Soutenance de thèse.

Quelques contributions à l'analyse statistique de données à structure de graphe.

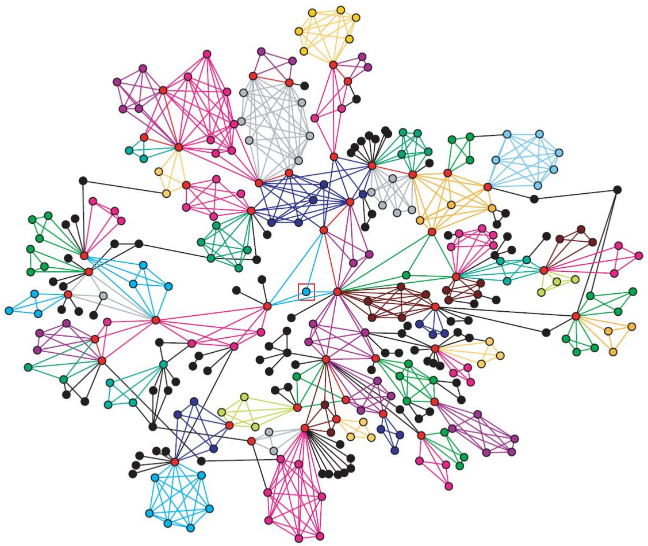
5 Décembre 2022

Etienne Lasalle

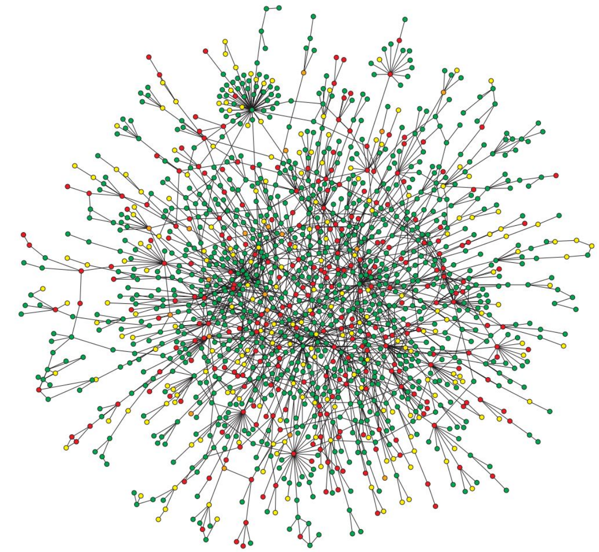
sous la direction de Pascal MASSART et Frédéric CHAZAL.

etienne.lasalle@universite-paris-saclay.fr

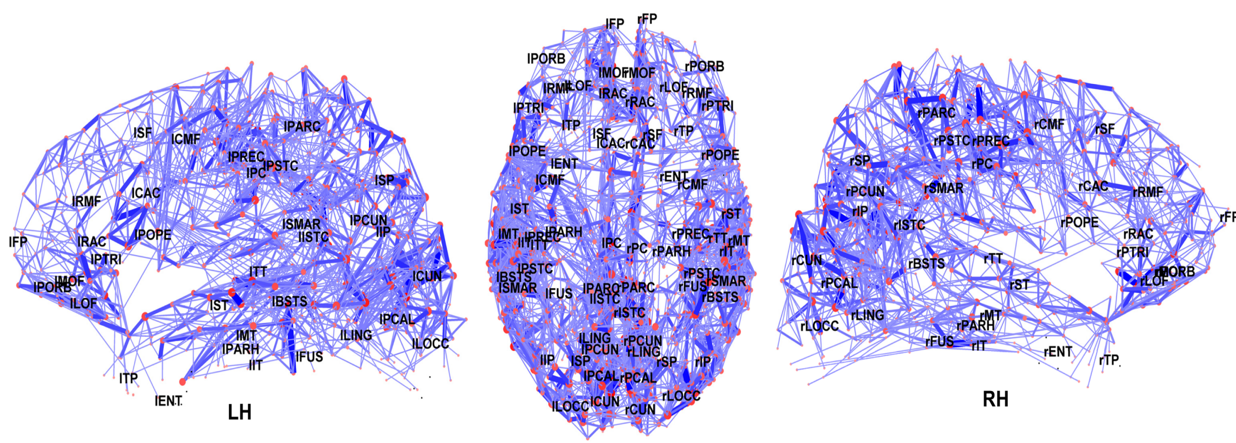
Theme of this thesis : graphs.



Co-authorship Network [PBV07]



Protein-protein interaction network [BO04]



Brain network [HCG+08]

[BO04] : Network biology: understanding the cell's functional organization., Barabasi *et al.*, 2008
[HCG+08] : Mapping the structural core of human cerebral cortex, Hagmann *et al.*, 2008
[PBV07] : Quantifying social group evolution, Palla *et al.*, 2007

Objectives

Goals :

- comparison of graph samples

$$(G_1, \ldots, G_N) \overset{i.i.d.}{\sim} P \quad ; \qquad (G'_1, \ldots, G'_M) \overset{i.i.d.}{\sim} Q$$

- theoretical results (asymptotic in sample size)

Requirements :

- take into account topological information
- graphs can be weighted
- graph sizes (same/different)
- node correspondance (known/unknown)

Outline

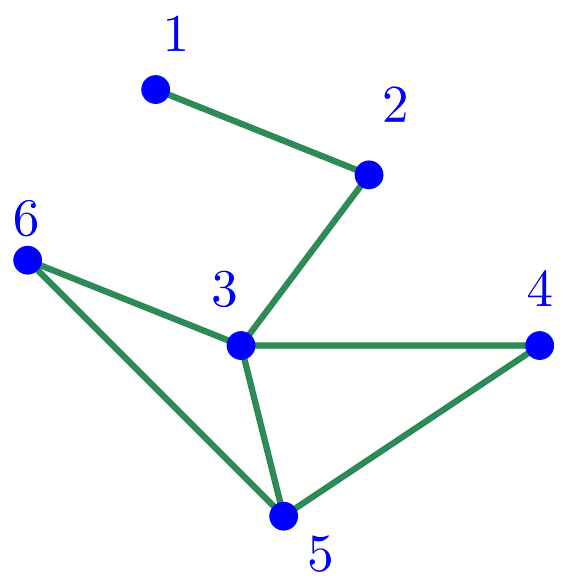
1. Heat diffusion distance processes [L21]
   - Distances
   - Processes

2. Functional central limit theorem and beyond [L21]
   - Donsker theorem and Gaussian approximation
   - Two-sample test

3. Detecting distribution shift
   - Experiments with MNIST
   - Experiments with Ripsnet

[L21] Heat diffusion distance processes: a statistically founded method to analyze graph data sets, L. , 2021, (Journal of Applied and Computational Topology).

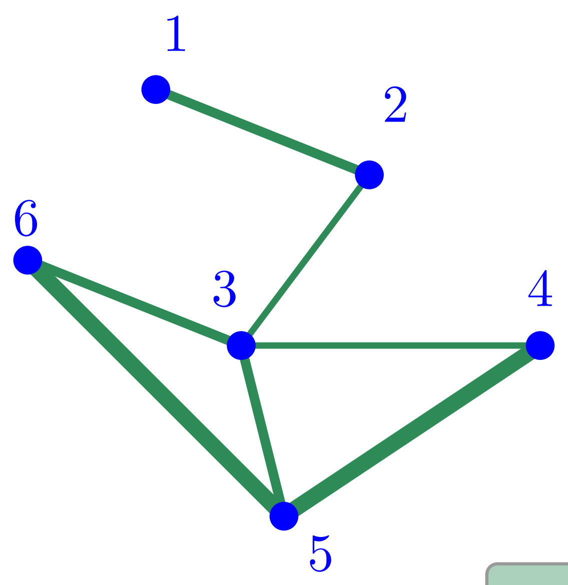
Mathematical representations.



$$\begin{array}{c c}
 & \begin{array}{cccccc} (1) & (2) & (3) & (4) & (5) & (6) \end{array} \\
\begin{array}{c} (1) \\ (2) \\ (3) \\ (4) \\ (5) \\ (6) \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}
\end{array} = A$$

Adjacency Matrix

Mathematical representations.

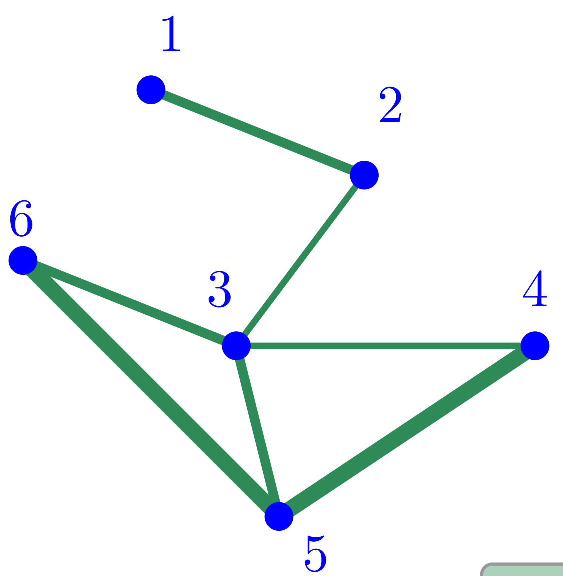


$w_{i,j} \in \mathbb{R}_+$

$$\begin{array}{c} \\ (1) \\ (2) \\ (3) \\ (4) \\ (5) \\ (6) \end{array} \begin{array}{c} \begin{array}{cccccc} (1) & (2) & (3) & (4) & (5) & (6) \end{array} \\ \begin{pmatrix} 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 2 & 3 & 0 & 3 \\ 0 & 0 & 2 & 0 & 3 & 0 \end{pmatrix} \end{array} = W$$

Weight Matrix

Mathematical representations.



$w_{i,j} \in \mathbb{R}_+$

$$\begin{array}{c c} & \begin{array}{cccccc} (1) & (2) & (3) & (4) & (5) & (6) \end{array} \\ \begin{array}{c} (1) \\ (2) \\ (3) \\ (4) \\ (5) \\ (6) \end{array} & \begin{pmatrix} 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 2 & 3 & 0 & 3 \\ 0 & 0 & 2 & 0 & 3 & 0 \end{pmatrix} \end{array} = W$$

Weight Matrix

$$L := D - W = \begin{pmatrix} 2 & -2 & 0 & 0 & 0 & 0 \\ -2 & 3 & -1 & 0 & 0 & 0 \\ 0 & -1 & 6 & -1 & -2 & -2 \\ 0 & 0 & -1 & 4 & -3 & 0 \\ 0 & 0 & -2 & -3 & 8 & -3 \\ 0 & 0 & -2 & 0 & -3 & 5 \end{pmatrix}$$

Laplacian Matrix

$$u^T L u = \sum_{i \neq j} w_{i,j} (u_i - u_j)^2$$

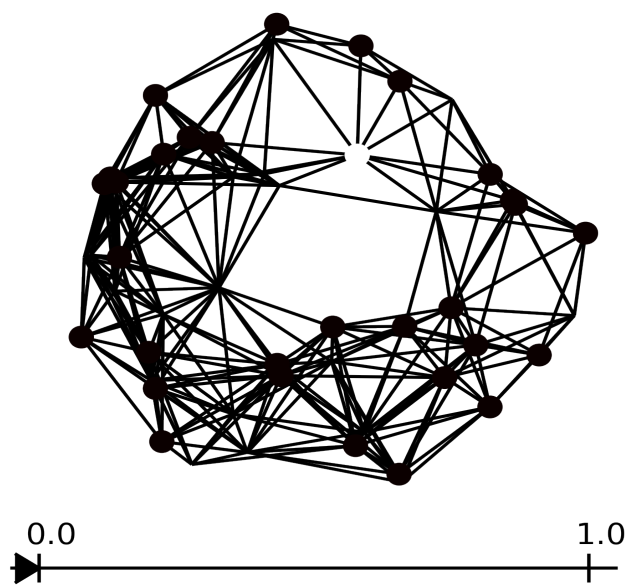
Heat diffusion on graphs.

Heat equation :

$u_0 \in \mathbb{R}^n$: initial heat distribution.

$$\frac{d}{dt} u_t = -L u_t, \quad t \geq 0$$

e^{-tL}, heat kernel at time t : $\quad u_t = e^{-tL} u_0$, $t \geq 0$:



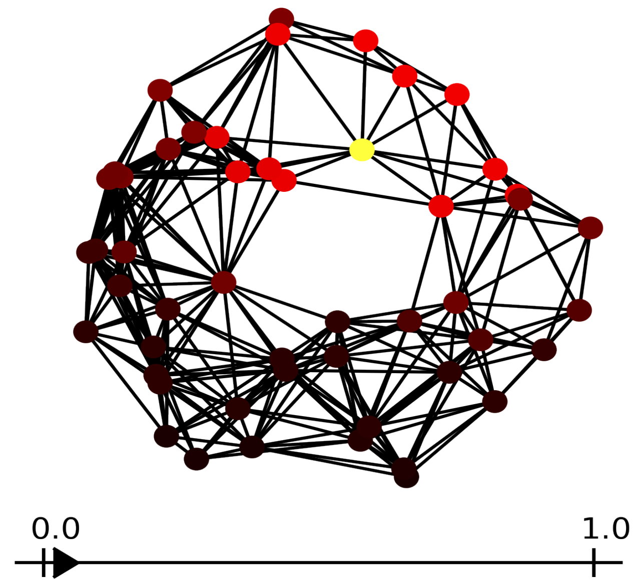
Heat diffusion on graphs.

Heat equation :

$u_0 \in \mathbb{R}^n$: initial heat distribution.

$$\frac{d}{dt} u_t = -L u_t, \quad t \geq 0$$

e^{-tL}, heat kernel at time t : $u_t = e^{-tL} u_0$, $t \geq 0$:



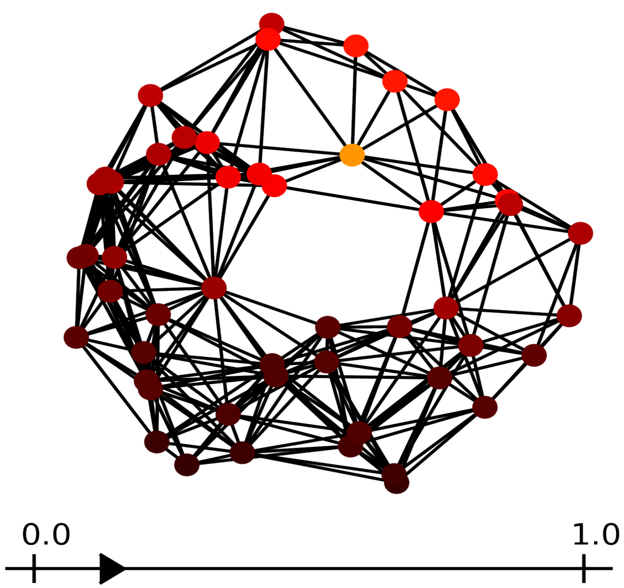
Heat diffusion on graphs.

Heat equation :

$u_0 \in \mathbb{R}^n$: initial heat distribution.

$$\frac{d}{dt} u_t = -L u_t, \quad t \geq 0$$

e^{-tL}, heat kernel at time t : $u_t = e^{-tL} u_0$, $t \geq 0$:



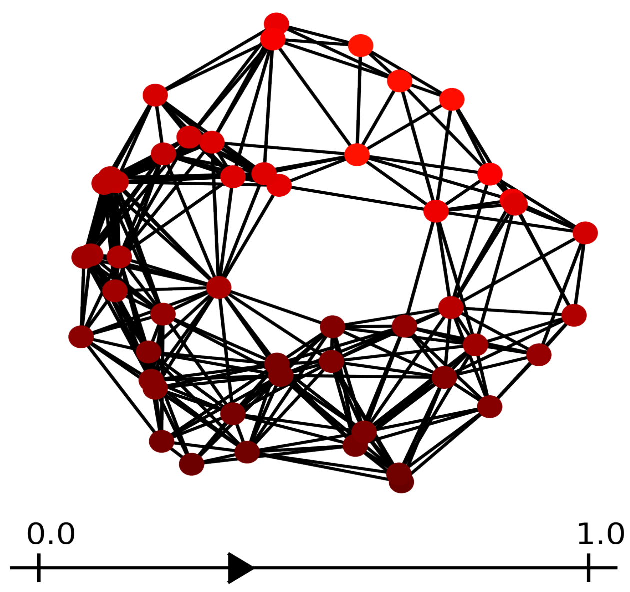
Heat diffusion on graphs.

