
x_n
Feature Vectors
$K = \left\{ e^{-\frac{1}{2p}\|x_i - x_j\|^2} \right\}_{i,j=1}^{n}$
Spectral Clustering

Gaussian, GAN, and real data

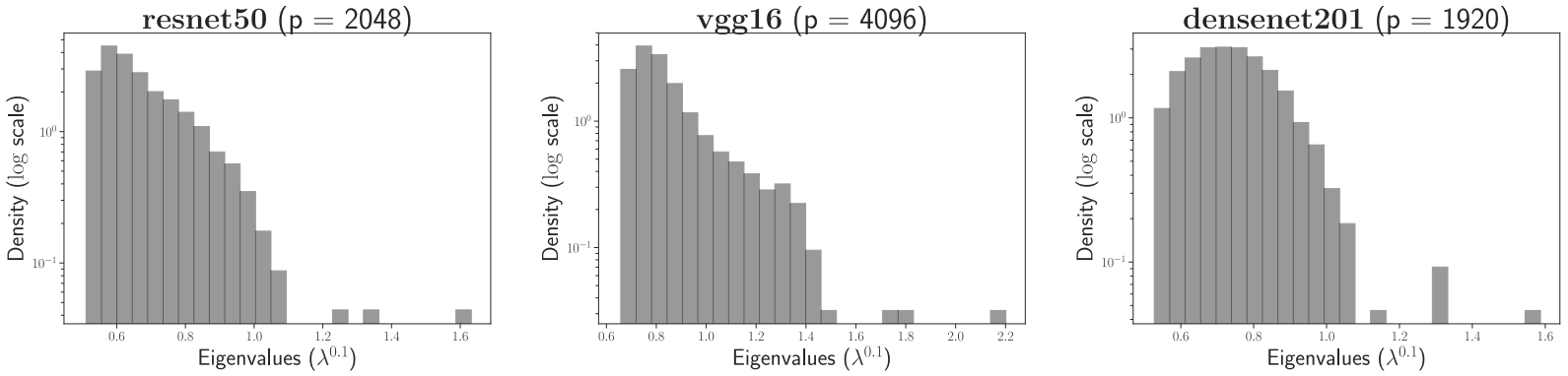
