IML Working Group, CERN 2018-10-12

Guiding New Physics Searches with Unsupervised Learning

[DS, Jacques - 1807.06038]

Andrea De Simone

andrea.desimone@sissa.it

Searches for New Physics Beyond the Standard Model
have been negative so far…

MAYBE:

1. New Physics (NP) is not accessible by LHC

new particles are too light/heavy
or interacting too weakly

2. We have not explored all the possibilities

new physics may be buried under large bkg
or hiding behind unusual signatures

“Don’t want to miss a thing” (in data)

closer look at current data
get ready for upcoming data from next run

Model-independent search

searches for specific models may be:
- time-consuming
- insensitive to unexpected/unknown processes

> New Statistical Test

Want a statistical test for NP which is:

1. **model-independent:**

   no assumption about underlying physical model to intepret data

   → more general

2. **non-parametric:**

   compare two samples as a whole (not just their means, etc.)

   → fewer assumptions, no max likelihood estim.

3. **un-binned:**

   high-dim feature space partitioned without rectangular bins

   → retain full multi-dim info of data

> Outline

1. Statistical test of dataset compatibility

- Nearest-Neighbors Two-Sample Test
- Identify Discrepancies
- Include Uncertainties

2. Applications to High-Energy Physics

1. Statistical test of dataset compatibility

- Nearest-Neighbors Two-Sample Test
- Identify Discrepancies
- Include Uncertainties

2. Applications to High-Energy Physics

> Two-sample Test

[a.k.a. "homogeneity test"]

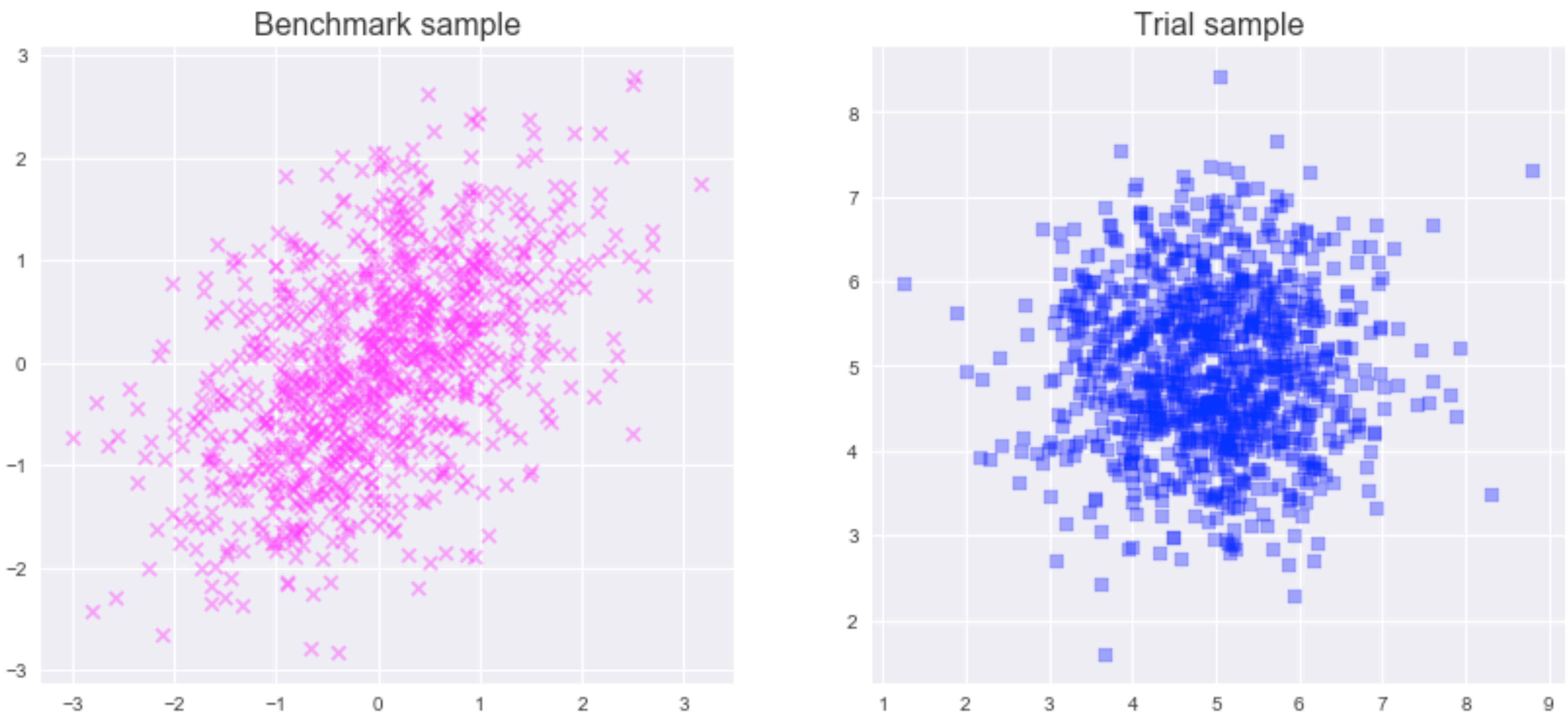
Two sets:

Trial: $\mathcal{T} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_T}\} \overset{\text{iid}}{\sim} p_T$

Benchmark: $\mathcal{B} = \{\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{N_B}\} \overset{\text{iid}}{\sim} p_B$

$\boldsymbol{x}_i, \boldsymbol{x}'_i \in \mathbb{R}^D$

probability distributions p_B, p_T unknown



e.g.: simulated SM bkg — real measured data

> Two-sample Test

Two sets:

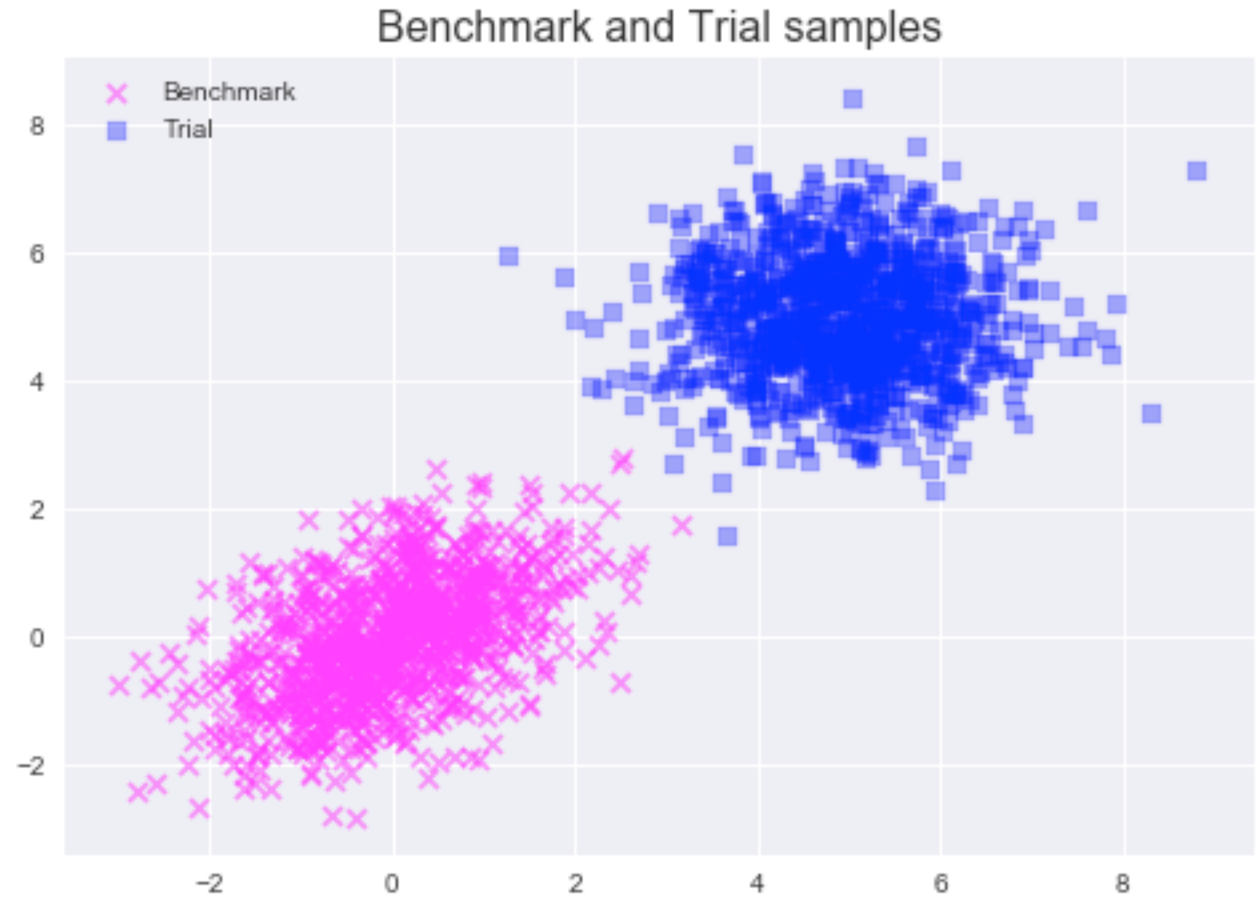
Trial: $\mathcal{T} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_T}\} \overset{\text{iid}}{\sim} p_T$

Benchmark: $\mathcal{B} = \{\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{N_B}\} \overset{\text{iid}}{\sim} p_B$