Heat equation :

$u_0 \in \mathbb{R}^n$: initial heat distribution.

$$\frac{d}{dt} u_t = -L u_t, \quad t \geq 0$$

e^{-tL}, heat kernel at time t : $\quad u_t = e^{-tL} u_0$, $t \geq 0$:



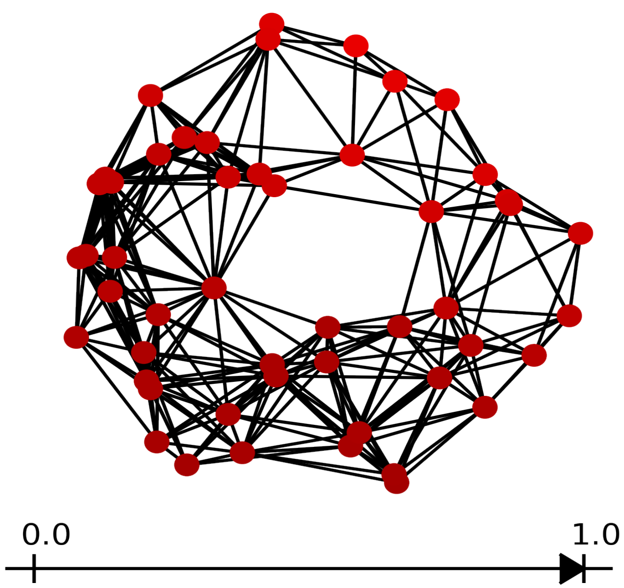
Heat diffusion on graphs.

Heat equation :

$u_0 \in \mathbb{R}^n$: initial heat distribution.

$$\frac{d}{dt} u_t = -L u_t, \quad t \geq 0$$

e^{-tL}, heat kernel at time t : $\quad u_t = e^{-tL} u_0$, $t \geq 0$:



Comparing graphs

Assumption :
same sizes n & known node correspondance.

Compare matrix representations :

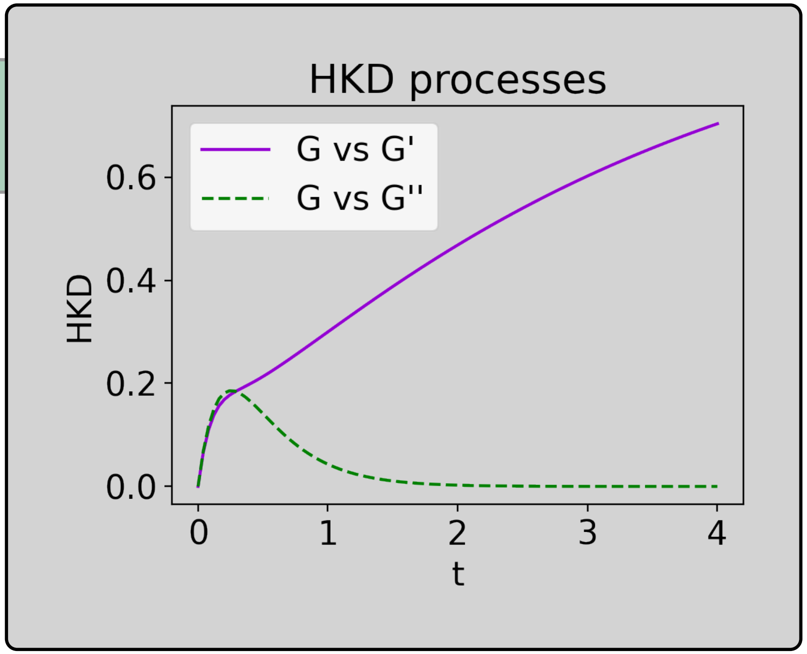
- Adjacency / weight matrix
- Laplacian matrix
- **Heat kernel**

Heat Kernel Distance : $D_t(G, G') = \|e^{-tL} - e^{-tL'}\|_F$ [HGJ13]

[HGJ13] : Graph diffusion distance: A difference measure for weighted graphs based on the graph Laplacian exponential kernel, Hammond *et al*, 2013

Comparing graphs

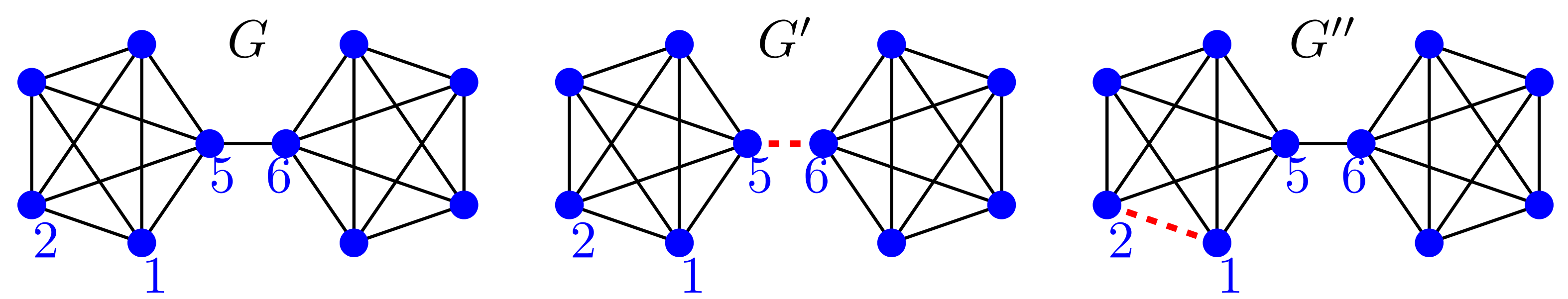
Assumption :
same sizes n & known node correspondance.



Compare matrix representations :

- Adjacency / weight matrix
- Laplacian matrix
- **Heat kernel**

Heat Kernel Distance : $D_t(G, G') = \|e^{-tL} - e^{-tL'}\|_F$ [HGJ13]



[HGJ13] : Graph diffusion distance: A difference measure for weighted graphs based on the graph Laplacian exponential kernel, Hammond *et al*, 2013

Comparing graphs

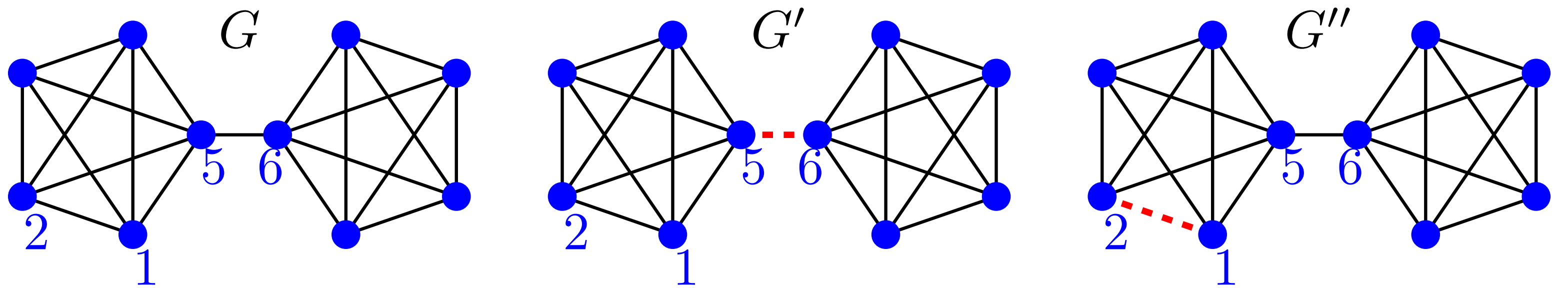
~~**Assumption :**~~
~~same sizes n & known node correspondance.~~

Compare matrix representations :

- Adjacency / weight matrix
- Laplacian matrix
- **Heat kernel**

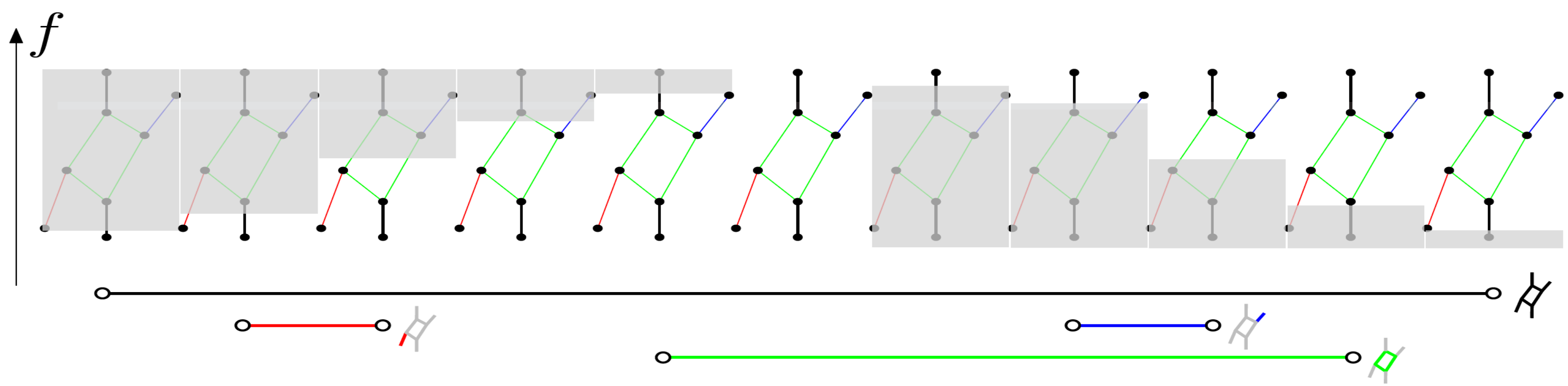
??

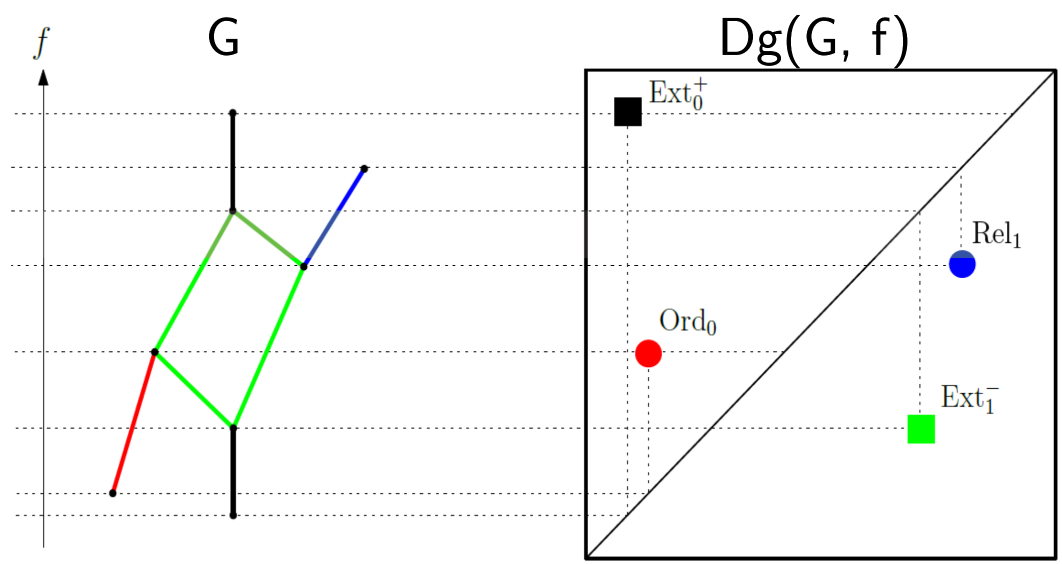
Heat Kernel Distance : $D_t(G, G') = \|e^{-tL} - e^{-tL'}\|_F$ [HGJ13]



[HGJ13] : Graph diffusion distance: A difference measure for weighted graphs based on the graph Laplacian exponential kernel, Hammond *et al*, 2013

Using Topological Data Analysis

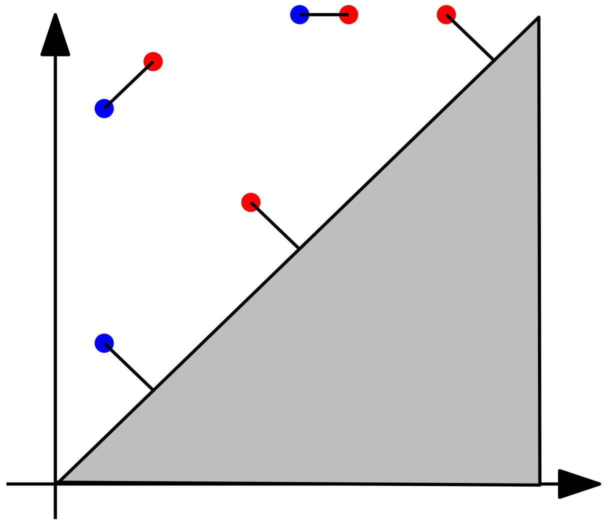




Figures from [CCIL+19]

[CCIL+19]: Perslay : A Neural Network Layer for Persistence Diagrams and New Graph Topological Signatures, Carriere, Chazal, Ike, Lacombe, Royer, Umeda, 2019

Comparing persistence diagrams



μ, ν : finite multisets of points in $\mathbb{R}^2$.
$\Delta = \{(a,a), \forall a \in \mathbb{R}\}$: diagonal

π : a matching from $\mu \cup \Delta$ to $\nu \cup \Delta$

$\Pi(\mu,\nu)$: set of all matchings

Bottleneck Distance :

$$d_B(\mu,\nu) := \inf_{\pi \in \Pi(\mu,\nu)} \sup_{x \in \mu \cup \Delta} \|x - \pi(x)\|_\infty$$

Another way to compare graphs

Heat Kernel Signature (HKS) : [SOG09] [HRG14]

$$h_t(G) : i \rightarrow (e^{-tL})_{i,i}$$ "Remaining heat at node i"

Graphs | Topological descriptors

$$\begin{tikzcd}
G \arrow[r, "\text{HKS}"] & (G, h_t(G)) \arrow[r, "\text{Pers. diagram}"] & Dg((G, h_t(G))) \\
G' \arrow[r] & (G', h_t(G')) \arrow[r] & Dg((G', h_t(G')))
\end{tikzcd}$$

Heat Persistence Distance (HPD) :

$$H_t(G, G') = \max_{D_g} \ d_B\left(Dg\left(G, h_t(G)\right), Dg\left(G', h_t(G')\right)\right)$$

[SOG09]: A concise and provably informative multiscale signature based on heat diffusion, Sun *et al*, 2009
[HRG14] : Stable and informative spectral signatures for graph matching, Hu *et al*, 2014

Recap of the distances.

Assumption :
same sizes n & known node correspondance.

Heat Kernel Distance (HKD):

$$D_t(G, G') = \|e^{-tL} - e^{-tL'}\|_F$$

Assumption :
No assumption.

Heat Persistence Distance (HPD) :

$$H_t(G, G') = \max_{D_g} \; d_B\left(Dg\left(G, h_t(G)\right), Dg\left(G', h_t(G')\right)\right)$$

Recap of the distances.

Assumption :
same sizes n & known node correspondance.

Heat Kernel Distance (HKD):

$$D_t(G, G') = \|e^{-tL} - e^{-tL'}\|_F$$

Assumption :
No assumption.

Heat Persistence Distance (HPD) :

$$H_t(G, G') = \max_{D_g} \; d_B\left(Dg\left(G, h_t(G)\right), Dg\left(G', h_t(G')\right)\right)$$

How can we choose t ?

To choose or not to choose?

Functional Point of View

$$D_\cdot(G,G') : \begin{array}{ccc} [0,T] & \mapsto & \mathbb{R} \\ t & \mapsto & D_t(G,G') \end{array} \quad \text{or} \quad H_\cdot(G,G') : \begin{array}{ccc} [0,T] & \mapsto & \mathbb{R} \\ t & \mapsto & H_t(G,G') \end{array}$$

Empirical Process Point of View

$$\{D_t(G,G'), \quad t \in [0,T]\} \quad \text{or} \quad \{H_t(G,G'), \quad t \in [0,T]\}$$

$$\mathcal{F}_{HKD} = \{D_t(\cdot), \quad t \in [0,T]\} \quad \text{or} \quad \mathcal{F}_{HPD} = \{H_t(\cdot), \quad t \in [0,T]\}$$

To choose or not to choose?