Results. **[Seddik,C'19]**

Gaussian, GAN, and real data

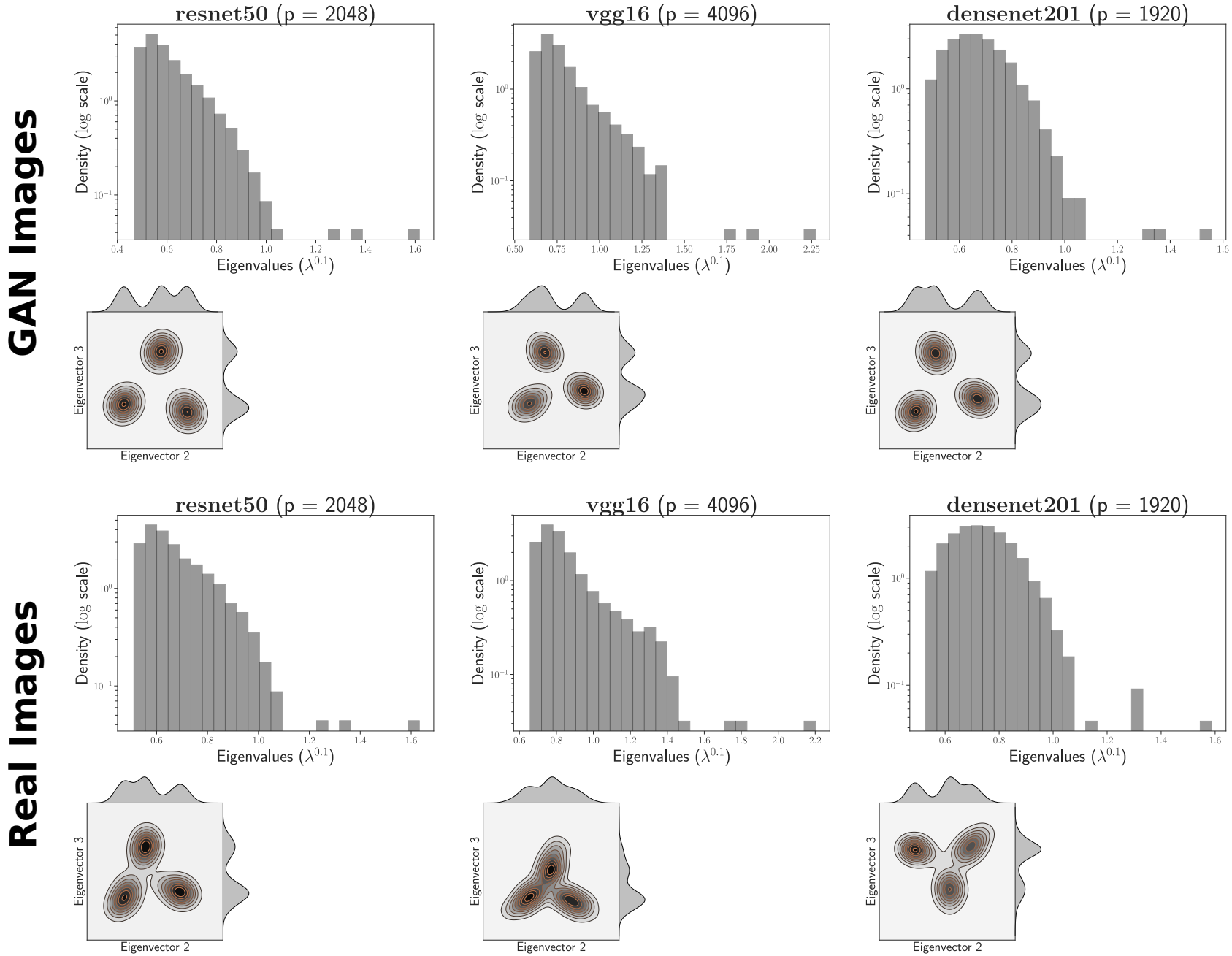
GAN Images



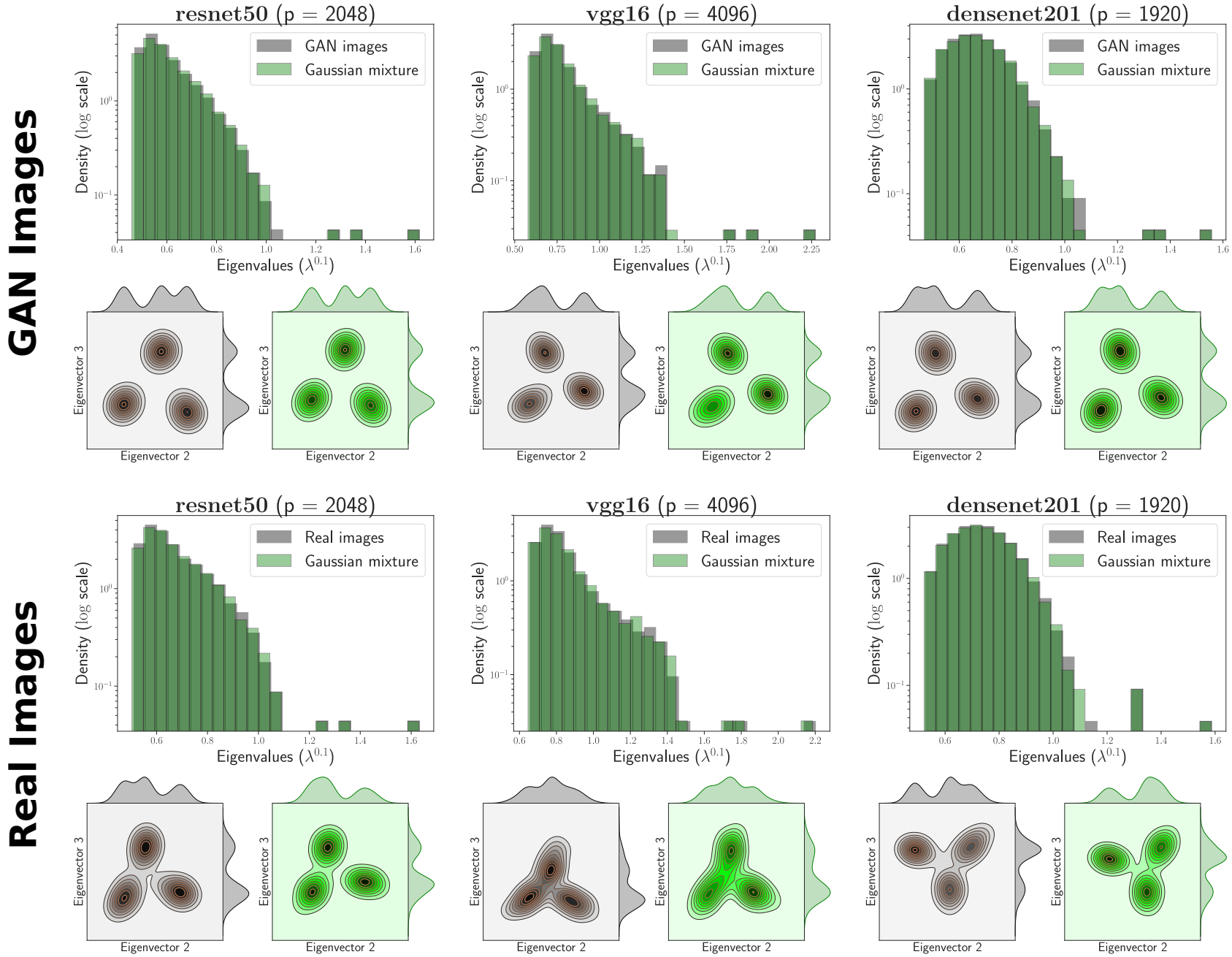
Real Images



Gaussian, GAN, and real data



Gaussian, GAN, and real data

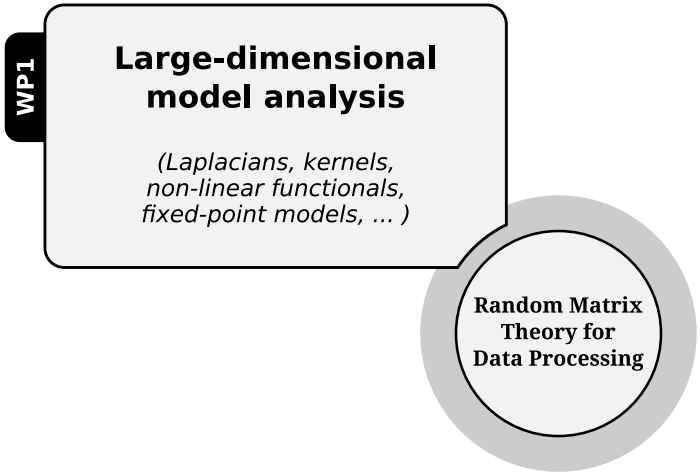


Conclusion

Our Research Activities:

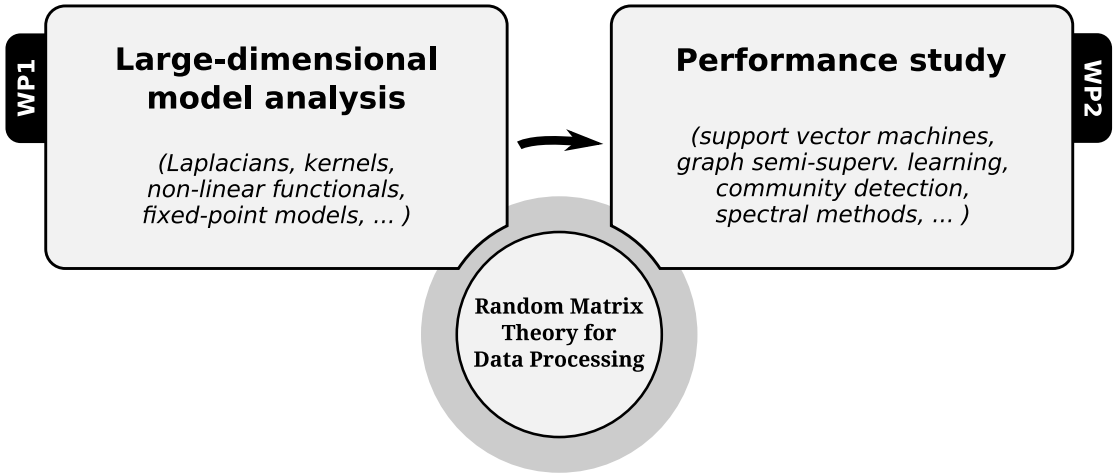
Conclusion

Our Research Activities:



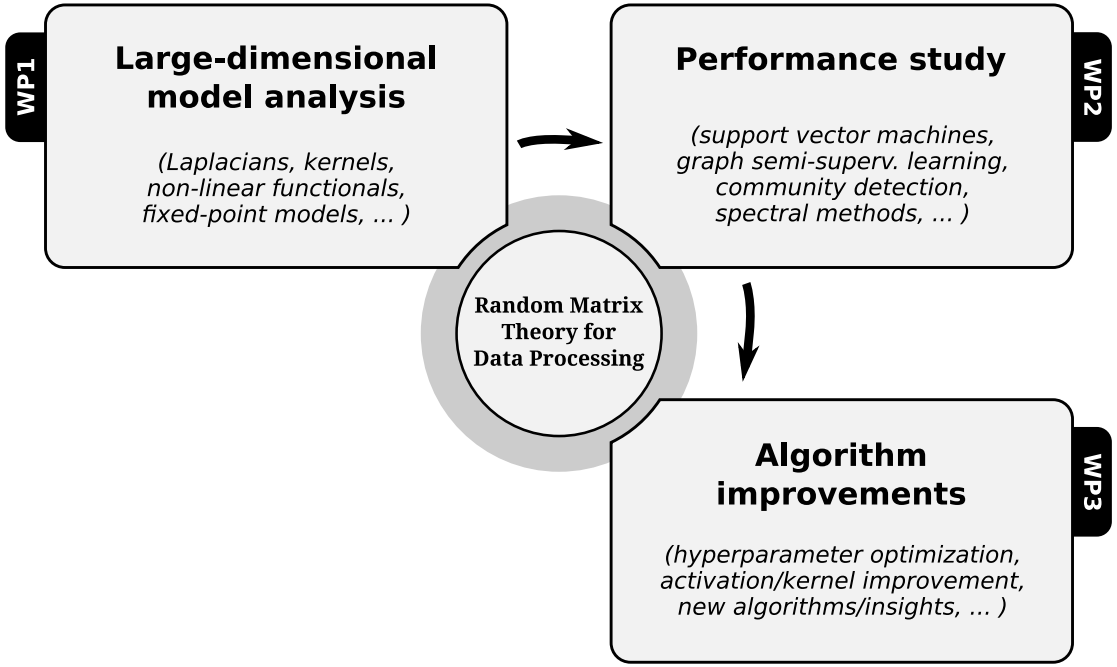
Conclusion

Our Research Activities:



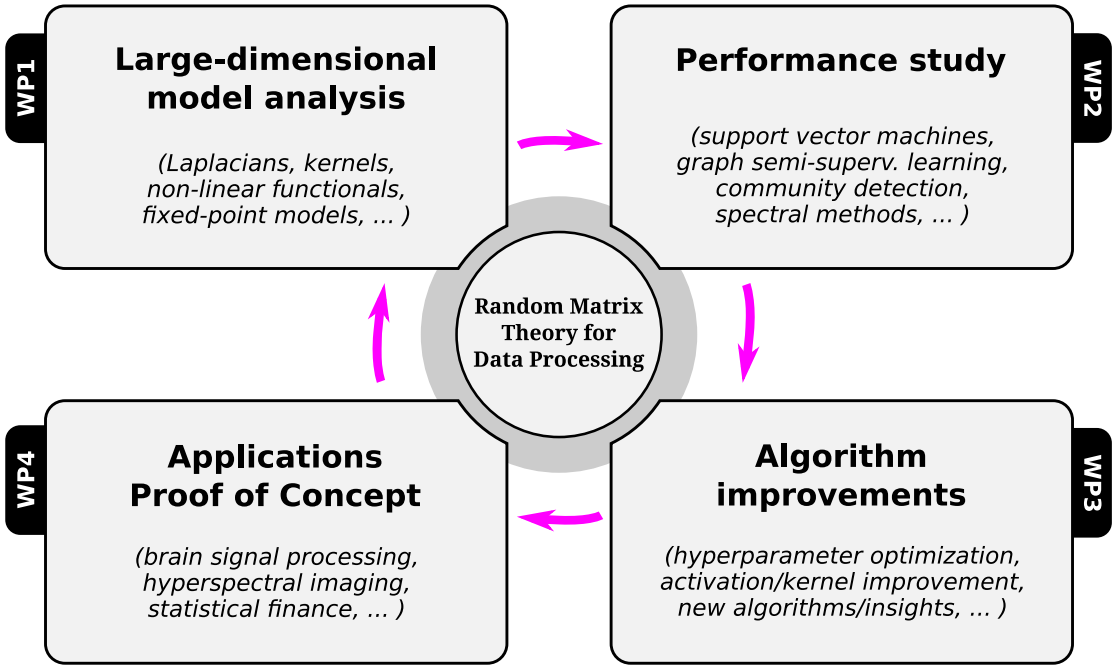
Conclusion

Our Research Activities:



Conclusion

Our Research Activities:

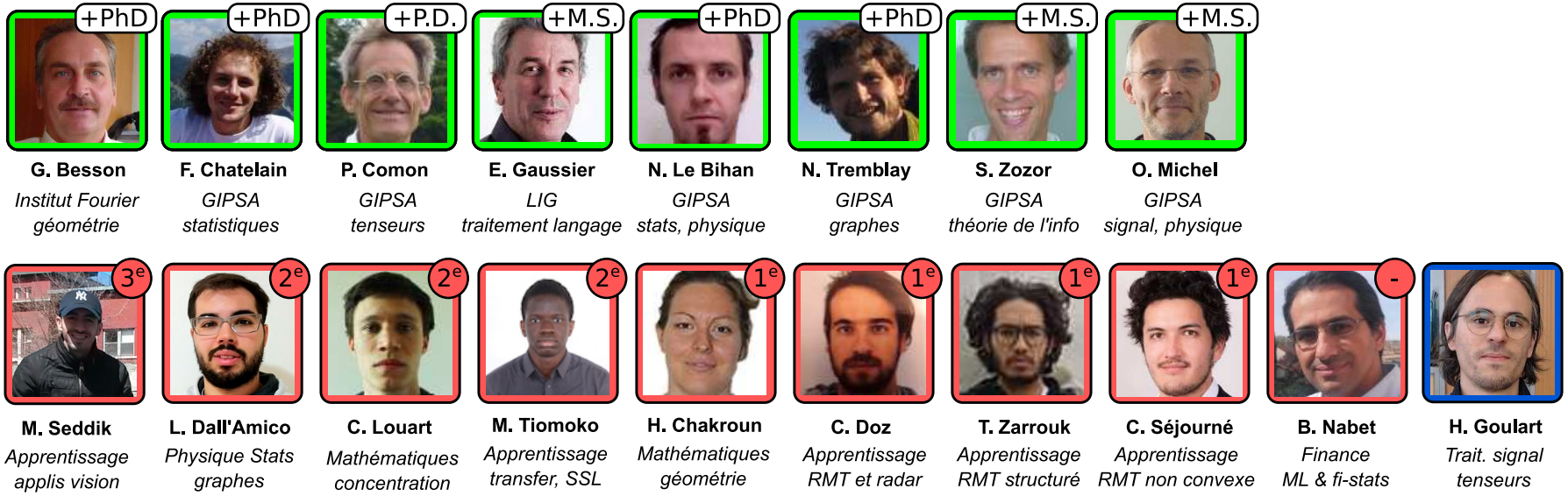


Our Team: the MIAI “LargeDATA” chair @ University Grenoble-Alpes

G. Besson — *Institut Fourier, géométrie*
F. Chatelain — *GIPSA, statistiques*
P. Comon — *GIPSA, tenseurs*
E. Gaussier — *LIG, traitement langage*
N. Le Bihan — *GIPSA, stats, physique*
N. Tremblay — *GIPSA, graphes*
S. Zozor — *GIPSA, théorie de l'info*
O. Michel — *GIPSA, signal, physique*

M. Seddik — *Apprentissage, applis vision*
L. Dall'Amico — *Physique Stats, graphes*
C. Louart — *Mathématiques, concentration*
M. Tiomoko — *Apprentissage, transfer, SSL*
H. Chakroun — *Mathématiques, géométrie*
C. Doz — *Apprentissage, RMT et radar*
T. Zarrouk — *Apprentissage, RMT structuré*
C. Séjourné — *Apprentissage, RMT non convexe*
B. Nabet — *Finance, ML & fi-stats*
H. Goulart — *Trait. signal, tenseurs*

Our Team: the MIAI “LargeDATA” chair @ University Grenoble-Alpes



Join us !

Thank you!

- **[C-Benaych'16]** R. Couillet, Benaych-Georges, "Kernel Spectral Clustering of Large Dimensional Data", Electronic Journal of Statistics, vol. 10, no. 1, pp. 1393-1454, 2016. [article]
- **[Mai,C'18]** X. Mai, R. Couillet, "A random matrix analysis and improvement of semi-supervised learning for large dimensional data", Journal of Machine Learning Research, vol. 19, no. 79, pp. 1-27, 2018. [article]
- **[Louart,C'18]** C. Louart, Z. Liao, R. Couillet, "A Random Matrix Approach to Neural Networks", The Annals of Applied Probability, vol. 28, no. 2, pp. 1190-1248, 2018. [article]
- **[Seddik,C'19]** M. Seddik, M. Tamaazousti, R. Couillet, "Kernel Random Matrices of Large Concentrated Data: The Example of GAN-Generated Image", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19), Brighton, UK, 2019. [article]
- H. Tiomoko Ali, R. Couillet, "Improved spectral community detection in large heterogeneous networks", Journal of Machine Learning Research, vol. 18, no. 225, pp. 1-49, 2018. [article]
- R. Couillet, M. Tiomoko, S. Zozor, E. Moisan, "Random matrix-improved estimation of covariance matrix distances", Journal of Multivariate Analysis, vol. 174, pp. 104531, 2019. [article]
- Z. Liao, R. Couillet, "A Large Dimensional Analysis of Least Squares Support Vector Machines", IEEE Transactions on Signal Processing, vol. 67, no.4, pp. 1065-1074, 2018. [article]