$\boldsymbol{x}_i, \boldsymbol{x}'_i \in \mathbb{R}^D$

probability distributions p_B, p_T unknown

Are *B,T* drawn from the same prob. distribution?



easy…

> Two-sample Test

Two sets:

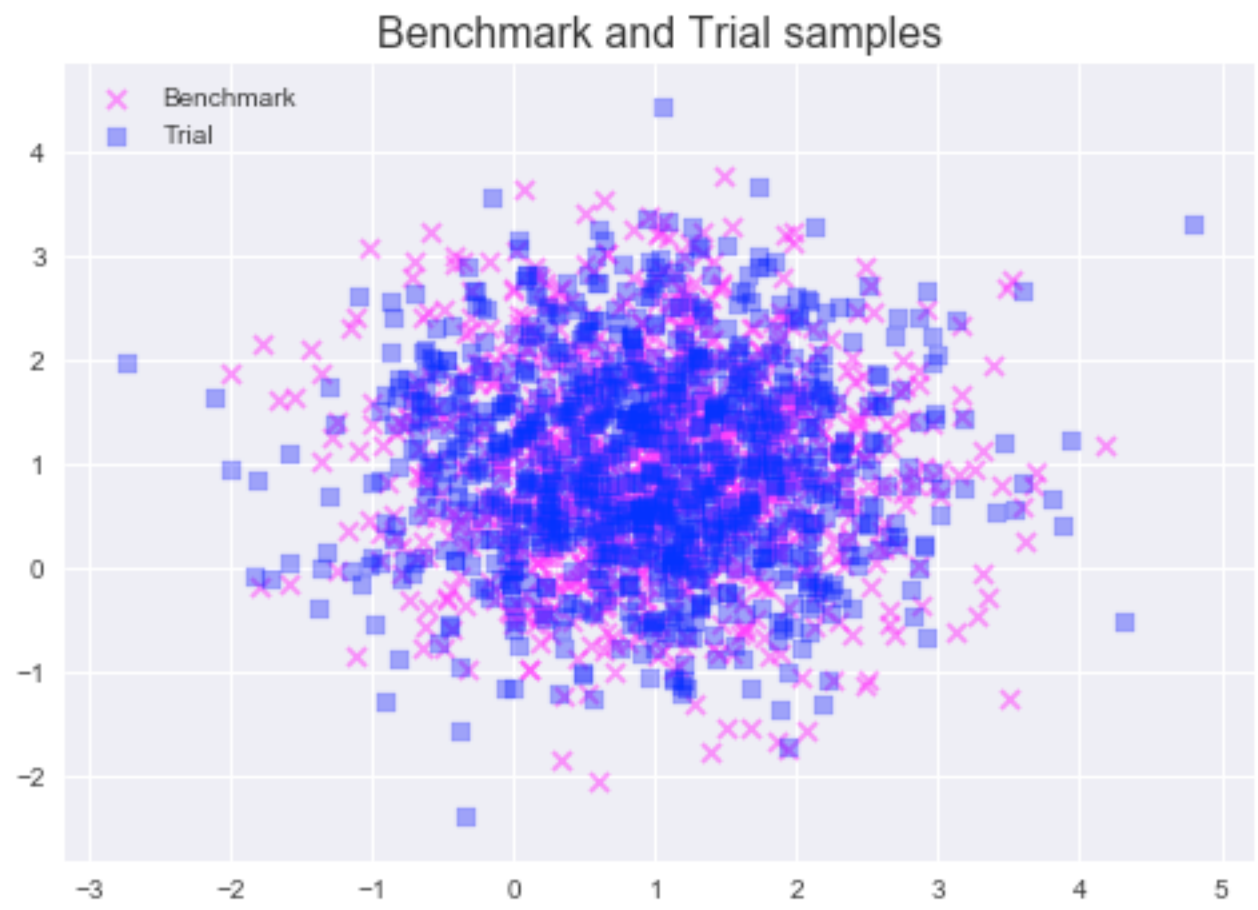
Trial: $\mathcal{T} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_T}\} \overset{\text{iid}}{\sim} p_T$

Benchmark: $\mathcal{B} = \{\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{N_B}\} \overset{\text{iid}}{\sim} p_B$

$\boldsymbol{x}_i, \boldsymbol{x}'_i \in \mathbb{R}^D$

probability distributions p_B, p_T unknown

Are B,T drawn from the same prob. distribution?



… hard!

> Two-sample Test

RECIPE:

1. Density Estimator

→ reconstruct PDFs from samples

2. Test Statistic (TS)

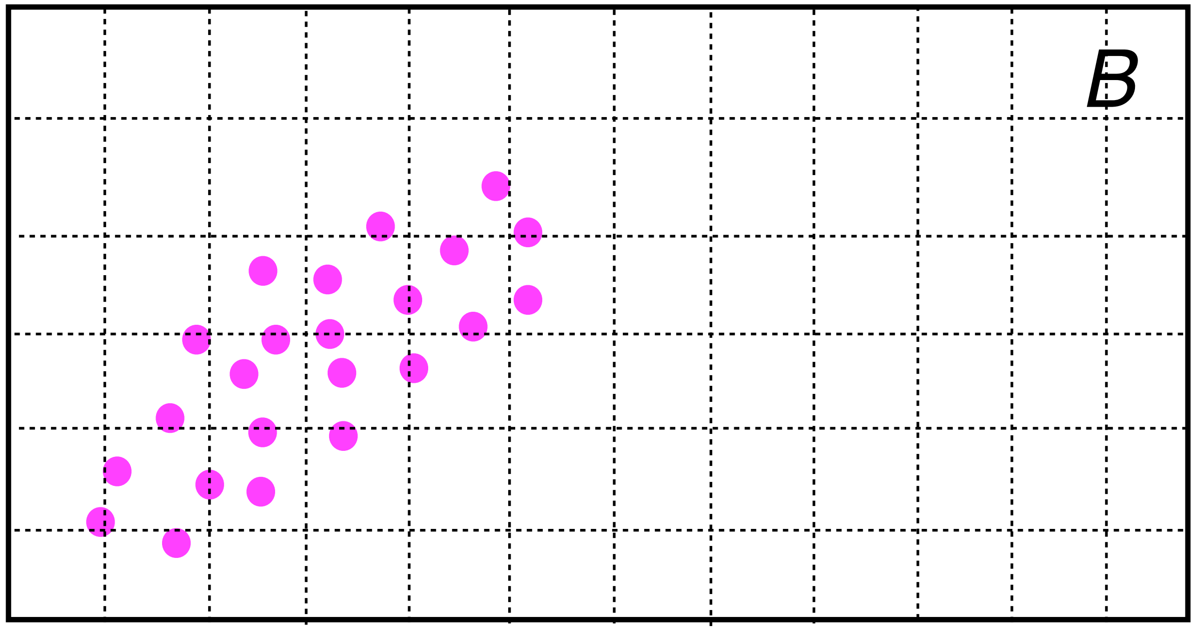
→ measure “distance” between PDFs

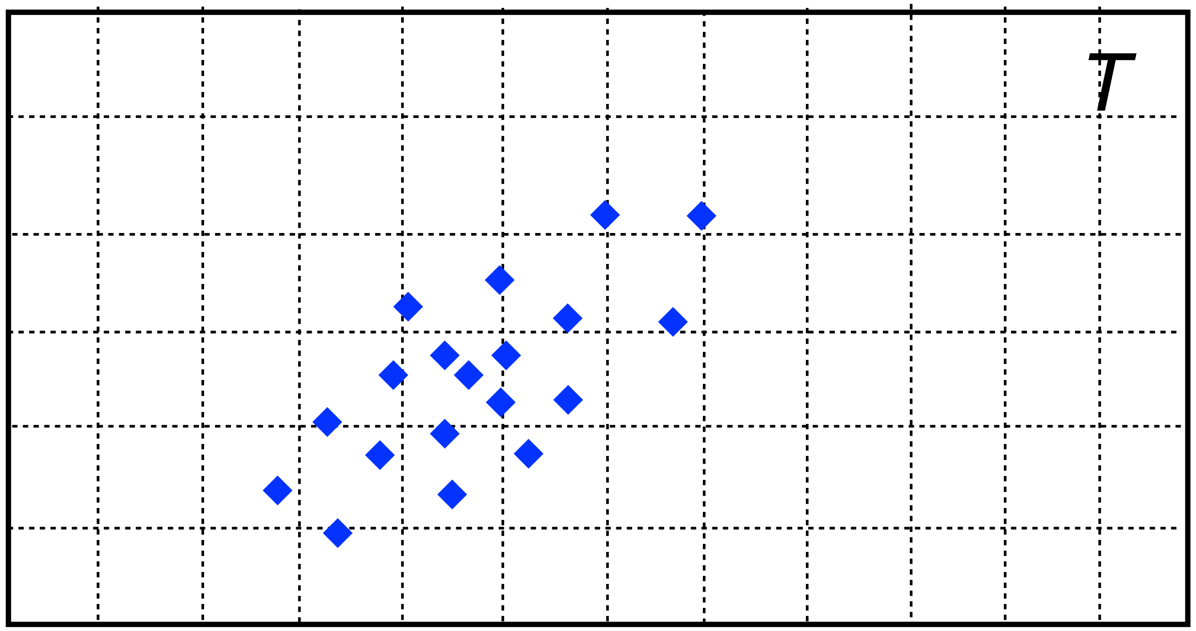
3. TS distribution

→ associate probabilities to TS
under null hypothesis H_0: $p_B = p_T$

4. *p* -value

→ accept/reject H_0





Divide the space in squared bins?