Functional Point of View

$$D_{\cdot}(G,G') : \begin{array}{ccc} [0,T] & \mapsto & \mathbb{R} \\ t & \mapsto & D_t(G,G') \end{array} \quad \text{or} \quad H_{\cdot}(G,G') : \begin{array}{ccc} [0,T] & \mapsto & \mathbb{R} \\ t & \mapsto & H_t(G,G') \end{array}$$

Empirical Process Point of View

$$\{D_t(G,G'), \quad t \in [0,T]\} \quad \text{or} \quad \{H_t(G,G'), \quad t \in [0,T]\}$$

$$\mathcal{F}_{HKD} = \{D_t(\cdot), \quad t \in [0,T]\} \quad \text{or} \quad \mathcal{F}_{HPD} = \{H_t(\cdot), \quad t \in [0,T]\}$$

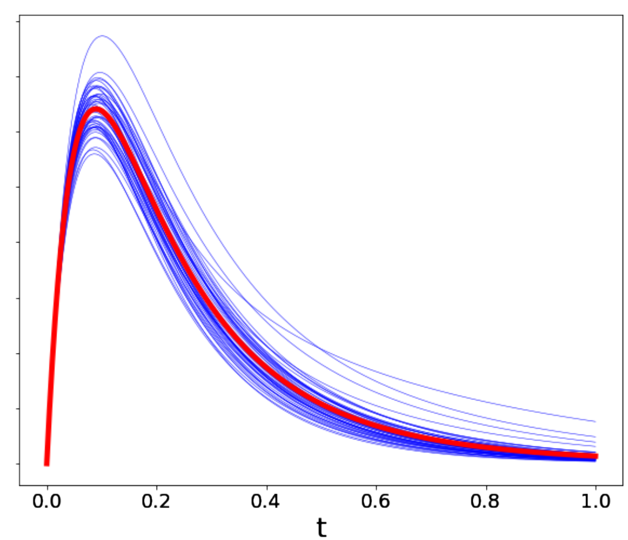
(G_0, G'_0) (G_1, G'_1)

(G_2, G'_2) $(G_{\dots}, G'_{\dots})$

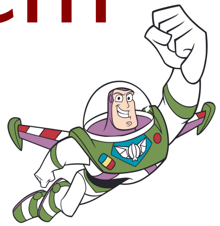
(G_N, G'_N)

Dataset

$\xrightarrow[\text{or } H_{\cdot}(\cdot)]{D_{\cdot}(\cdot)}$



2. Functional central limit theorem and beyond.



General empirical processes.

- $X_1, \ldots, X_N \sim P$ (i.i.d sample) $X_i \in \mathcal{X}$
- P_N : empirical measure
- $\mathcal{F} = \{f_t,\ t \in [0,T]\}$: a family of measurable functions.

t fixed,
$$\sqrt{N}(P_N - P)f_t = \sqrt{N}\Big(\frac{1}{N}\sum_{i=1}^{N} f_t(X_i) - \mathbb{E}_P\left[f_t(X)\right]\Big)$$
$$\xrightarrow{\mathcal{D}} \mathcal{N}(0, \sigma_t^2)$$

General empirical processes.

- $X_1, \ldots, X_N \sim P$ (i.i.d sample) $X_i \in \mathcal{X}$
- P_N : empirical measure
- $\mathcal{F} = \{f_t,\ t \in [0,T]\}$: a family of measurable functions.

t fixed,
$$\sqrt{N}(P_N - P)f_t = \sqrt{N}\Big(\frac{1}{N}\sum_{i=1}^{N} f_t(X_i) - \mathbb{E}_P\left[f_t(X)\right]\Big)$$
$$\xrightarrow{\mathcal{D}} \mathcal{N}(0, \sigma_t^2)$$

Definition : $\mathcal{F}$ is said to be *Donsker* if

$$\{\sqrt{N}(P_N - P)f_t,\ t \in [0,T]\} \xrightarrow{weak} \text{Gaussian Process } \mathbb{G}$$

$\forall h : \mathcal{C}([0,T]) \to \mathbb{R}$, continuous and bounded

$$\lim_{N\to\infty} \mathbb{E}\left[h\Big(\sqrt{N}(P_N - P)f_\cdot\Big)\right] = \mathbb{E}\left[h(\mathbb{G})\right]$$

Definition : $\{G_N f_t,\ t \in [0,T]\}$ admits a *Gaussian approximation* with rate r_N,

if $\forall \lambda > 1$, $\exists \rho, N_0$, such that $\forall N \geq N_0$
one can construct on the same probability space $X_1, \ldots, X_N$
and a version of the Gaussian limit process $\mathbb{G}^{(N)}$ such that

$$\mathbb{P}\left(\|G_N f_\cdot - \mathbb{G}_\cdot^{(N)}\|_\infty > \rho\, r_N\right) \leq N^{-\lambda}.$$

Main theoretical result

Assumptions :

(L) - $\exists k > 0$, $\forall x \in \mathcal{X}$, $t \to f_t(x)$ is k-Lipschitz continuous
(B) - $\exists M > 0$, $\forall x \in \mathcal{X}$, $\forall t \in [0, T]$, $|f_t(x)| \leq M$

Result :

- $\mathcal{F}$ is Donsker
- $\{G_N f_t,\ t \in [0, T]\}$ admits a Gaussian approximation with

$$r_N = N^{-1/7}(\log N)^{9/14}$$

[L21]

Refinement :

- $\rho = B_1 + B_2\sqrt{\lambda + 1} + B_3 M(kT + M)^{5/2}(\lambda + 2/7)$
- $N_0 = \mathcal{O}\left(M^{7+u} + (kT + M)^{2+u}\right)$

Applications : [L21]

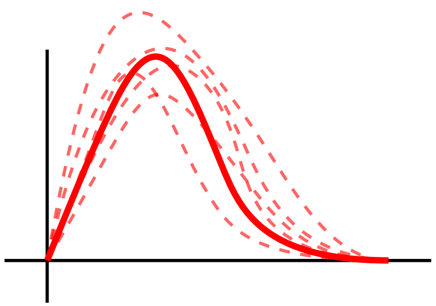
- $\forall (G, G')$ of size n, with weights in $[0, w_{\max}]$,
  $t \to D_t(G, G')$ is $n^{3/2} w_{\max}$-lipschitz continuous and bounded by $\sqrt{n}$
- $\forall (G, G')$ of size at most n, with weights in $[0, w_{\max}]$,
  $t \to H_t(G, G')$ is $2n w_{\max}$-lipschitz continuous and bounded by 1.

[L21] Heat diffusion distance processes: a statistically founded method to analyze graph data sets, L. , 2021, (Journal of Applied and Computational Topology).

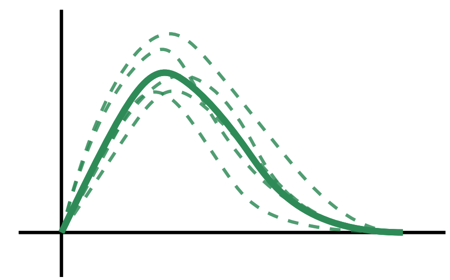
Two-sample Test (for pairs of graphs)

$X_1, \ldots, X_N \sim P$ a sample
$P_N = N^{-1} \sum_i \delta_{X_i}$

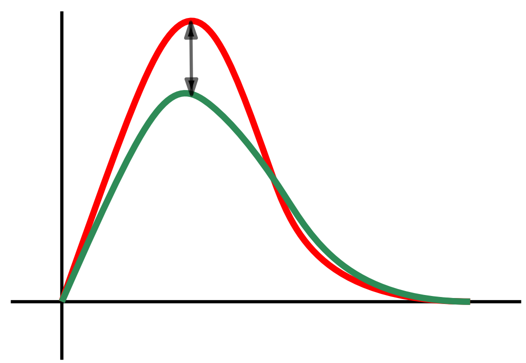
$Y_1, \ldots, Y_M \sim Q$ a sample
$Q_M = M^{-1} \sum_i \delta_{Y_i}$



$$\mathcal{H}_0 : P = Q \quad \text{or} \quad \mathcal{H}_1 : P \neq Q$$



Idea : compute $T_{N,M} = \frac{\sqrt{NM}}{\sqrt{N+M}} \| P_N D. - Q_M D. \|_\infty$.



$\boxed{\mathbb{P}_{\mathcal{H}_0}\left(T_{N,M} > c\right) \leq \alpha}$

- reject $\mathcal{H}_0$, if $T_{N,M} > c$
- retain $\mathcal{H}_0$, otherwise

Two-sample Test (for pairs of graphs)

Idea : estimate c by resampling.

$$T_{N,M} = \frac{\sqrt{NM}}{\sqrt{N+M}} \|P_N D. - Q_M D.\|_\infty$$

$$\hat{T}_{N,M} := \frac{\sqrt{NM}}{\sqrt{N+M}} \|\hat{P}_N D. - \hat{Q}_M D.\|_\infty$$

resampled from

$Z = (X_1, \ldots, X_N, Y_1, \ldots, Y_M)$

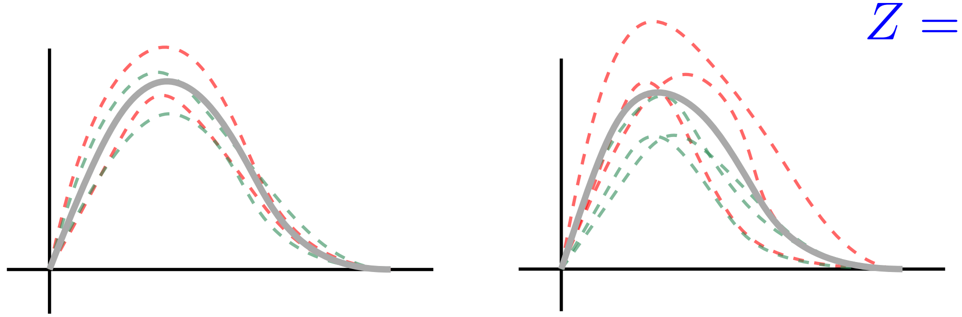
Two-sample Test (for pairs of graphs)

Idea : estimate c by resampling.

$$T_{N,M} = \frac{\sqrt{NM}}{\sqrt{N+M}} \|P_N D. - Q_M D.\|_\infty$$

$$\hat{T}_{N,M} := \frac{\sqrt{NM}}{\sqrt{N+M}} \|\hat{P}_N D. - \hat{Q}_M D.\|_\infty$$

resampled from
$Z = (X_1, \ldots, X_N, Y_1, \ldots, Y_M)$



(Sous $\mathcal{H}_0$) $\quad \hat{T}_{N,M} \,\Big|\, Z \xrightarrow{(d)} \|\mathbb{G}.\|_\infty \xleftarrow{(d)} T_{N,M}$

$\tilde{c}$: estimation of the α-upper quantile of $\hat{T}_{N,M}|Z$

$$\lim_{N,M\to\infty} \mathbb{P}_{\mathcal{H}_0}\left(T_{N,M} \geq \tilde{c}\right) \leq \alpha$$

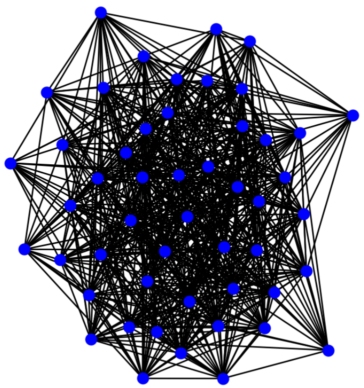
if $PD. \neq QD.$, $\lim_{N,M\to\infty} \mathbb{P}_{\mathcal{H}_1}\left(T_{N,M} \geq \tilde{c}\right) = 1$

Simulations : Stochastic Models

Erdös-Renyi (ER)

$n = 50$

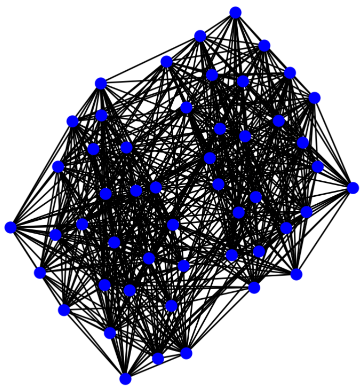
$p = 0.5$



Stochastic Block Model (SBM)

$n_1 = n_2 = 25$

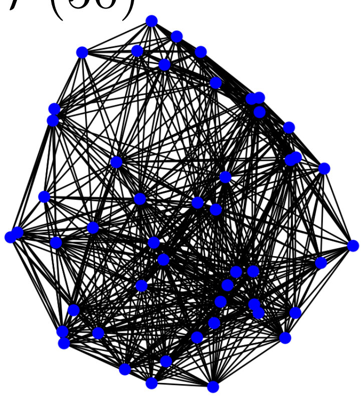
$p = \begin{pmatrix} 0.75, & 0.25 \\ 0.25 & 0.75 \end{pmatrix}$



Geometric (Disk)

$n = 50$ or $\mathcal{P}(50)$

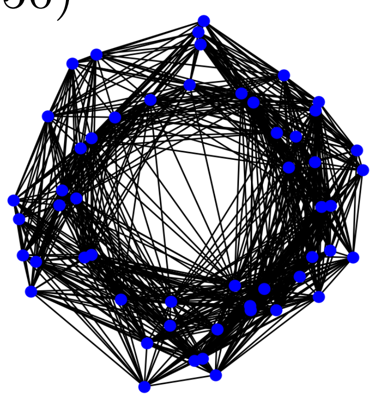
$p = 0.5$



Geometric (Annulus)

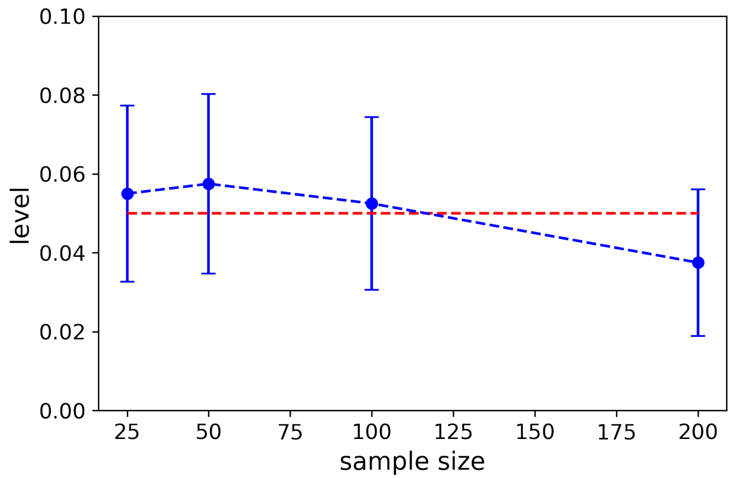
$n = 50$ or $\mathcal{P}(50)$

$p = 0.5$

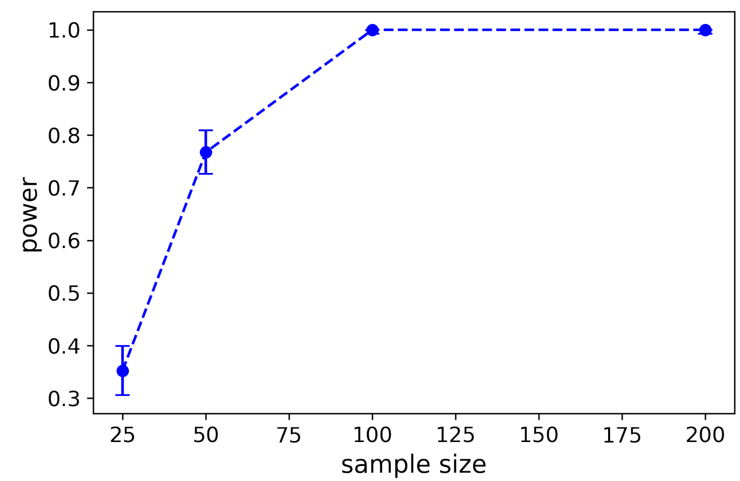


Simulations : Two-sample Tests

HKD



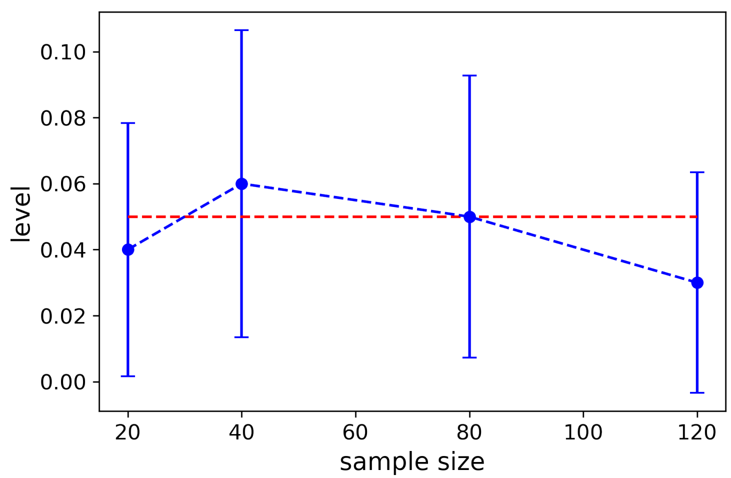
ER-ER vs ER-ER



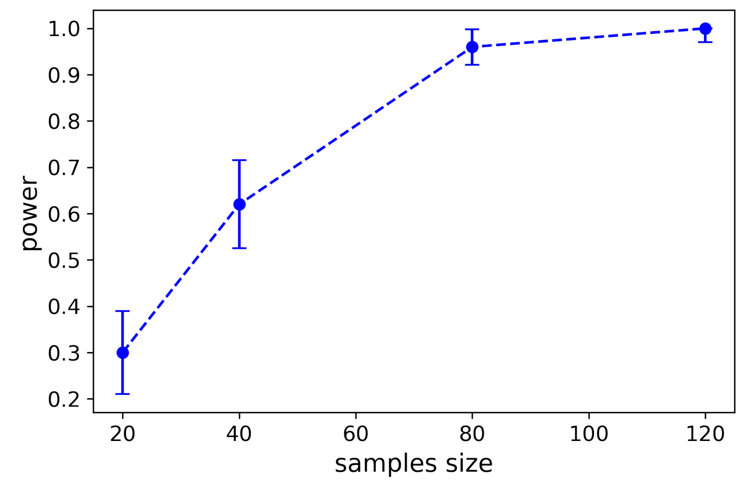
ER-ER vs ER-SBM

Level 95%, bootstrap sample size : 1000, number of tests : 400

HPD



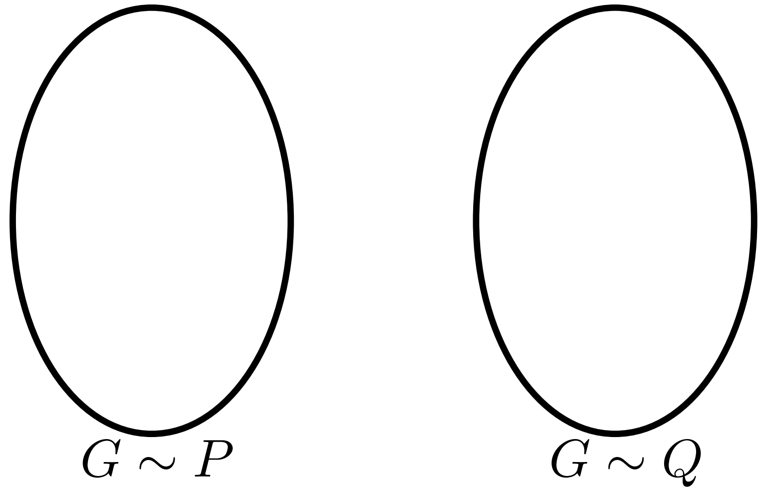
Disk-Disk vs Disk-Disk



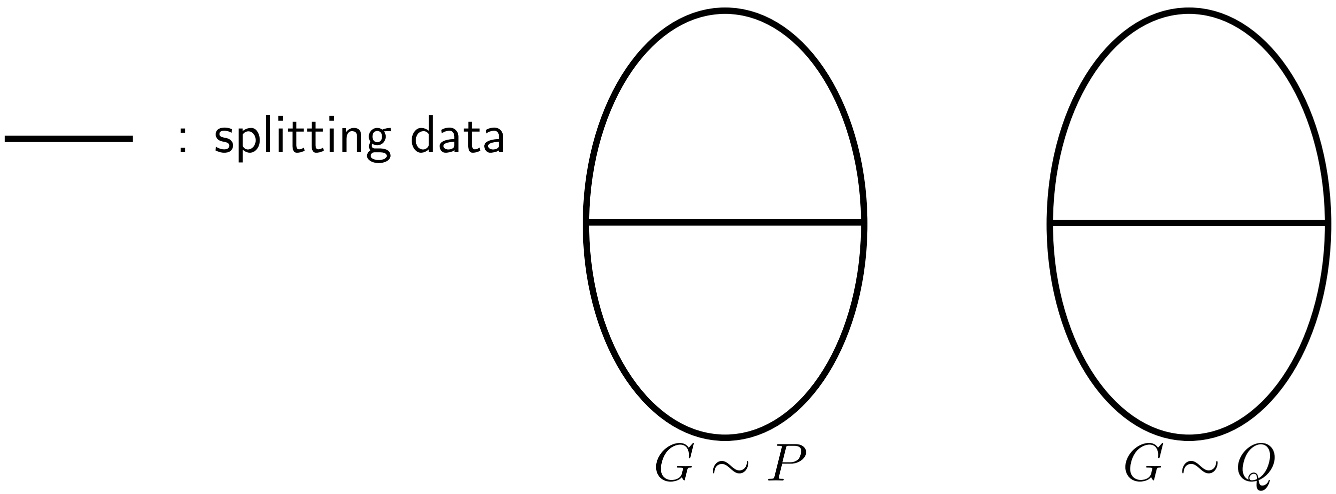
Disk-Disk vs Disk-Annulus

Level 95%, bootstrap sample size : 1000, number of tests : 100

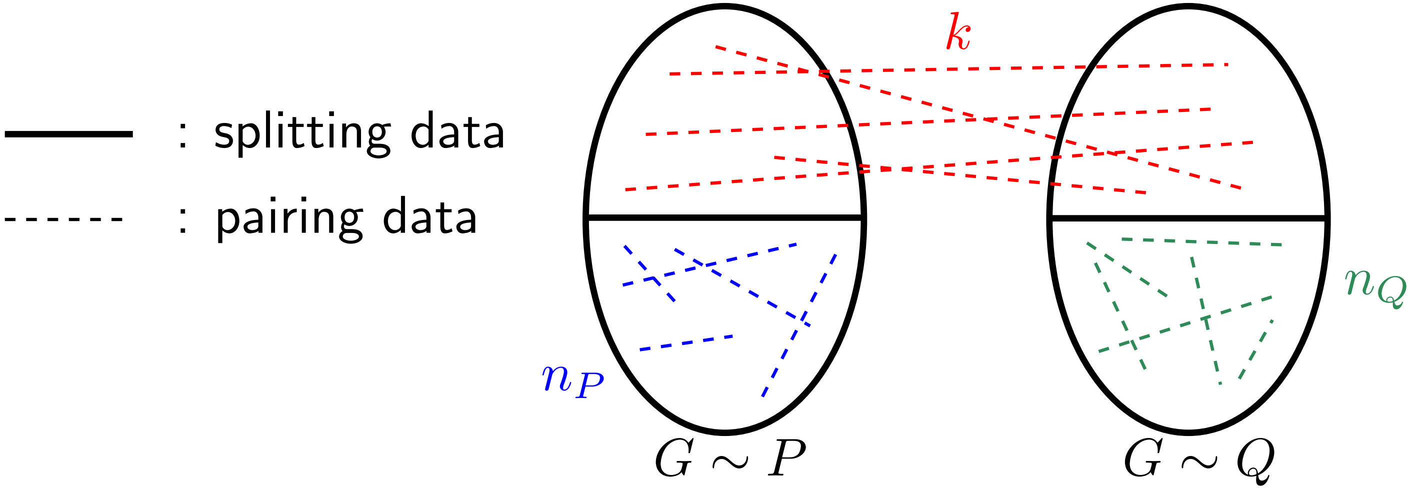
Two-sample Test (individual graphs)



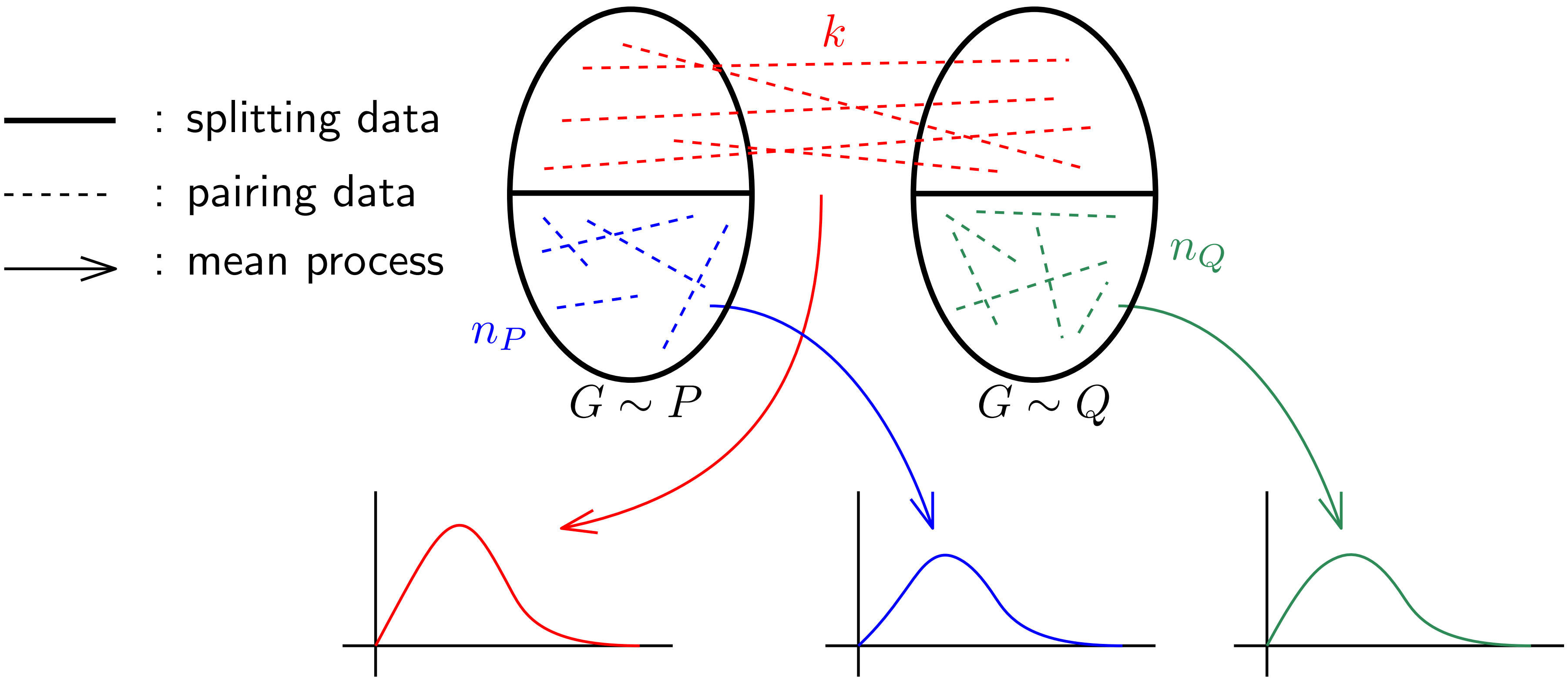
Two-sample Test (individual graphs)



Two-sample Test (individual graphs)



Two-sample Test (individual graphs)

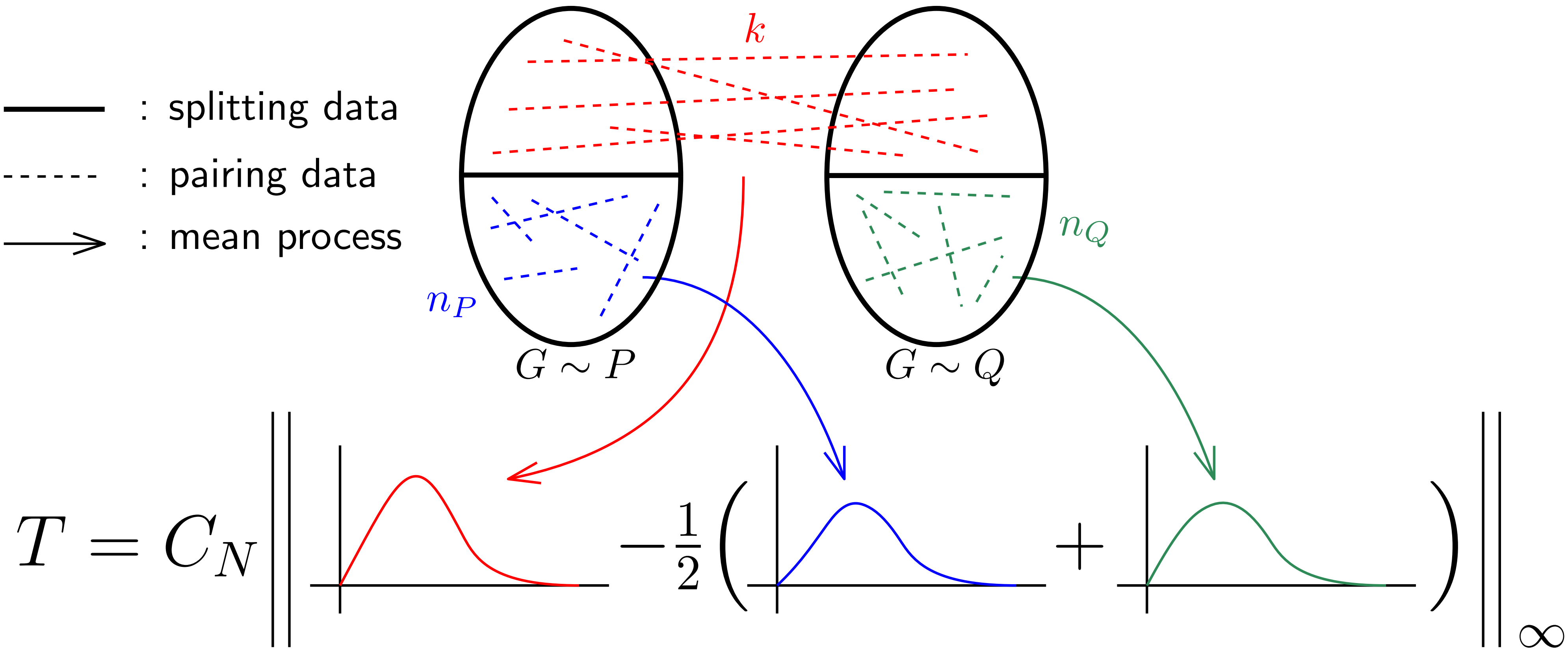


Two-sample Test (individual graphs)