- ✓ easy
- ✓ can use simple statistics (e.g. χ^2)
- ✘ hard/slow/impossible in high-*D*

Need un-binned multivariate approach

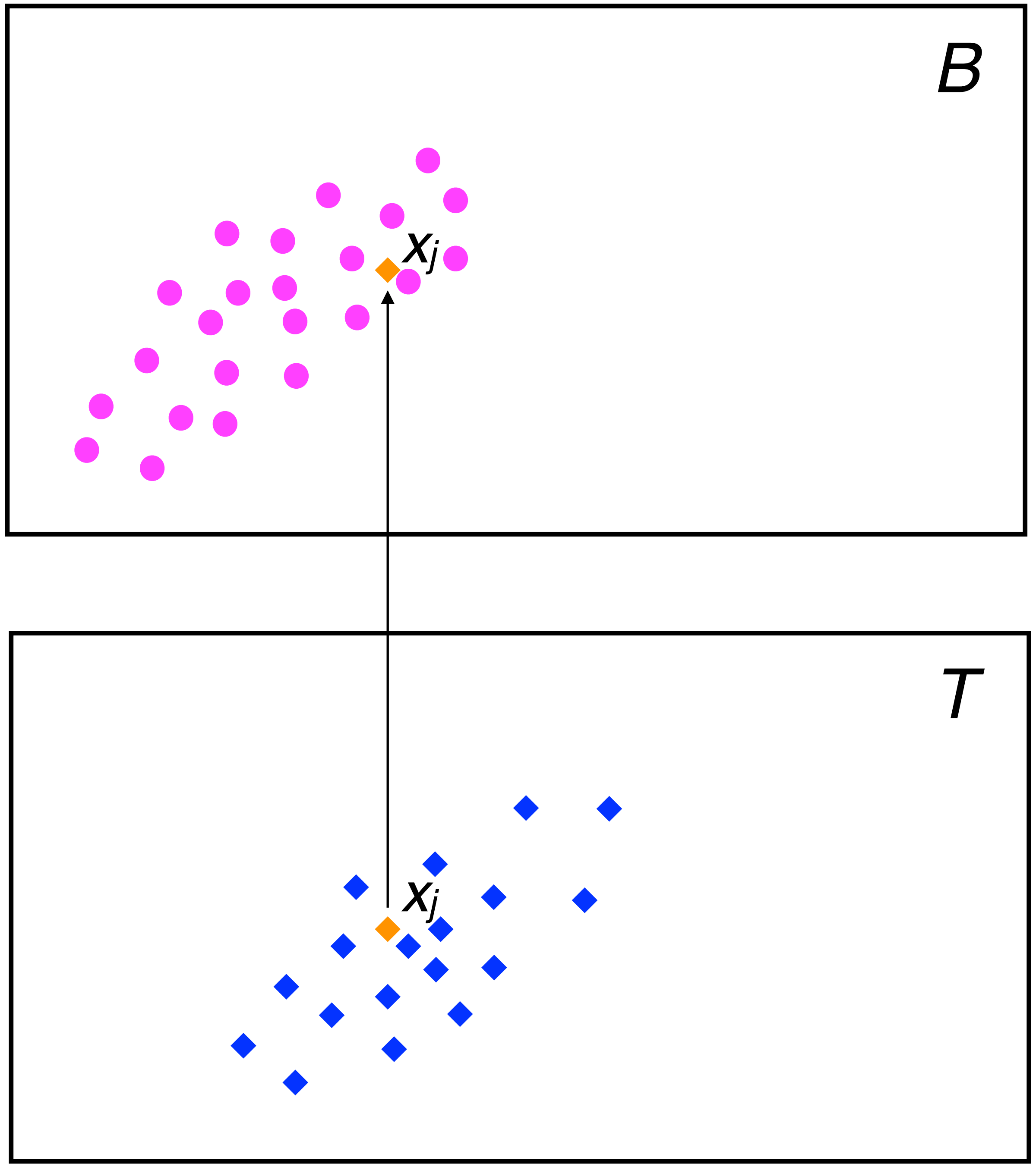
Find PDFs *estimators*: $\hat{p}_B(\boldsymbol{x}), \hat{p}_T(\boldsymbol{x})$
e.g. based on densities of points:

$$\hat{p}_{B,T}(\boldsymbol{x}) = \frac{\rho_{B,T}(\boldsymbol{x})}{N_{B,T}}$$

Nearest Neighbors!

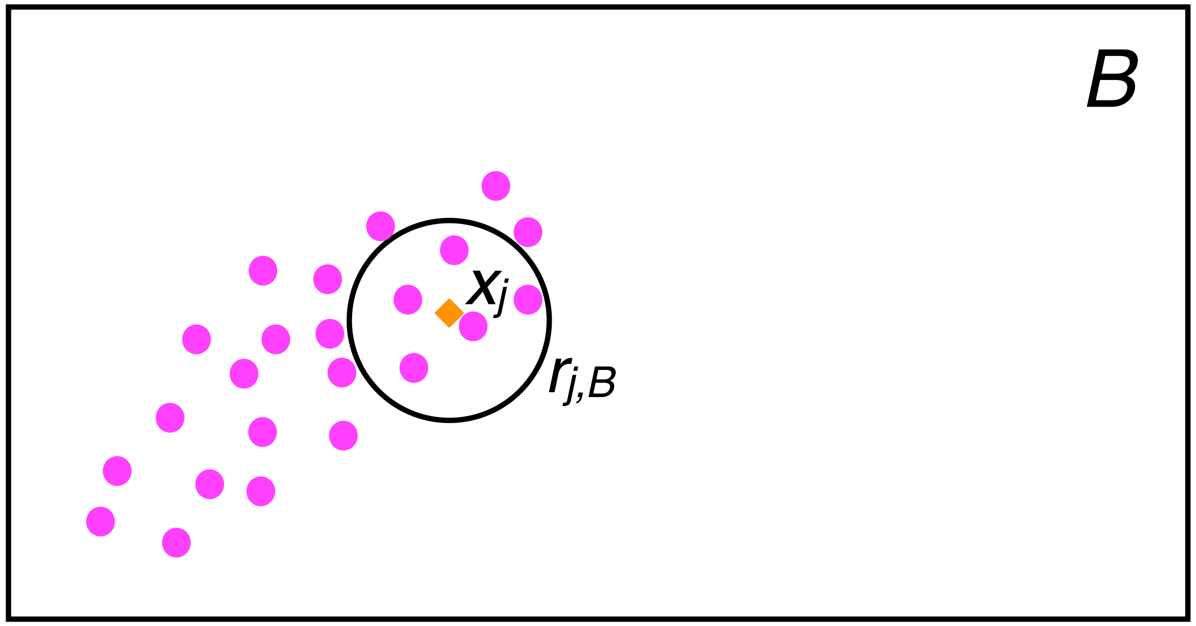
[Schilling - 1986][Henze - 1988]
[Wang et al. - 2005,2006]
[Dasu et al. - 2006][Perez-Cruz - 2008]
[Sugiyama et al. - 2011][Kremer et al, 2015]

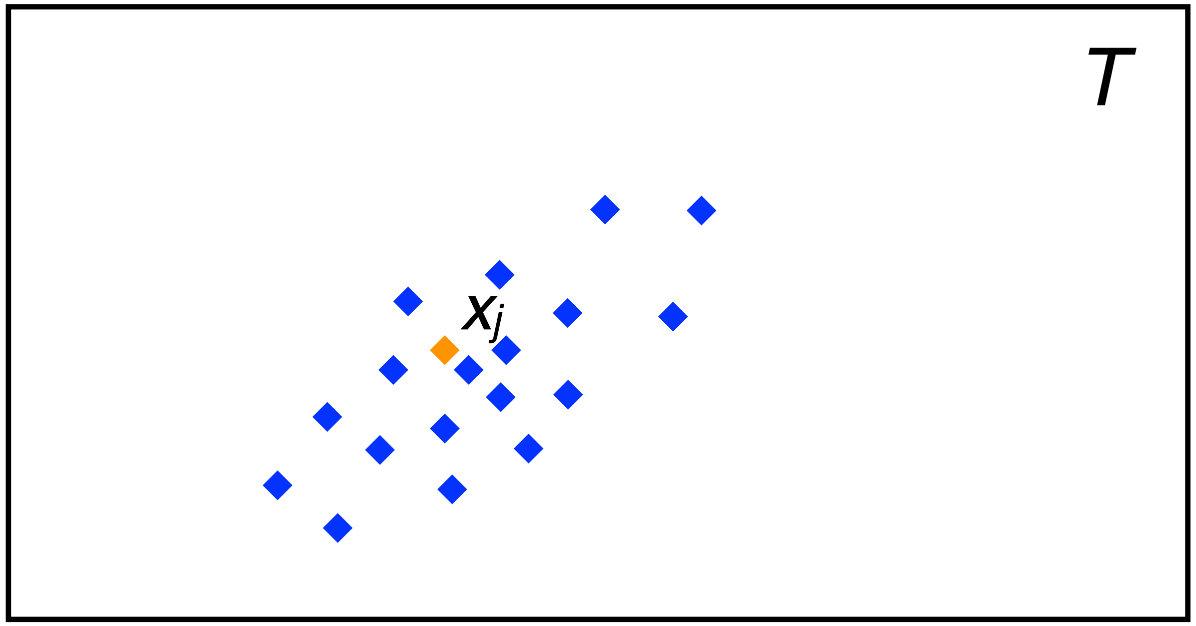
> 1. Density Estimator



- Fix integer K.
- Choose query point x_j in T and draw it in B.