: splitting data

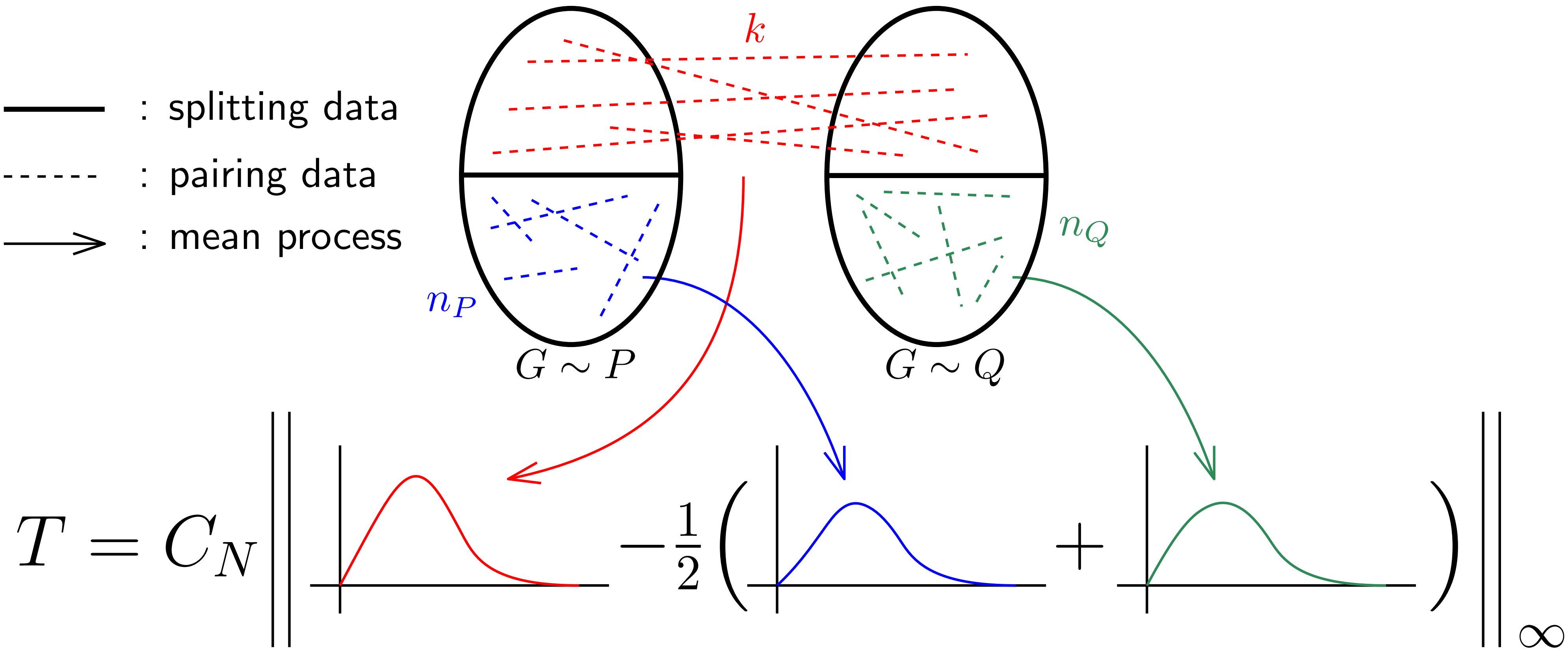
: pairing data

: mean process



with $C_N = \sqrt{\frac{4kn_Pn_Q}{kn_P + kn_Q + 4n_Pn_Q}}$

Two-sample Test (individual graphs)



with $C_N = \sqrt{\frac{4kn_Pn_Q}{kn_P + kn_Q + 4n_Pn_Q}}$

By bootstrap : we find $\tilde{c}$ such that if $P = Q$

$$\lim_{N\to\infty} \mathbb{P}\left(T > \tilde{c}\right) \leq \alpha.$$

- If $T > \tilde{c}$, conclude $P \neq Q$.
- If $T \leq \tilde{c}$, conclude $P = Q$.

Detecting distribution shift.

Distribution shift

A supervized ML algorithm in development :

- Data : train set + test set
- From the train set,
  learn a function (e.g. classifier)
- From the test set,
  evaluate the trained function.

Will the trained function perform well once deployed
in the "**real world**" ?

Distribution shift

A supervized ML algorithm in development :

- Data : train set + test set
- From the train set,
  learn a function (e.g. classifier)
- From the test set,
  evaluate the trained function.

Will the trained function perform well once deployed
in the "**real world**"?

- Enhance "real world" performances.
  (Robustness, data augmentation, ...)
- Detect a potential shift of distribution.

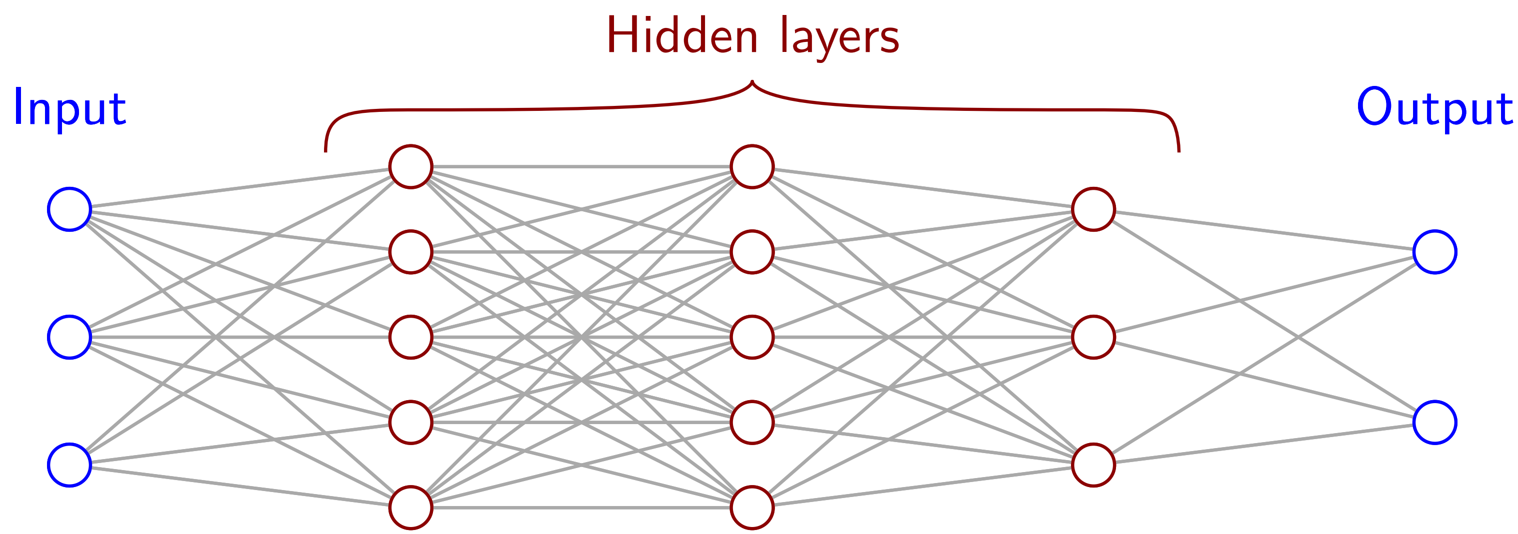
$$\left.\begin{array}{r}\text{development data} \sim P \\ \text{"real world" data} \sim Q\end{array}\right\}$$

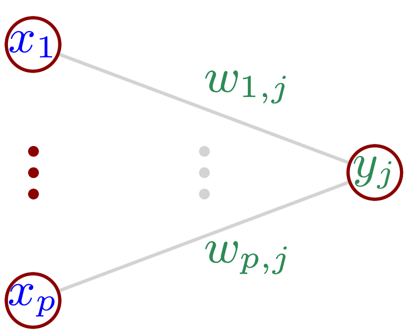
- If $P = Q$, ok! ✓
- Else, $P \neq Q$, risk of poor behavior!

Can we detect if the distribution has shifted?

Even better: can we detect if the algorithm is performing poorly?

With Neural Networks

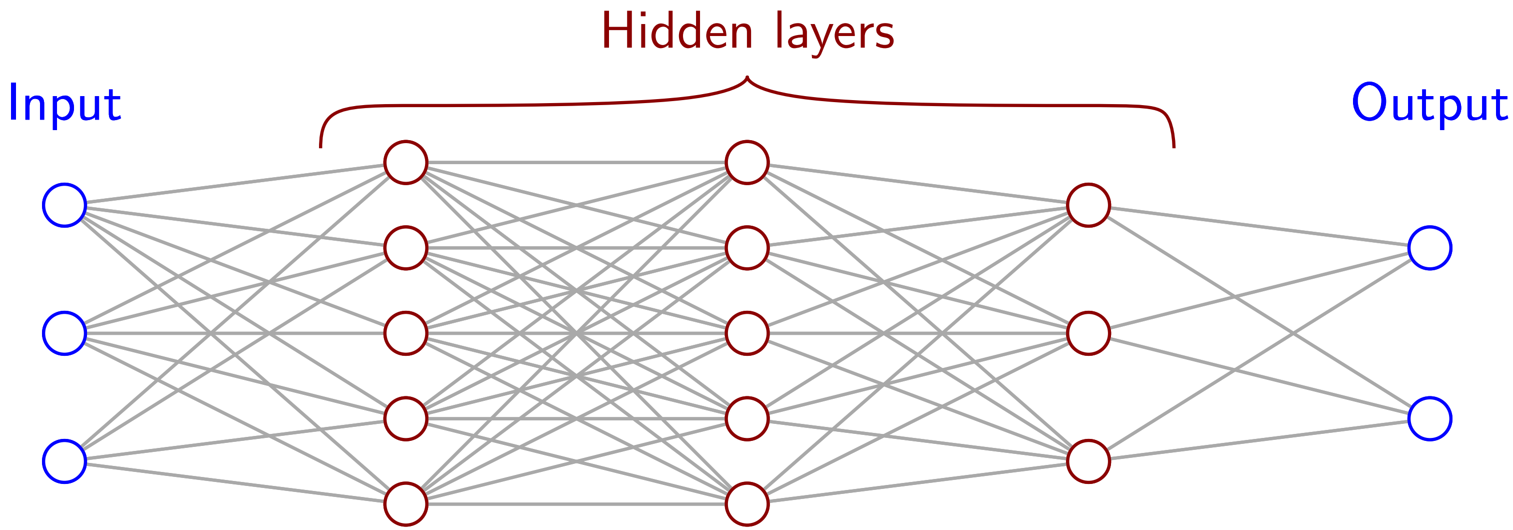


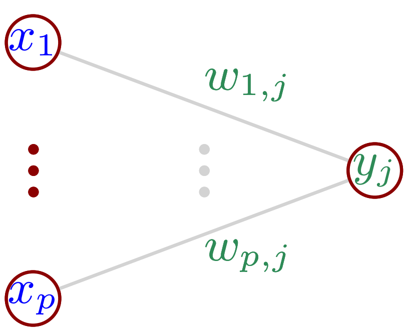


$$\boxed{y_j = \sigma({w_j}^T x + b_j)}$$

$$\begin{cases} \sigma : \mathbb{R} \to \mathbb{R} : \text{ activation function (fixed)} \\ w_j \in \mathbb{R}^p, \text{ NN weights (trained)} \\ b_j \in \mathbb{R}, \text{ bias. (trained)} \end{cases}$$

With Neural Networks





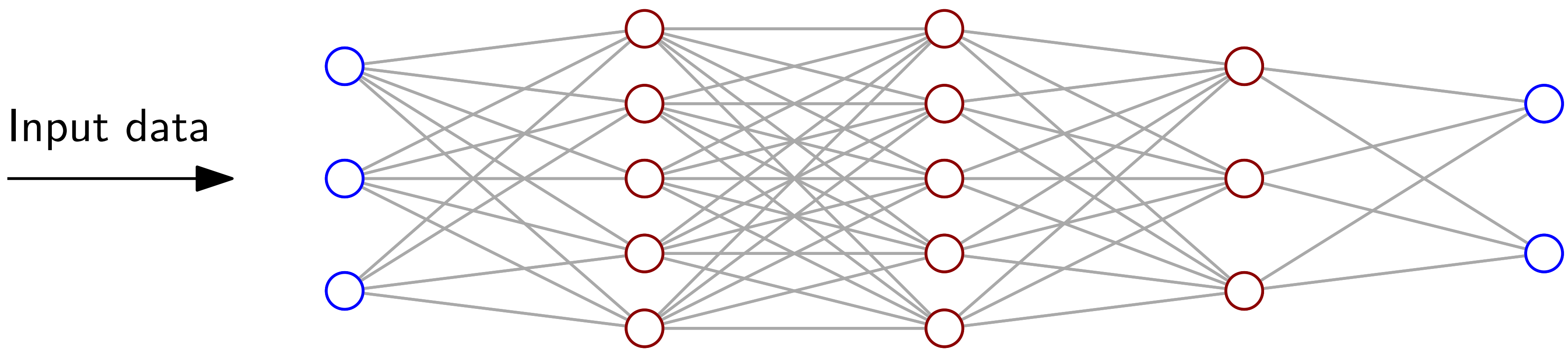
$$\boxed{y_j = \sigma({w_j}^T x + b_j)} \quad \begin{cases} \sigma : \mathbb{R} \to \mathbb{R} : \text{ activation function (fixed)} \\ w_j \in \mathbb{R}^p, \text{ NN weights (trained)} \\ b_j \in \mathbb{R}, \text{ bias. (trained)} \end{cases}$$

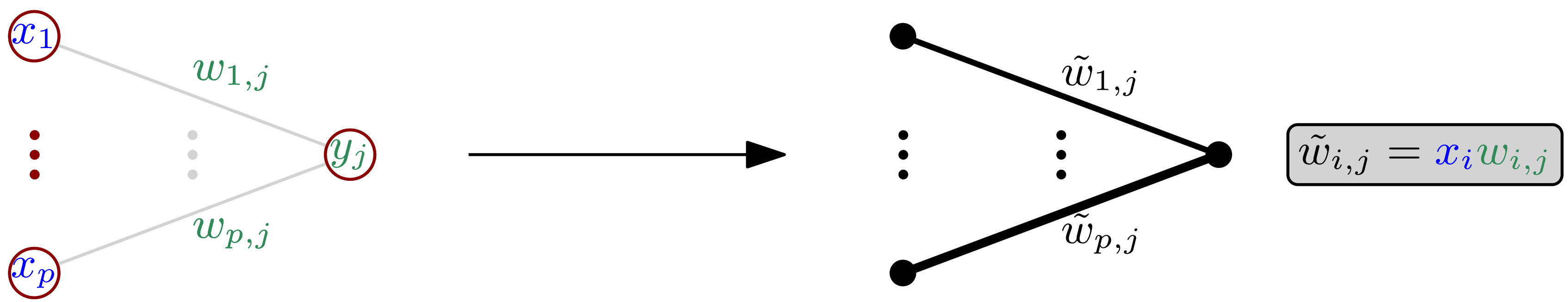
Can we use the underlying graph structure to detect distribution shifts ?

Activation Graphs

Consider :

- a trained neural network,
- one data instance.



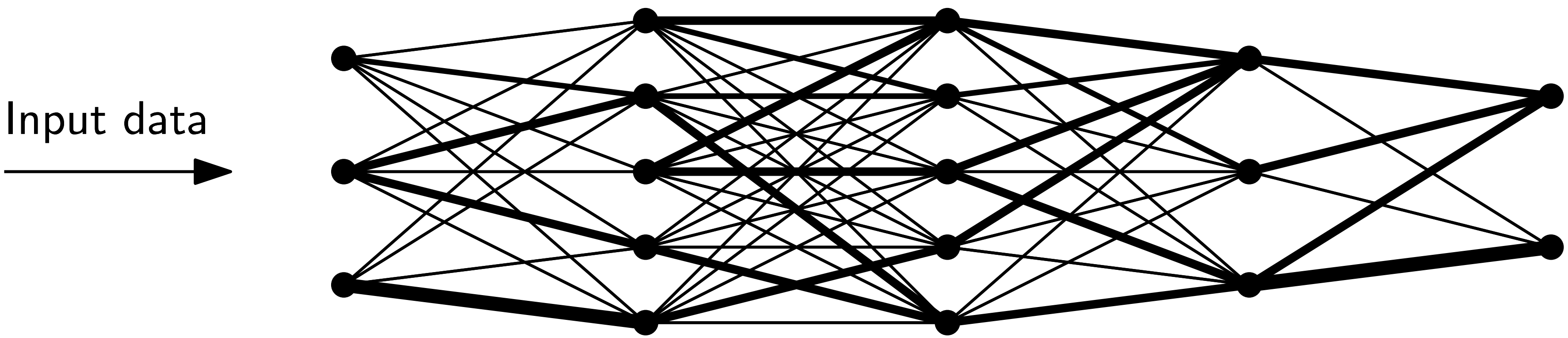


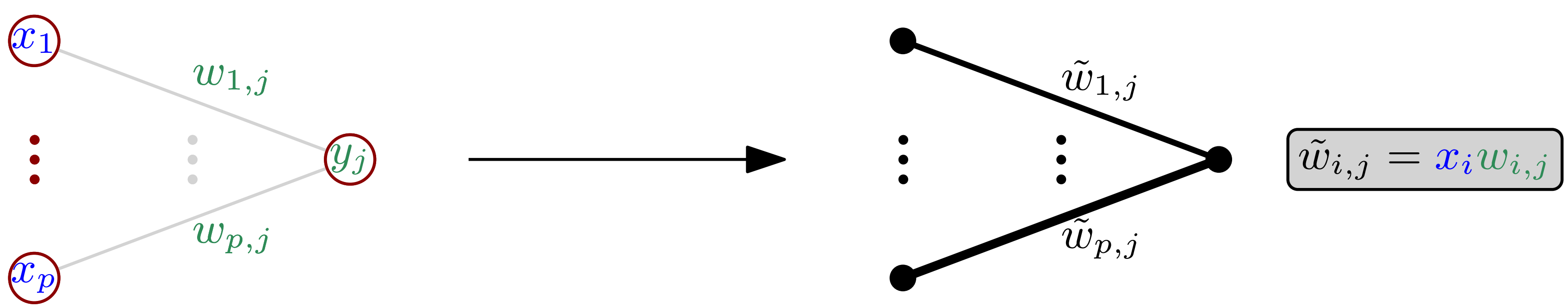
- weighted graph
- same vertex set
- associated to each data instance
- characterize the processing of the data

Activation Graphs

Consider :

- a trained neural network,
- one data instance.