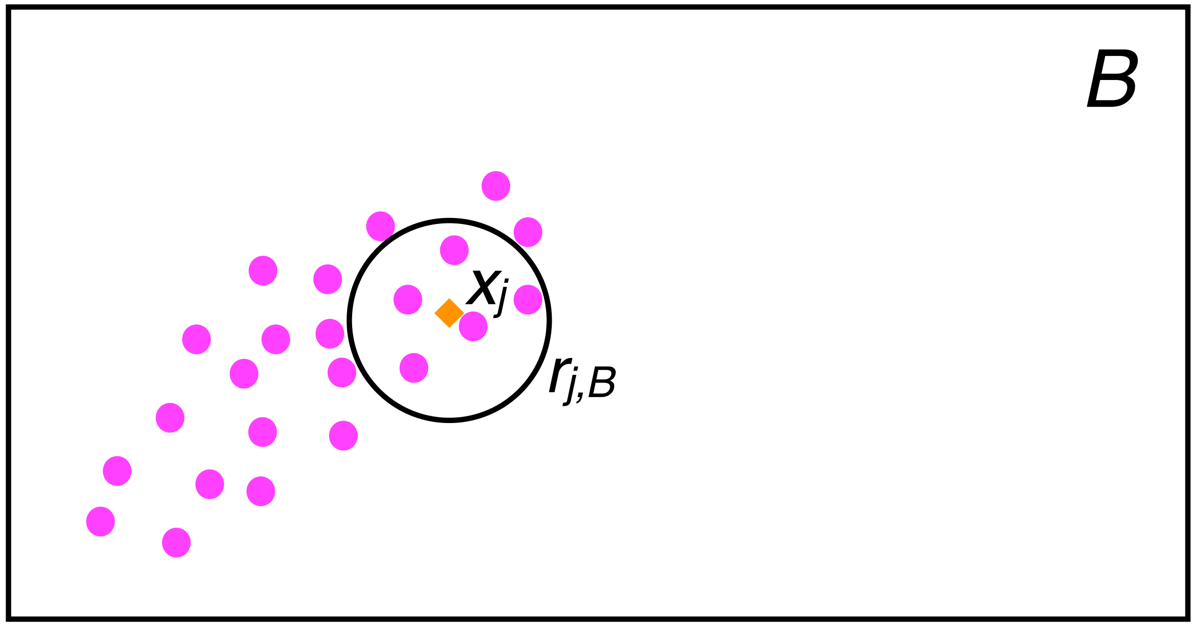
> 1. Density Estimator

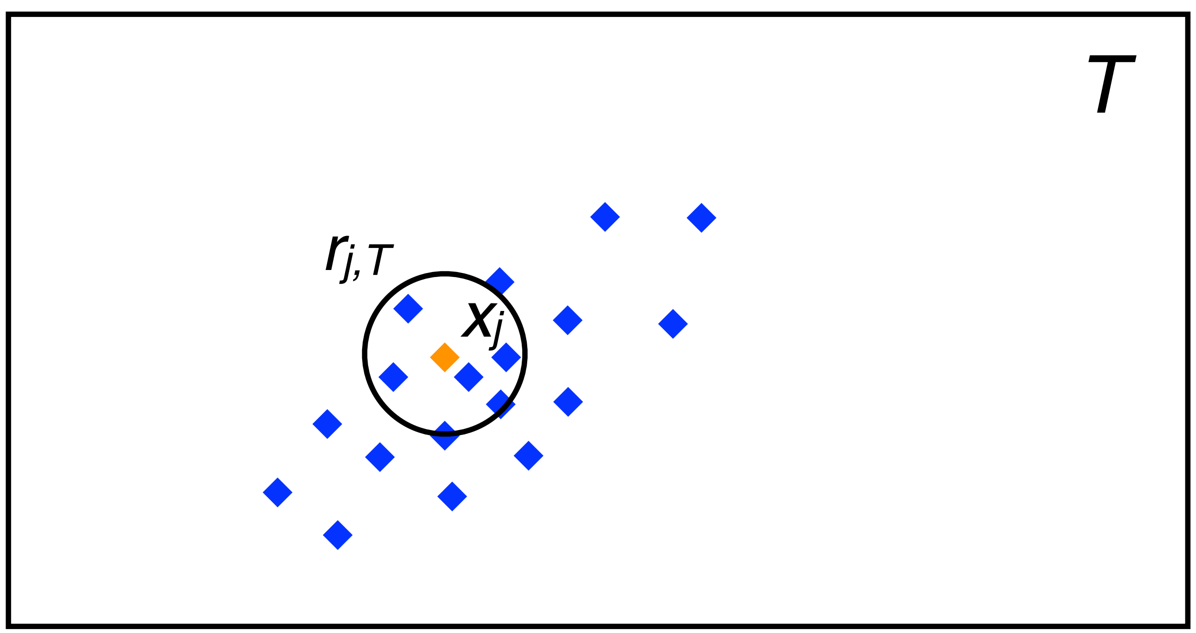




- Fix integer K.
- Choose query point x_j in T and draw it in B.
- Find the distance $r_{j,B}$ of the Kth-NN of x_j in B.

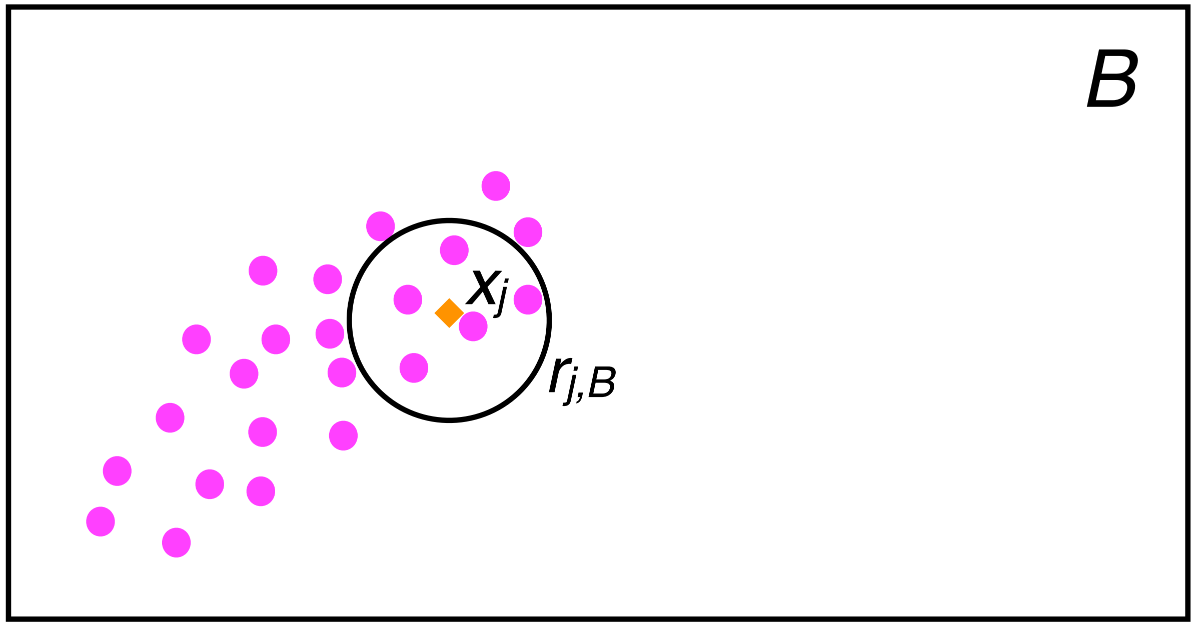
> 1. Density Estimator

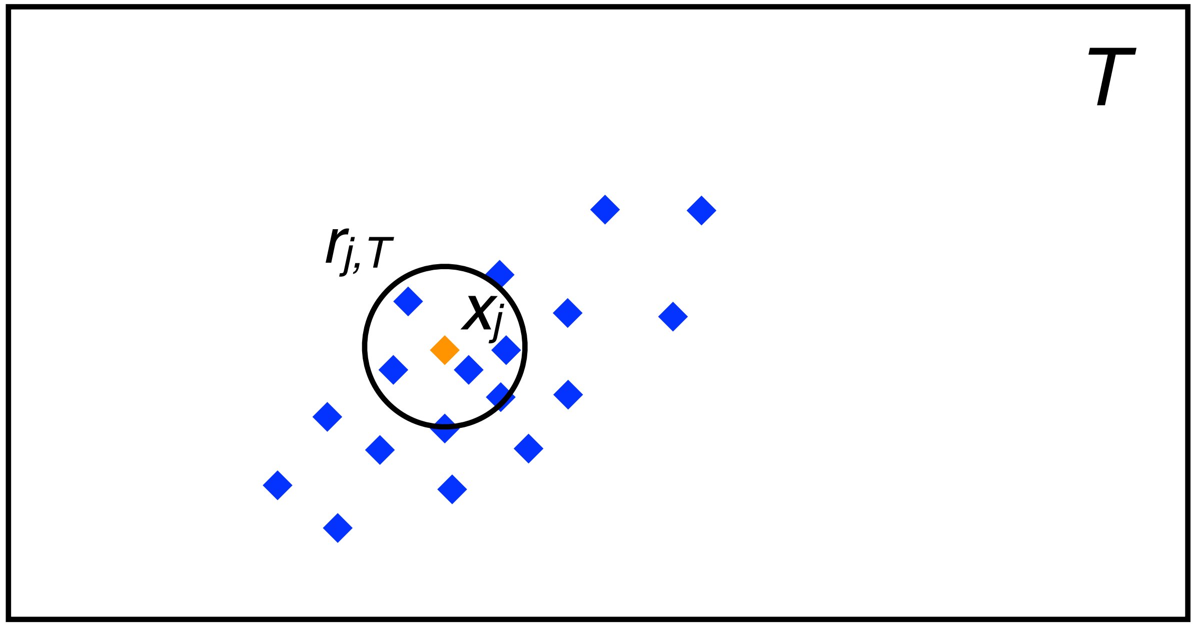




- Fix integer K.
- Choose query point x_j in T and draw it in B.
- Find the distance $r_{j,B}$ of the Kth-NN of x_j in B.
- Find the distance $r_{j,T}$ of the Kth-NN of x_j in T.

> 1. Density Estimator





- Fix integer K.
- Choose query point x_j in T and draw it in B.
- Find the distance $r_{j,B}$ of the Kth-NN of x_j in B.
- Find the distance $r_{j,T}$ of the Kth-NN of x_j in T.
- Estimate PDFs:

$$\hat{p}_B(\boldsymbol{x}_j) = \frac{K}{N_B} \frac{1}{\omega_D r_{j,B}^D}$$

$$\hat{p}_T(\boldsymbol{x}_j) = \frac{K}{N_T - 1} \frac{1}{\omega_D r_{j,T}^D}$$

> 2. Test Statistic

- Measure of the “distance” between 2 PDFs

- Define **Test Statistic**:
  (detect under-/over-densities)

$$\boxed{\mathrm{TS}(\mathcal{B},\mathcal{T}) = \frac{1}{N_T}\sum_{j=1}^{N_T}\log\frac{\hat{p}_T(\boldsymbol{x}_j)}{\hat{p}_B(\boldsymbol{x}_j)}}$$

- Related to Kullback-Leibler divergence as: $\mathrm{TS}(\mathcal{B},\mathcal{T}) = \hat{D}_{\mathrm{KL}}(\hat{p}_T||\hat{p}_B)$

$$D_{\mathrm{KL}}(p||q) \equiv \int_{\mathbb{R}^D} p(\boldsymbol{x})\log\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}d\boldsymbol{x}$$

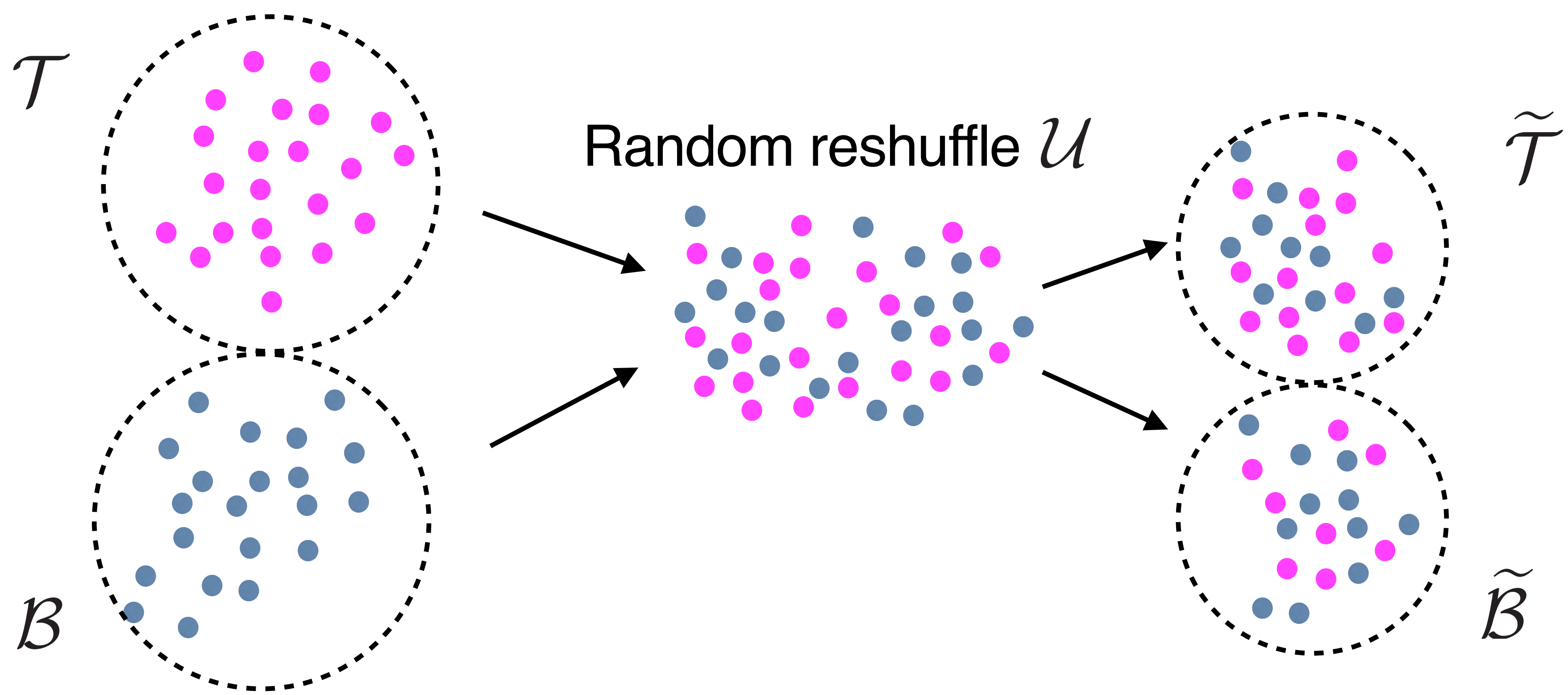
- From NN-estimated PDFs: $\mathrm{TS}_{\mathrm{obs}} = \frac{D}{N_T}\sum_{j=1}^{N_T}\log\frac{r_{j,B}}{r_{j,T}} + \log\frac{N_B}{N_T-1}$

- **Theorem:** this estimator converges to $D_{\mathrm{KL}}(p_B||p_T)$,
  in large sample limit

[Wang et al. - 2005,2006]

How is TS distributed? **Permutation test!**

Assume p_B=p_T. Union set: $\mathcal{U} = \mathcal{T} \cup \mathcal{B}$

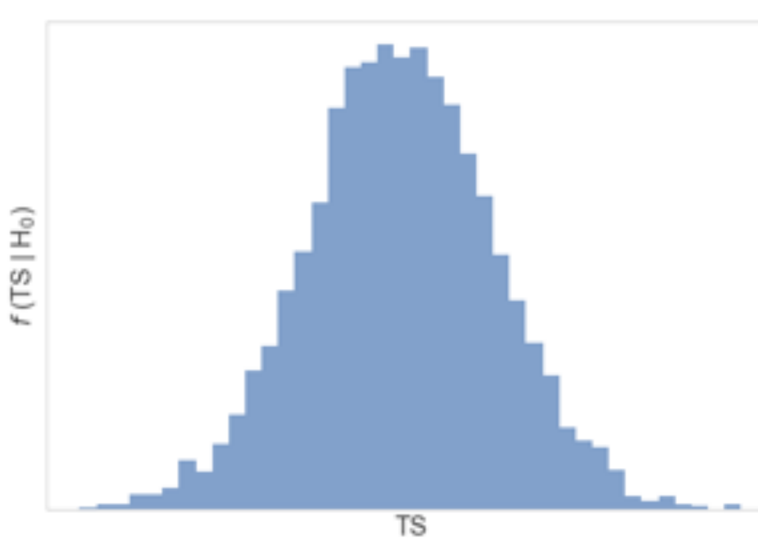


Compute the test statistic TS_n on:

$$(\tilde{\mathcal{B}}, \tilde{\mathcal{T}})$$

Repeat many times.