- weighted graph
- same vertex set
- associated to each data instance
- characterize the processing of the data

MNIST

The corruptions :

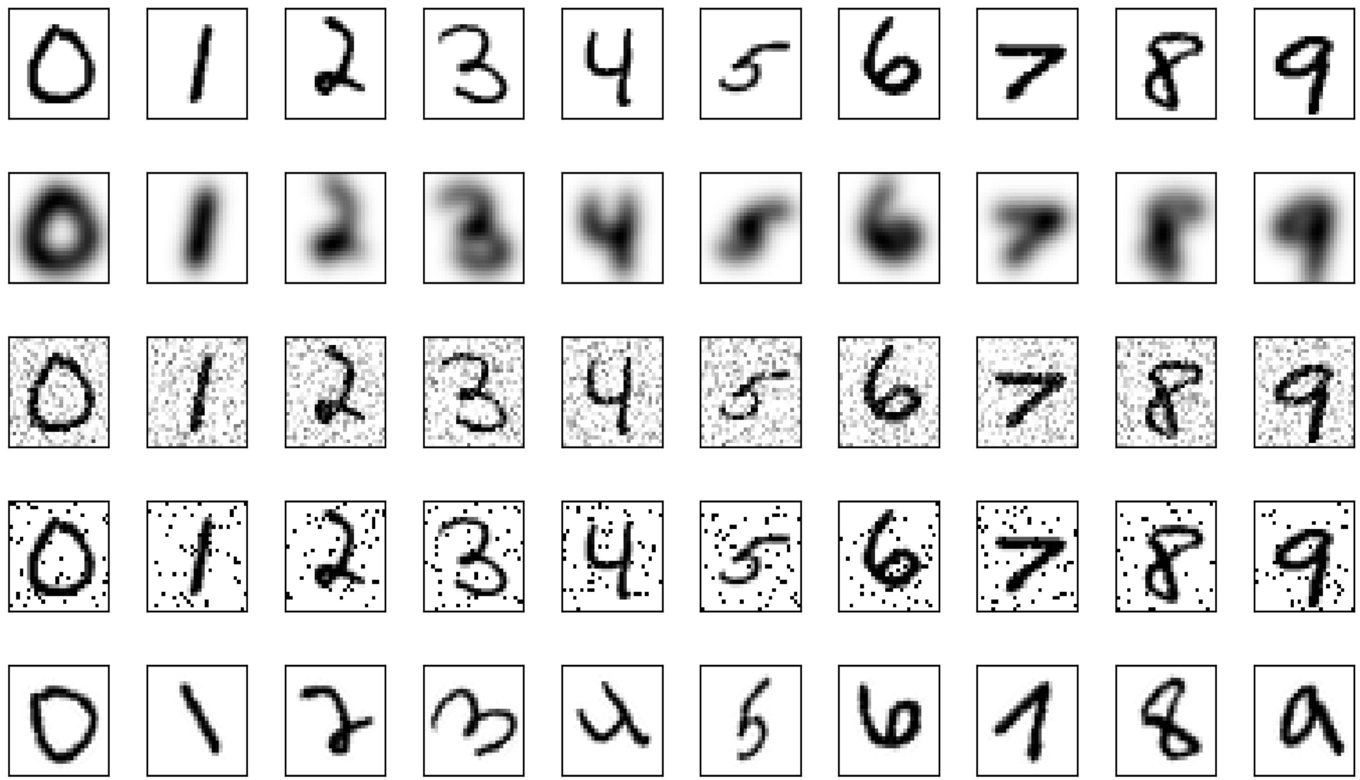
Original

Gaussian blur

Gaussian noise

Random black pixels

Rotation

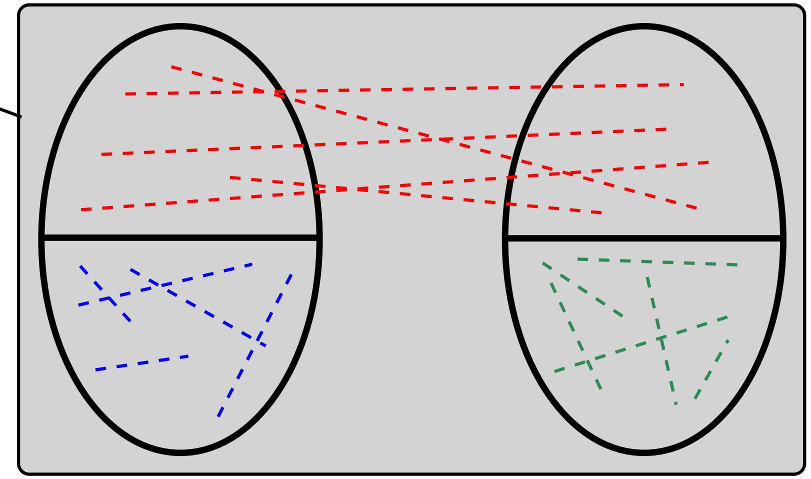


The methods

- HD
- HDy (pairing by label)
- HDmt (repeat pairing / multiple testing)
- hammond (use $\max_t D_t(G, G')$)
- BBSD [LWS18]

 [LWS18] Detecting and Correcting for Label Shift with Black Box Predictors, Lipton *et al*, 2018.

The methods

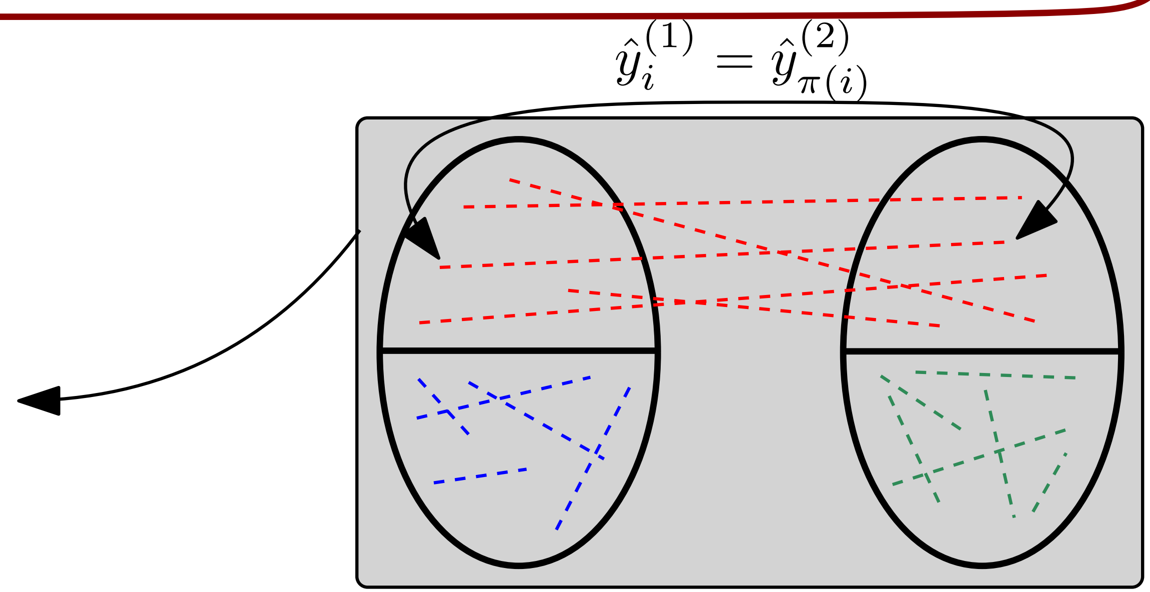


- HD
- HDy (pairing by label)
- HDmt (repeat pairing / multiple testing)
- hammond (use $\max_t D_t(G, G')$)
- BBSD [LWS18]

 [LWS18] Detecting and Correcting for Label Shift with Black Box Predictors, Lipton *et al*, 2018.

The methods

- HD
- HDy (pairing by label)

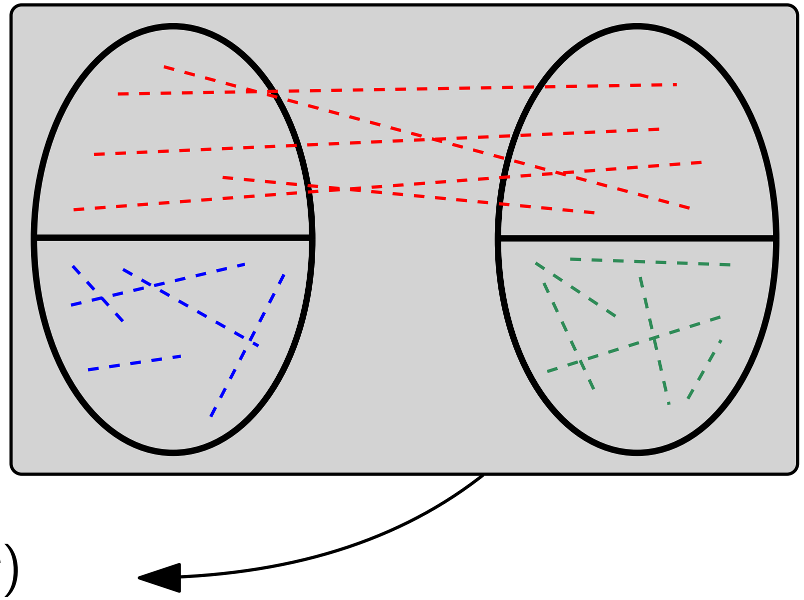


- HDmt (repeat pairing / multiple testing)
- hammond (use $\max_t D_t(G, G')$)
- BBSD [LWS18]

 [LWS18] Detecting and Correcting for Label Shift with Black Box Predictors, Lipton *et al*, 2018.

The methods

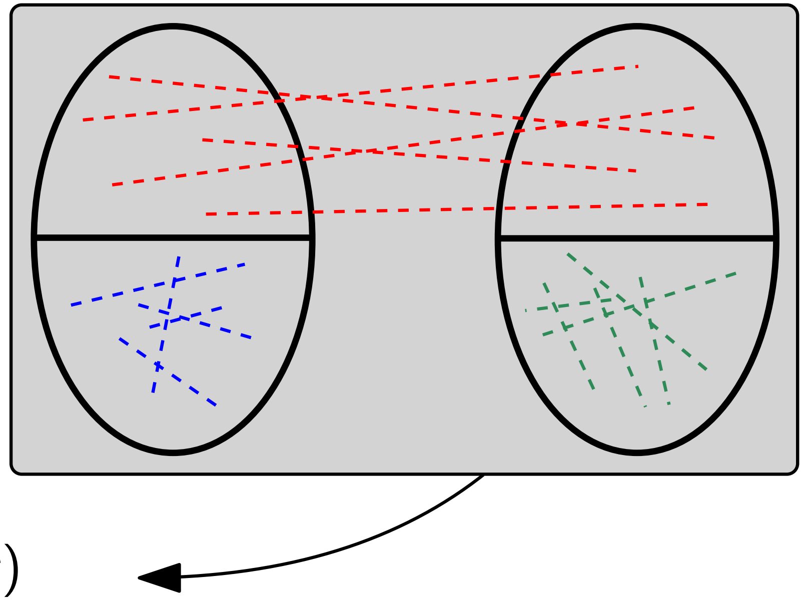
- HD
- HDy (pairing by label)
- HDmt (repeat pairing / multiple testing)
- hammond (use $\max_t D_t(G, G')$)
- BBSD [LWS18]



 [LWS18] Detecting and Correcting for Label Shift with Black Box Predictors, Lipton *et al*, 2018.

The methods

- HD
- HDy (pairing by label)



- HDmt (repeat pairing / multiple testing)
- hammond (use $\max_t D_t(G, G')$)
- BBSD [LWS18]

 [LWS18] Detecting and Correcting for Label Shift with Black Box Predictors, Lipton *et al*, 2018.

The methods

- HD

Compare the means of $D_\cdot(G_i^k, G_{\pi(i)}^{k'})$ using $\|\cdot\|_\infty$, and the functional CLT.

- HDy (pairing by label)

- HDmt (repeat pairing / multiple testing)

- hammond (use $\max_t D_t(G, G')$)

Compare the means of $\|D_\cdot(G_i^k, G_{\pi(i)}^{k'})\|_\infty$ using the standard CLT.

- BBSD [LWS18]

[LWS18] Detecting and Correcting for Label Shift with Black Box Predictors, Lipton *et al*, 2018.

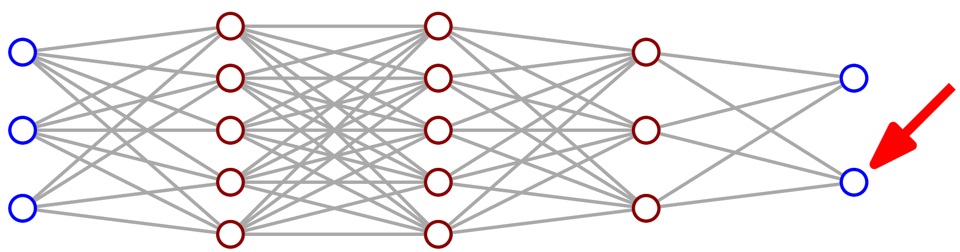
The methods

- HD

- HDy (pairing by label)

- HDmt (repeat pairing / multiple testing)

- hammond (use $\max_t D_t(G, G')$)



- BBSD [LWS18]

For each output neuron :

  For both samples, extract the output neuron values.

  Perform a Kolmogorov-Smirnov two-sample test.

Combine tests with Bonferroni procedure (multiple tests).