Distribution of TS under H_0: $f(\mathrm{TS}|H_0) \leftarrow \{\mathrm{TS}_n\}$
[asymptotically normal with zero mean]



> 4. *p*-value

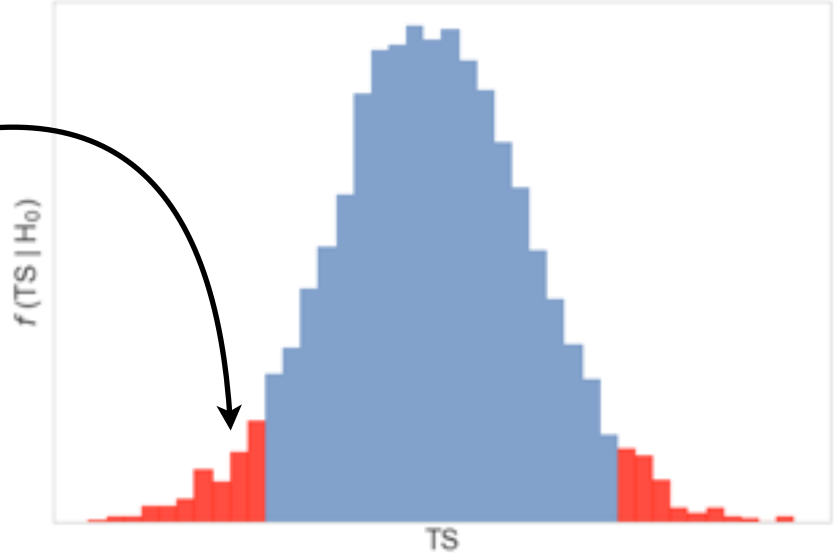
- $\hat{\mu}, \hat{\sigma}$: mean,variance of TS distribution $f(\mathrm{TS}|H_0)$

- Standardize the TS: $\mathrm{TS} \to \mathrm{TS}' \equiv \dfrac{\mathrm{TS} - \hat{\mu}}{\hat{\sigma}}$

- TS' distributed according to $f'(\mathrm{TS}'|H_0) = \hat{\sigma} f(\hat{\mu} + \hat{\sigma}\mathrm{TS}'|H_0)$

- Two-sided *p*-value:

$$p = 2\int_{|\mathrm{TS}'_{\mathrm{obs}}|}^{+\infty} f'(\mathrm{TS}'|H_0)\, d\mathrm{TS}'$$

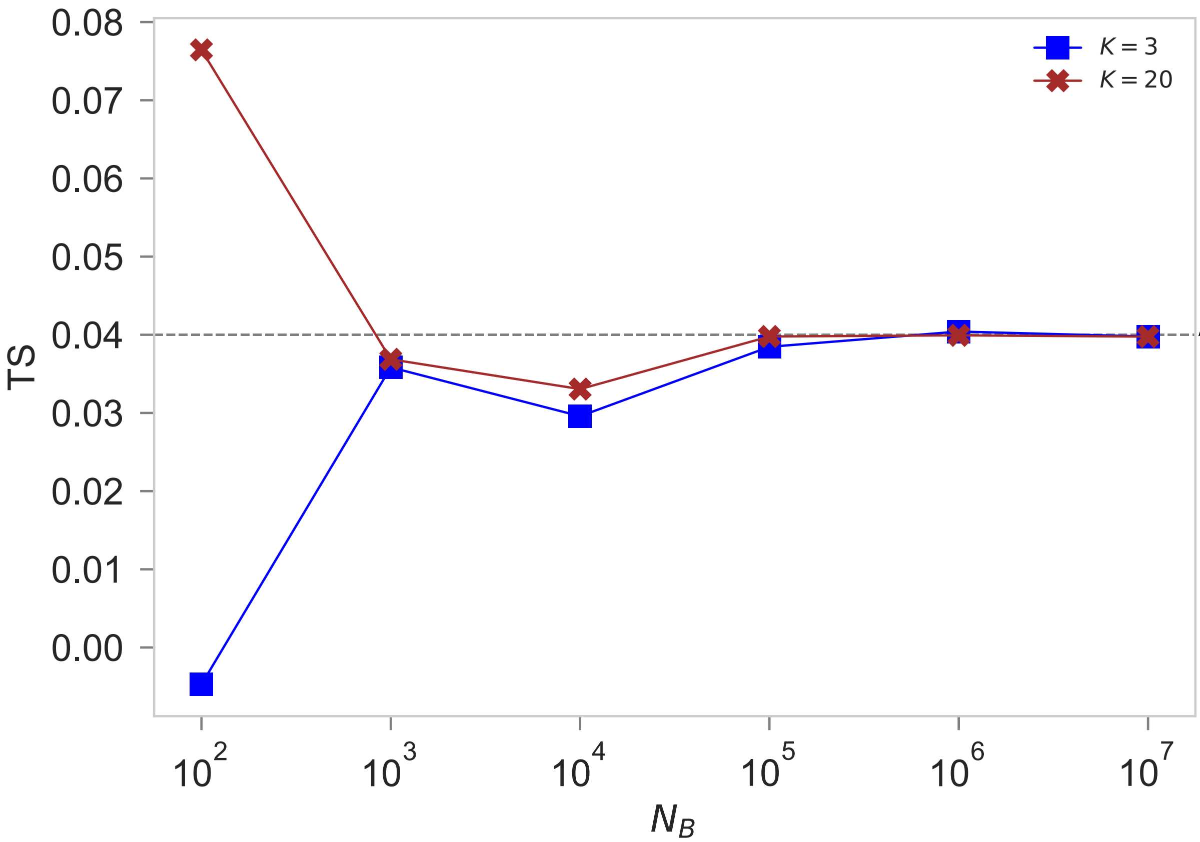


- Equivalent significance: $Z \equiv \Phi^{-1}(1 - p/2)$

> 2D Gaussian Example

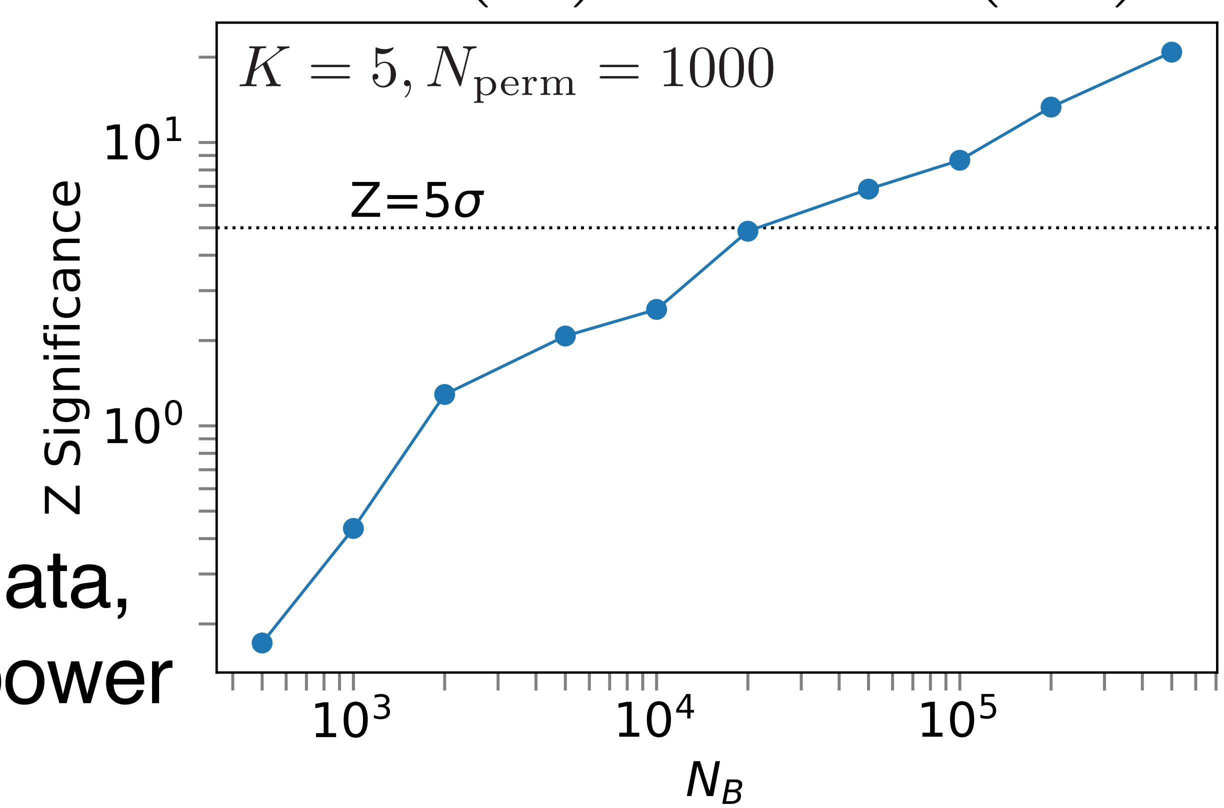
$$p_B = \mathcal{N}(\boldsymbol{\mu}_B, \Sigma_B) \qquad p_T = \mathcal{N}(\boldsymbol{\mu}_T, \Sigma_T) \qquad \Sigma_B = \Sigma_T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\boldsymbol{\mu}_B = \begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix} \qquad \boldsymbol{\mu}_T = \begin{pmatrix} 1.2 \\ 1.2 \end{pmatrix}$$



exact KL divergence

$$\boldsymbol{\mu}_B = \begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix} \qquad \boldsymbol{\mu}_T = \begin{pmatrix} 1.15 \\ 1.15 \end{pmatrix}$$



more data, more power

INPUT:

Trial sample: $\mathcal{T} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_T}\} \overset{\text{iid}}{\sim} p_T$

Benchmark sample: $\mathcal{B} = \{\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{N_B}\} \overset{\text{iid}}{\sim} p_B$

$\boldsymbol{x}_i, \boldsymbol{x}'_i \in \mathbb{R}^D$

p_B, p_T unknown

K: number of nearest neighbors

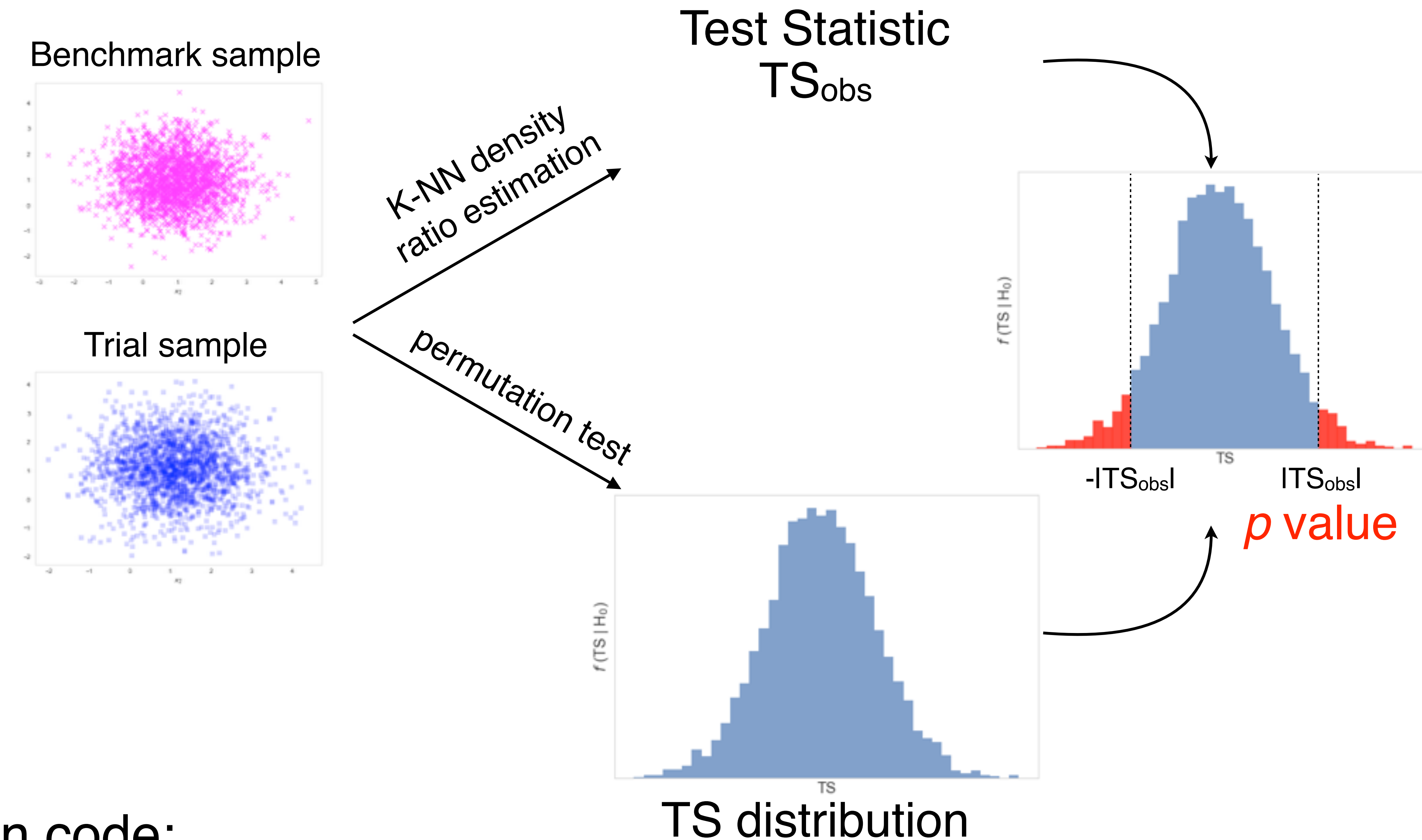
N_{perm}: number of permutations

OUTPUT:

***p*-value of the null hypothesis H_0: $p_B = p_T$**

[check compatibility between 2 samples]

> NN2ST: Summary



Python code:
github.com/de-simone/NN2ST

1. Statistical test of dataset compatibility

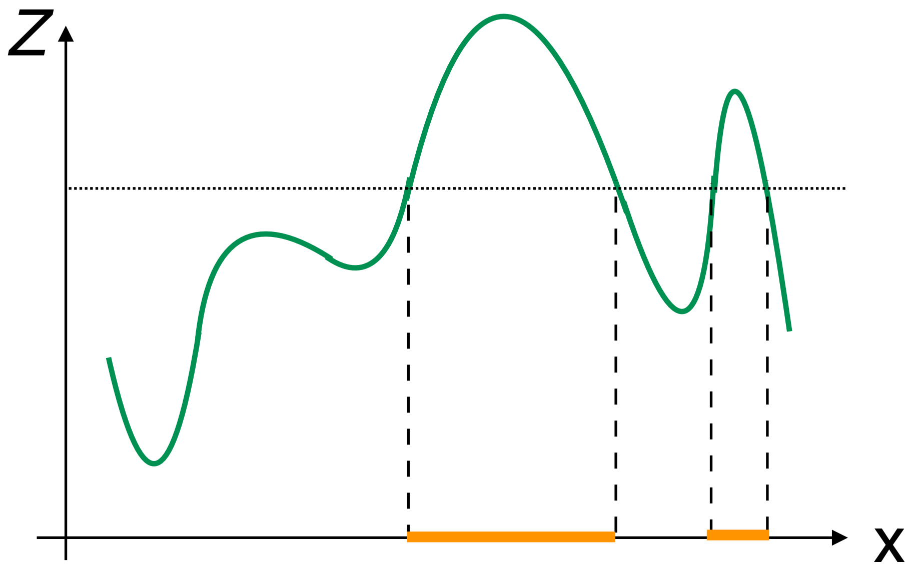
- Nearest-Neighbors Two-Sample Test
- Identify Discrepancies
- Include Uncertainties

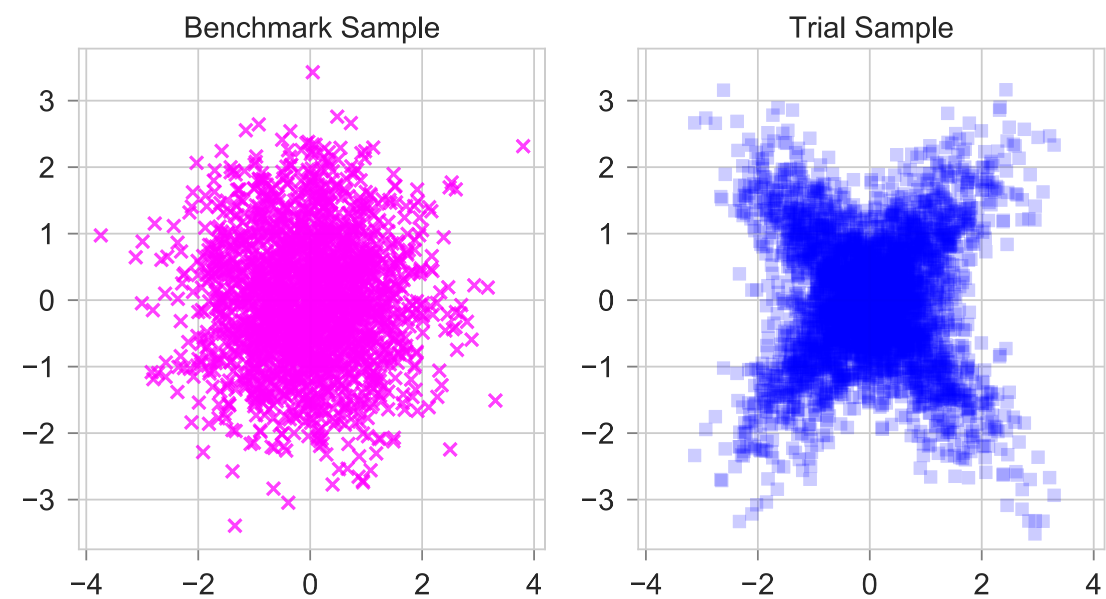
2. Applications to High-Energy Physics

> Where are the discrepancies?

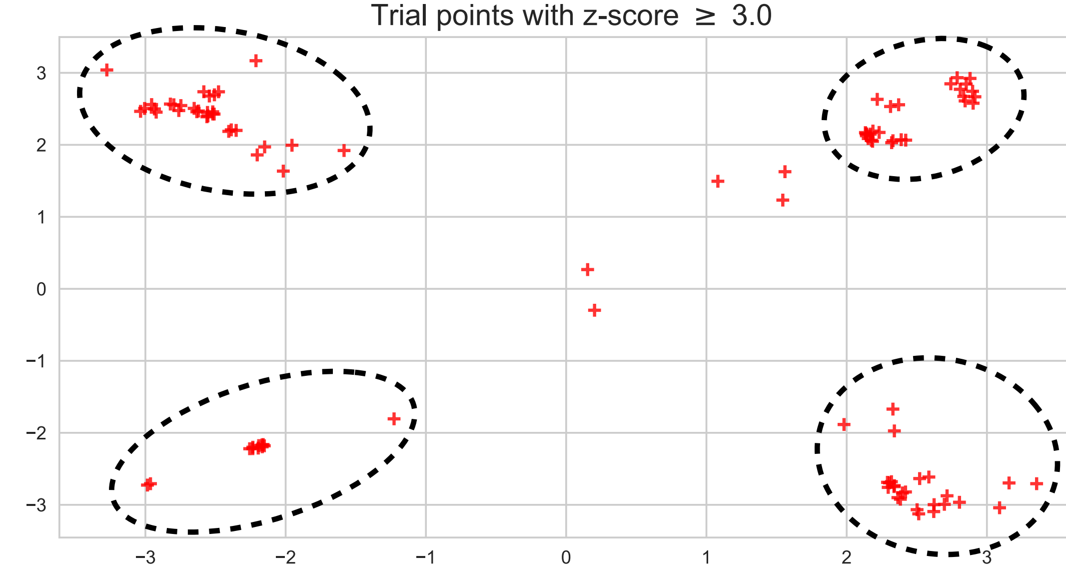
Bonus: Characterize regions with significant discrepancies

1. "Score" field over T: $$Z(\boldsymbol{x}_j) \equiv \frac{u(\boldsymbol{x}_j) - \bar{u}}{\sigma_u}$$ with: $$u(\boldsymbol{x}_j) \equiv \log \frac{r_{j,B}}{r_{j,T}}$$ $$\mathrm{TS}_{\mathrm{obs}} = D\,\bar{u} + \mathrm{const}$$





2. Identify points where $Z(\boldsymbol{x}) > c$
   They contribute the most to large $\mathrm{TS}_{\mathrm{obs}}$
   ⟶ high-discrepancy (anomalous) regions



3. Apply a clustering algorithm to group them

1. Statistical test of dataset compatibility

- Nearest-Neighbors Two-Sample Test
- Identify Discrepancies
- Include Uncertainties

2. Applications to High-Energy Physics

> Sample Uncertainties

How to include sample uncertainties?