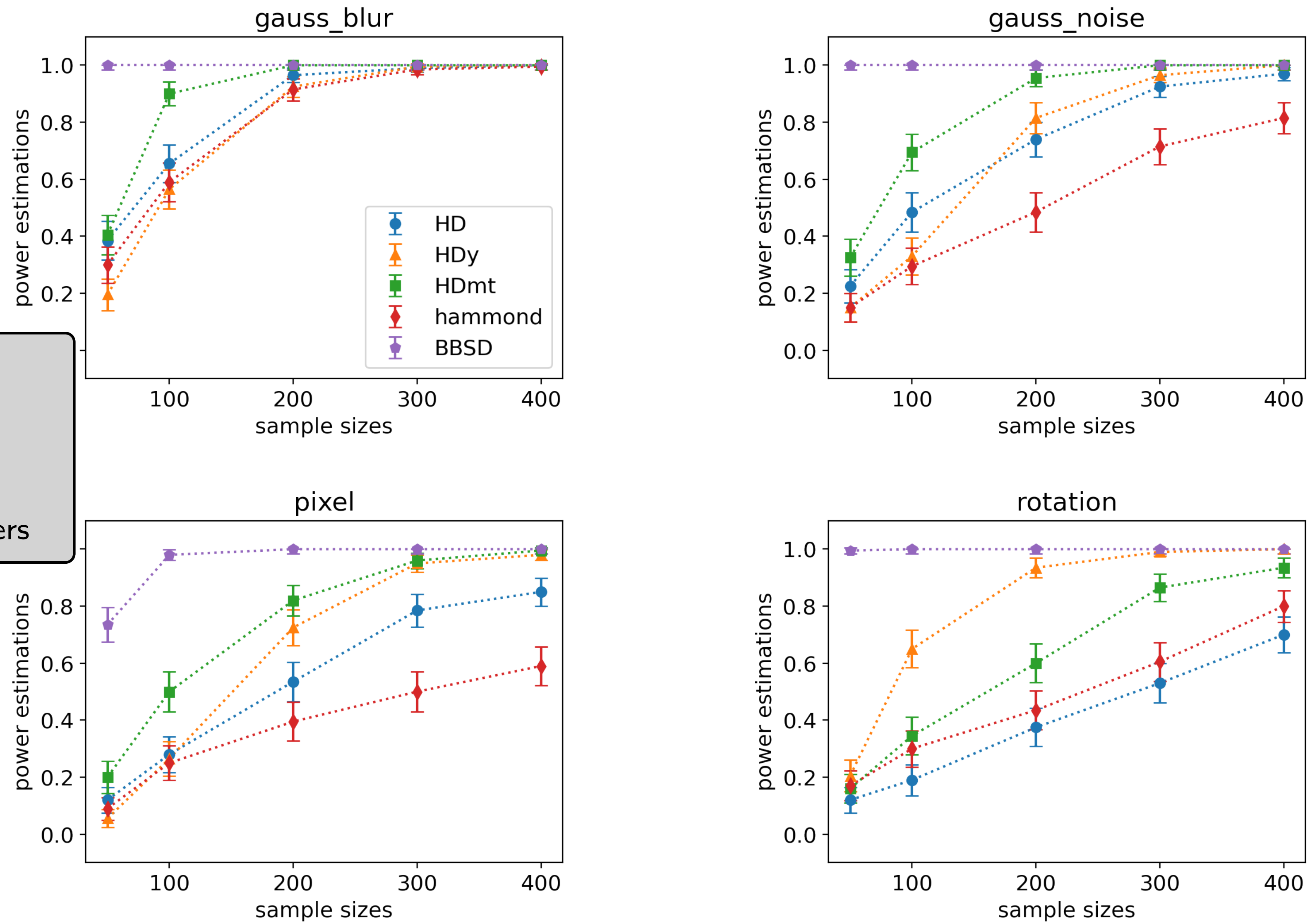
 [LWS18] Detecting and Correcting for Label Shift with Black Box Predictors, Lipton *et al*, 2018.

Results

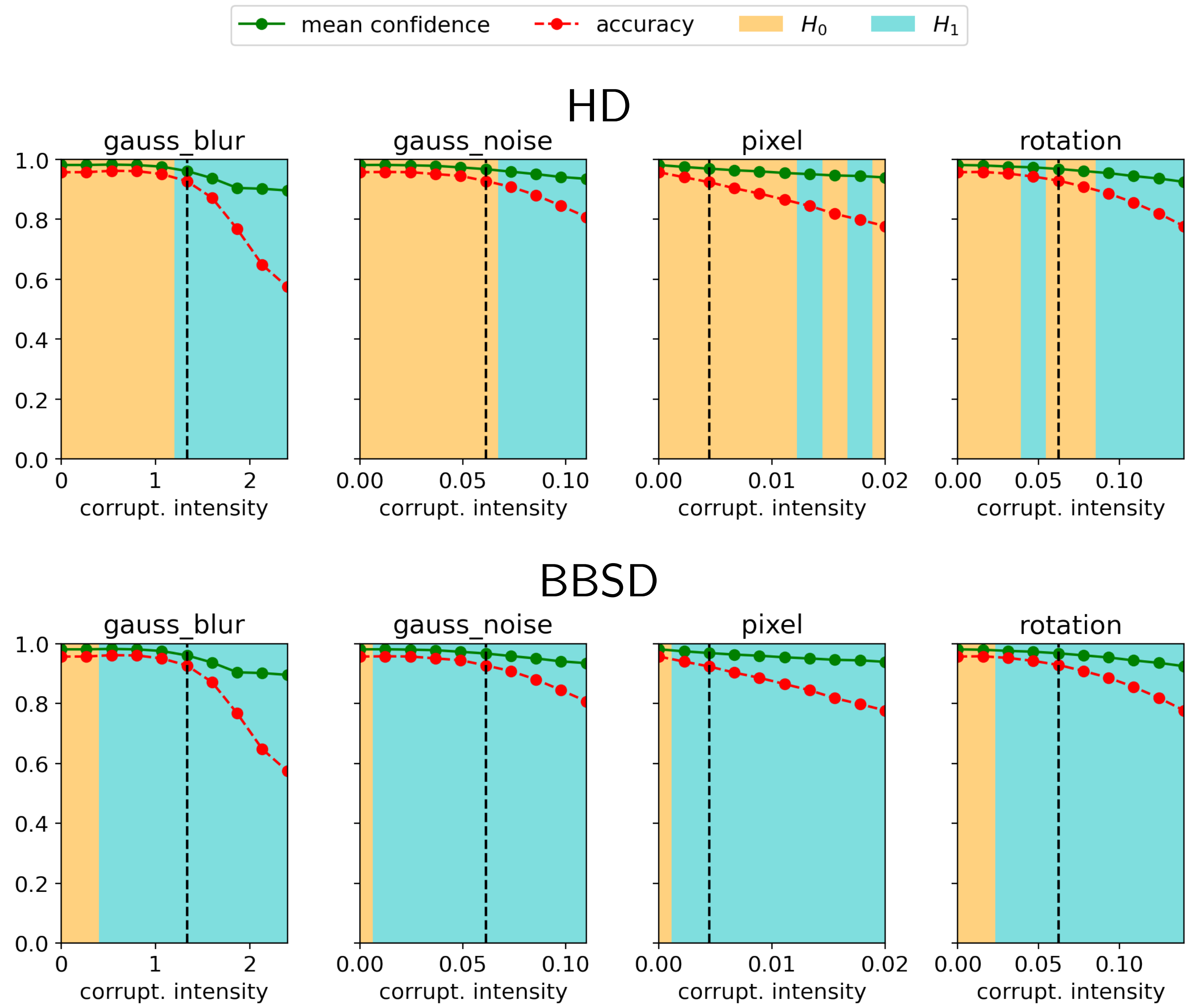
<u>Neural Network :</u>
Dense layers, 3 hidden layers (16 neurons).
Accuracy : $\simeq 0.96$

Repeat 200x $\begin{cases} \text{1st sample : original} \\ \text{2nd sample : corrupted} \\ \text{Compute the test.} \end{cases}$

- HD $\geq$ hammond
- HDy $\geq$ HD
- HDmt $\geq$ HD
- BBSD $\geq$ all others



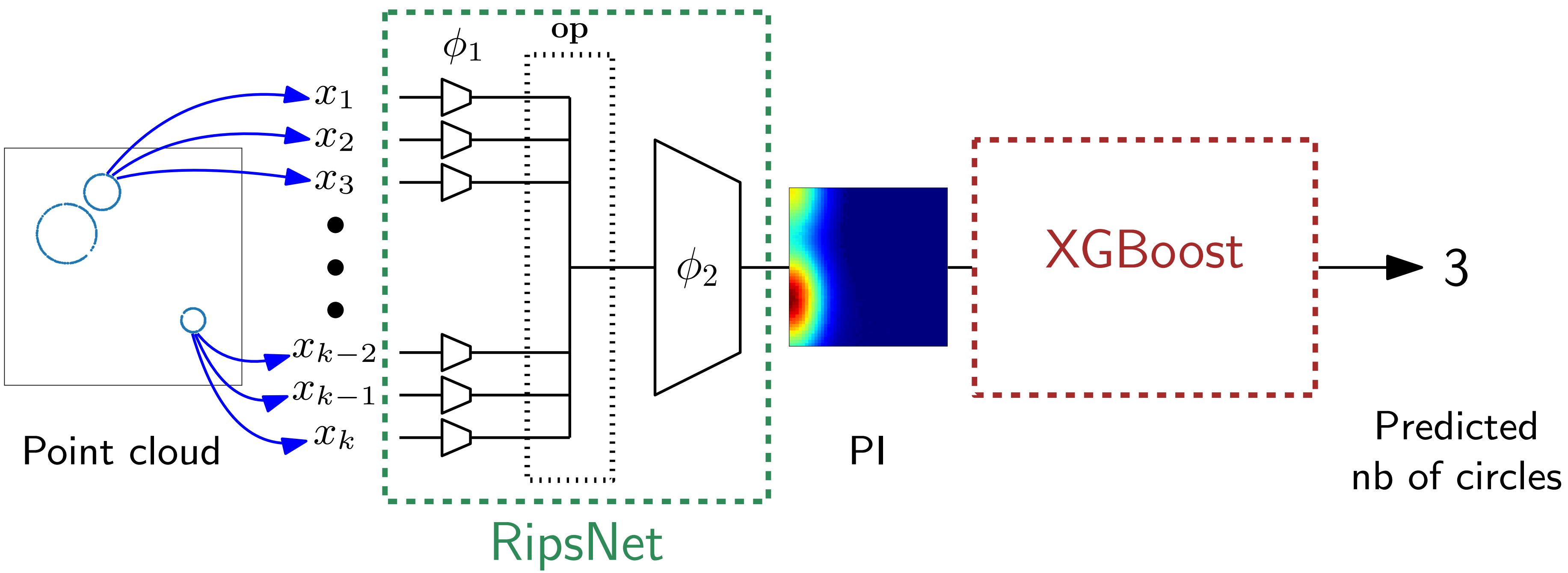
Results (full data)



50% increase of the error rate

$$\textbf{Ripsnet}: X = \{x_1, \ldots, x_k\} \mapsto \phi_2\left(\mathbf{op}\left(\{\phi_1(x_i)\}_{1\le i\le k}\right)\right),$$

$\phi_1 : \mathbb{R}^d \to \mathbb{R}^{d'}$	dense neural network
$\phi_2 : \mathbb{R}^{d'} \to \mathbb{R}^K$	dense neural network
$\mathbf{op}$:	permutation invariant operator (*i.e.*, mean)



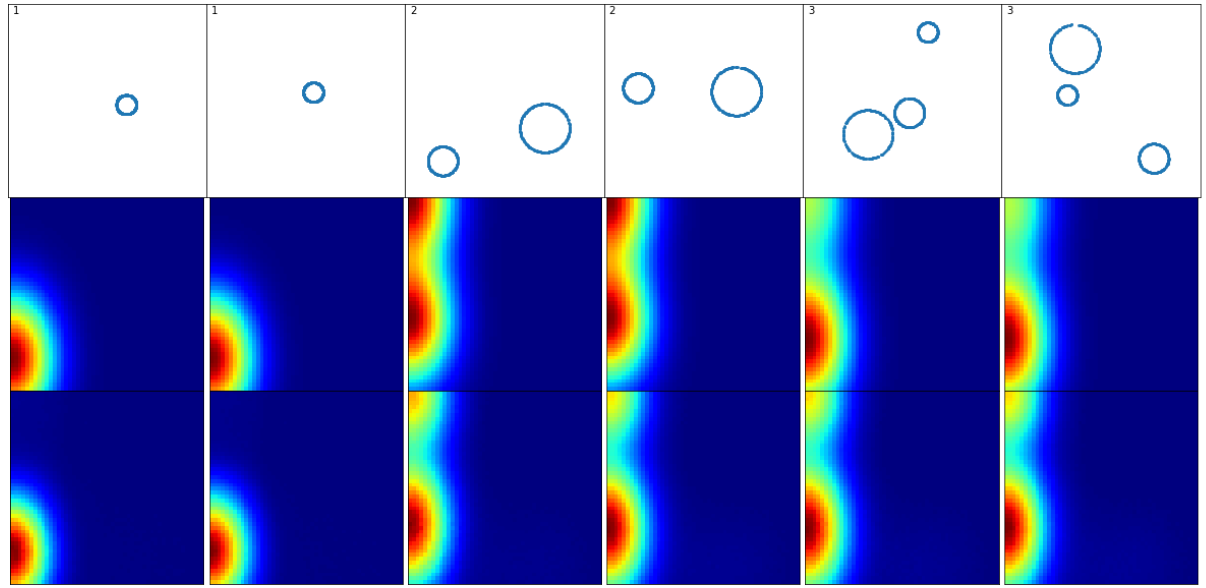
[dSHC+22] RipsNet: a general architecture for fast and robust estimation of the persistent homology of point clouds, De Surrel *et al*, 2022.

The data

Clean point clouds

Exact PIs

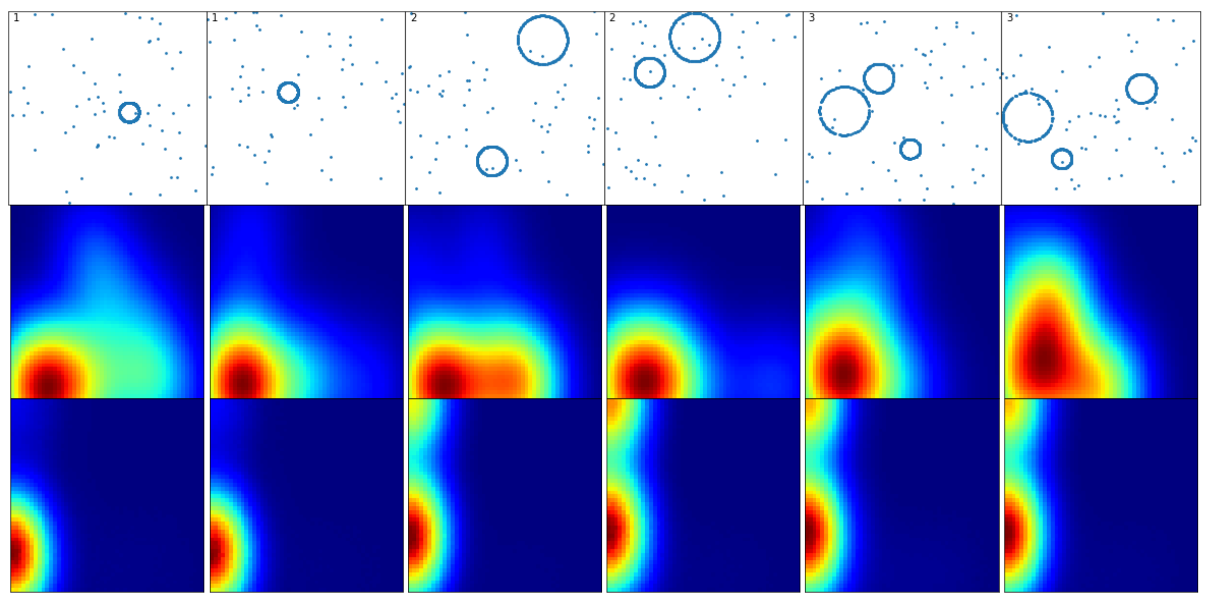
Estimated PIs



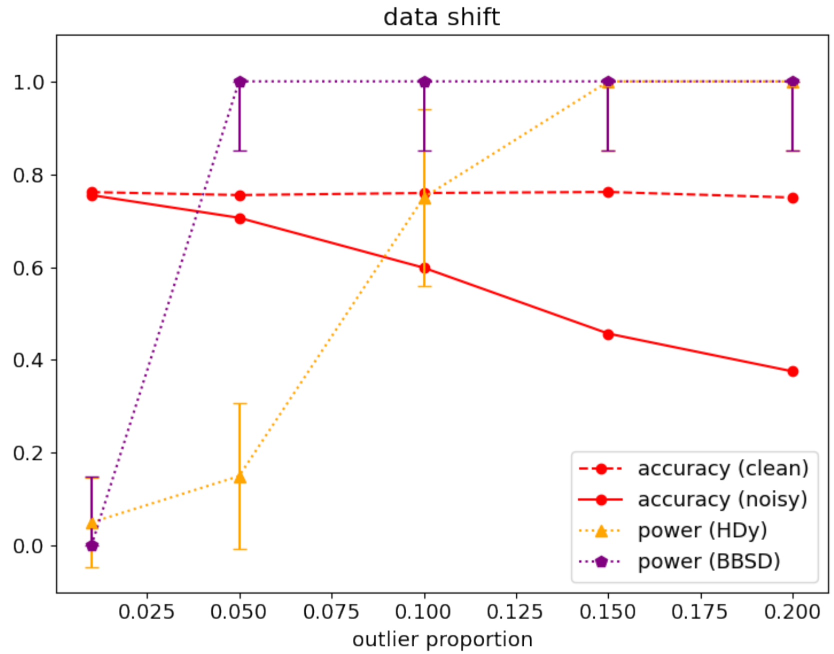
Noisy point clouds

Exact PIs

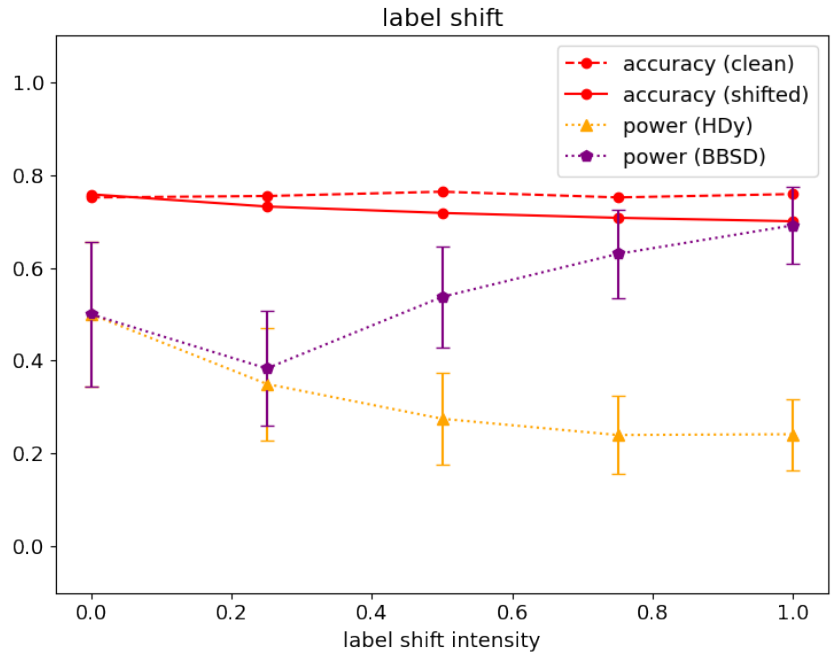
Estimated PIs



Results



data shift
1.0
0.8
0.6
0.4
0.2
0.0
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
outlier proportion
accuracy (clean)
accuracy (noisy)
power (HDy)
power (BBSD)



label shift
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
label shift intensity
accuracy (clean)
accuracy (shifted)
power (HDy)
power (BBSD)

Conclusions

Publication :

L., 2021, Heat diffusion distance processes: a statistically founded method to analyze graph data sets, arXiv:2109.13213 (Journal of Applied and Computational Topology).

Perspectives :

- Theoretical study of the test power (special cases)
- Interplay between graph size and sample size
- Extensions to classical learning tasks on graphs (clustering, classification, outlier detection, change-point detection for time series)
- Study of neural networks (over-fitting, over-parametrization, ...)

THANK YOU FOR YOUR ATTENTION !

Simulations : Two-sample Tests

Neyman-Pearson regime :

sample of size N

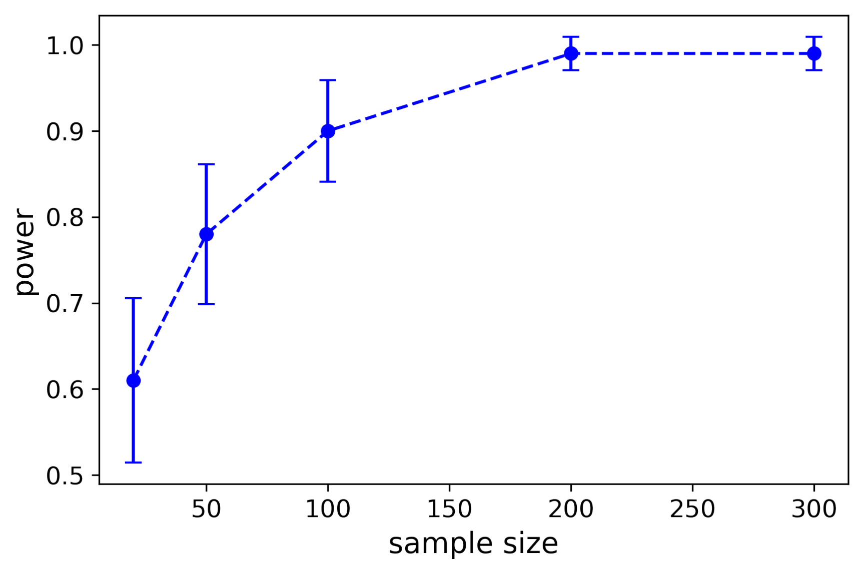
Neyman-Pearson test : $ER(p_1(N))$ vs $ER(p_2(N))$

$$|p_1(N) - p_2(N)| \gg 1/\sqrt{N}$$

$$p_1(N) \rightarrow 0.5 \leftarrow p_2(N)$$

$n = 50$

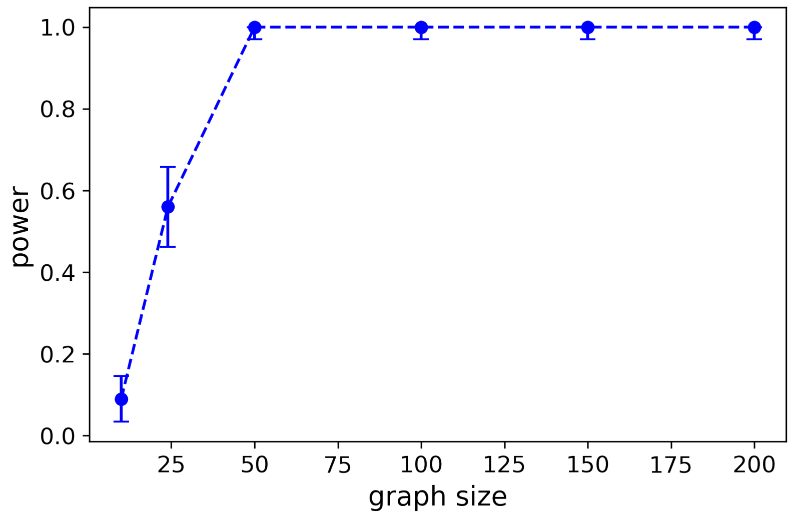
$$|p_1(N) - p_2(N)| \sim \log(N)/\sqrt{N}$$



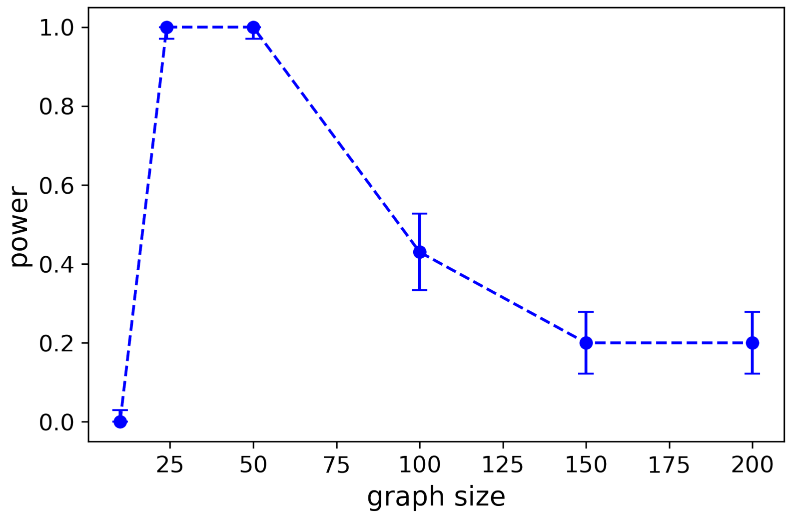
Influence of the graph sizes

ER-ER vs ER-SBM

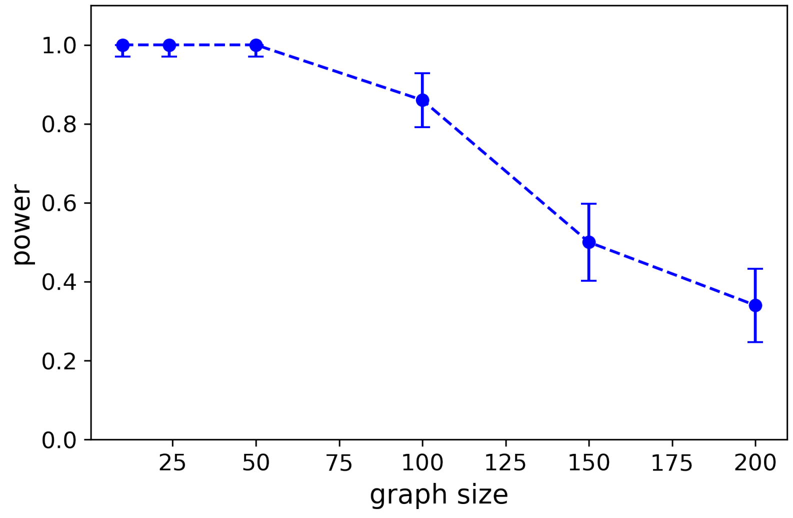
$p, q \sim c$



$p, q \sim c/n$



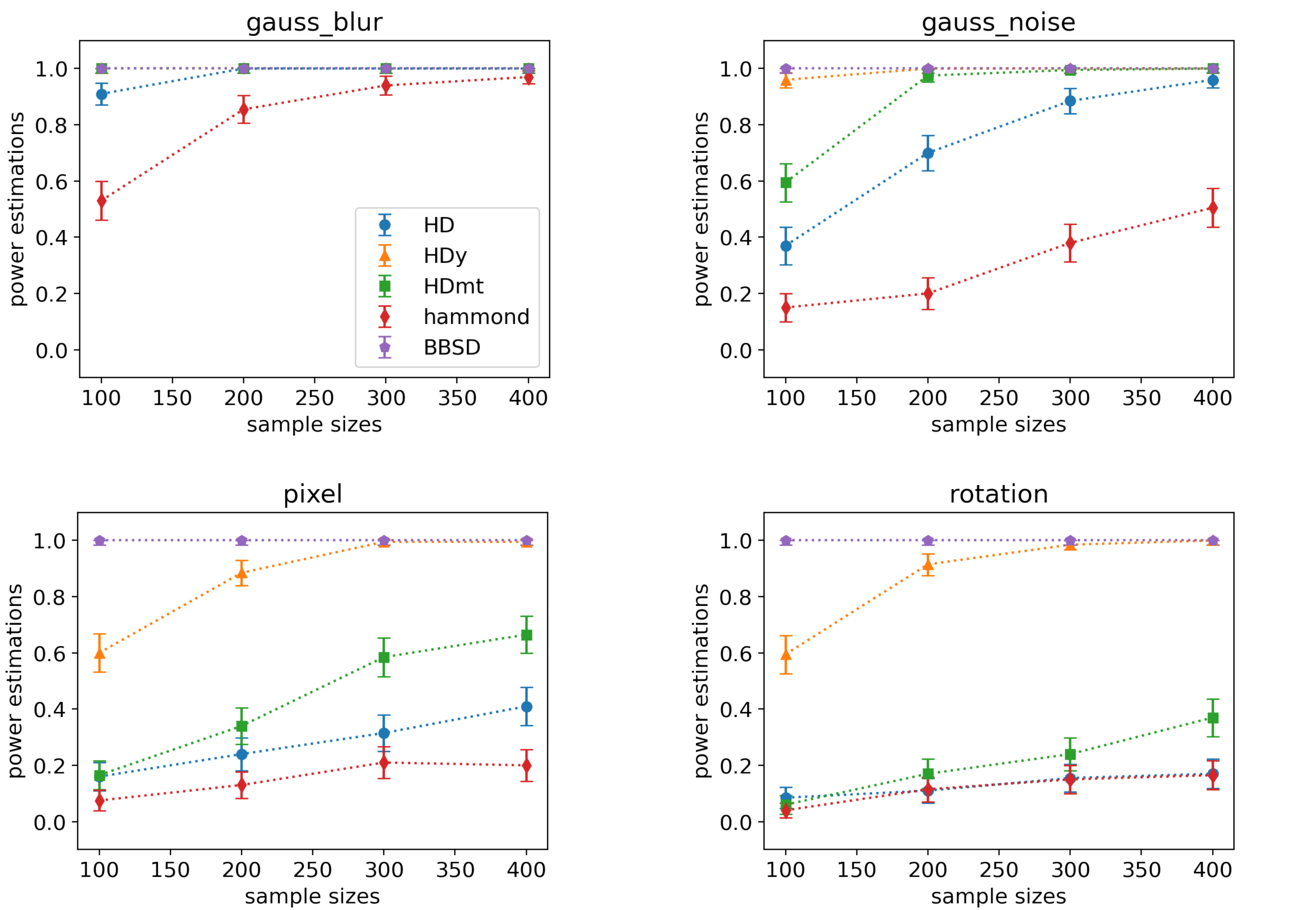
$p, q \sim c/n^{3/4}$



Results (CNN)

Neural Network :
Convolution layers + 3 dense layers (32 neurons).
Accuracy : $\simeq 0.989$

Repeat 200x



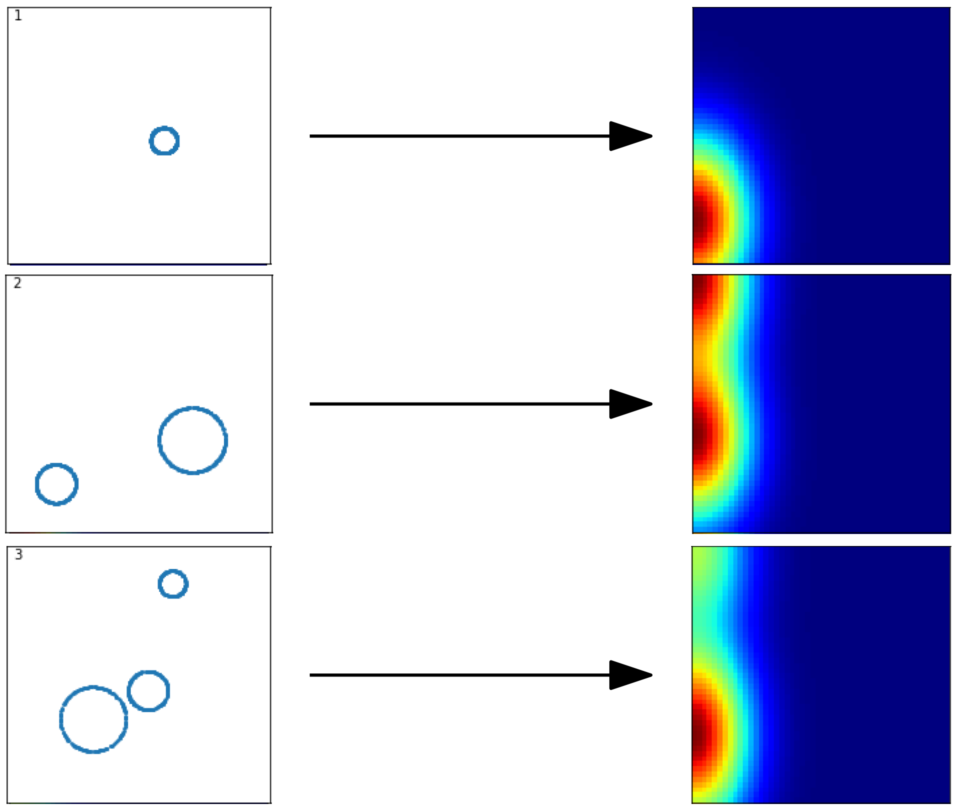
Ripsnet [dSHC+22]

$$\textbf{Ripsnet} : X = \{x_1, \ldots, x_k\} \mapsto \phi_2\left(\mathbf{op}\left(\{\phi_1(x_i)\}_{1 \leq i \leq k}\right)\right),$$

$\phi_1 : \mathbb{R}^d \to \mathbb{R}^{d'}$ dense neural network
$\phi_2 : \mathbb{R}^{d'} \to \mathbb{R}^K$ dense neural network
$\mathbf{op}$: permutation invariant operator (*i.e.*, mean)

Input : point cloud

Output : persistent image



[dSHC+22] RipsNet: a general architecture for fast and robust estimation of the persistent homology of point clouds, De Surrel *et al*, 2022.