1. Model feature uncertainties

$$F_{\mathcal{B}}(\mathbf{x}), F_{\mathcal{T}}(\mathbf{x})$$

[e.g. zero-mean gaussians]

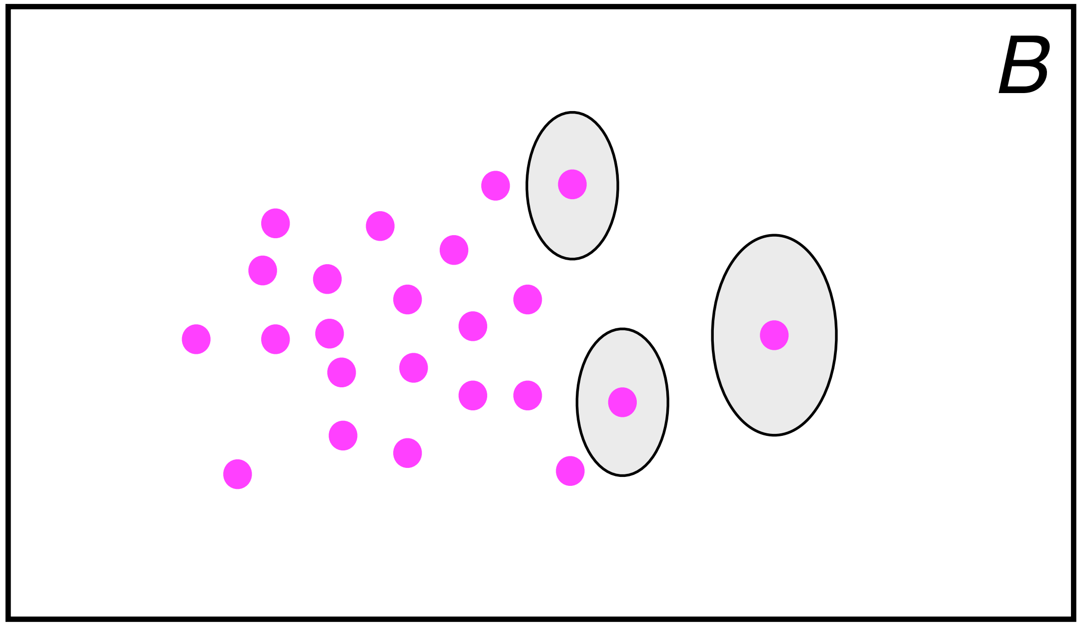
2. New samples by adding random noise sampled from $F_{B,T}$:

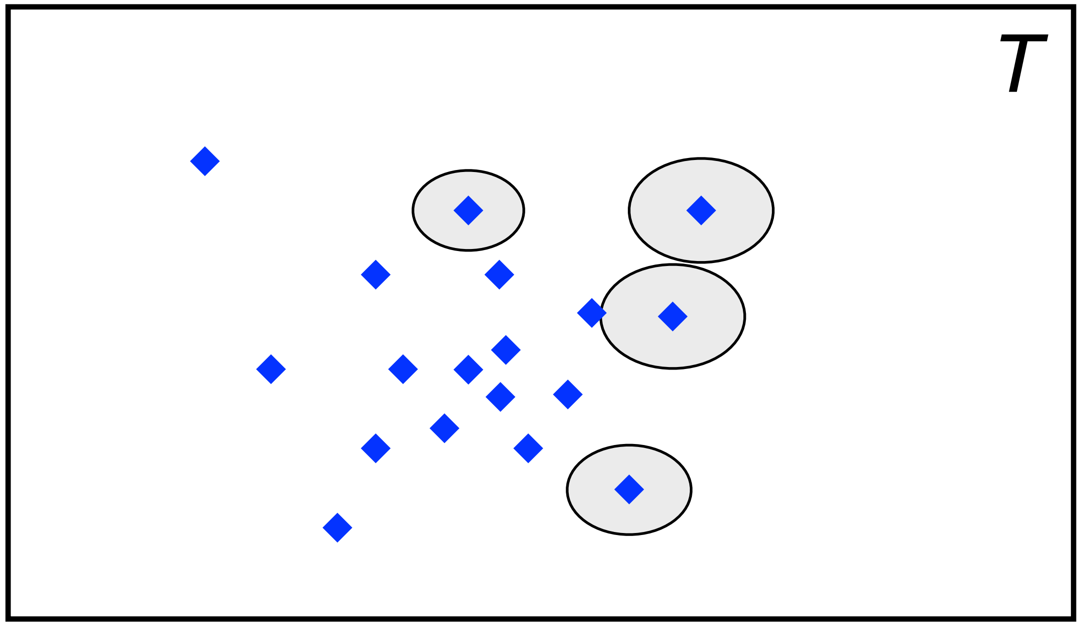
$$\begin{aligned} \mathcal{T}_u &= \{\boldsymbol{x}_i + \Delta\mathbf{x}_i\}_{i=1}^{N_T} \\ \mathcal{B}_u &= \{\boldsymbol{x}'_i + \Delta\mathbf{x}'_i\}_{i=1}^{N_B} \end{aligned}$$

3. Compute TS on new samples

$$\mathrm{TS}_u \equiv \mathrm{TS}(\mathcal{B}_u, \mathcal{T}_u) = \mathrm{TS}_{\mathrm{obs}} + U$$

4. Repeat many times to reconstruct *f(U)*





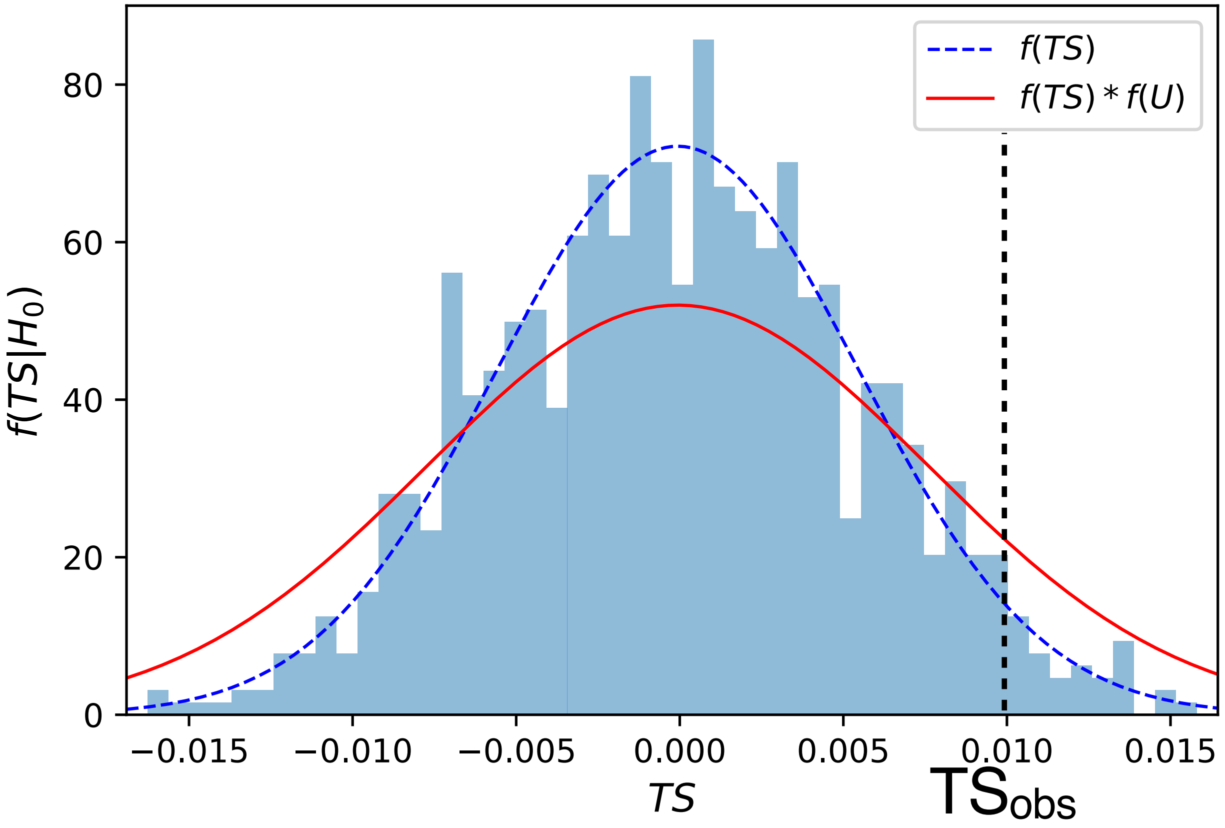
> Sample Uncertainties

How to include sample uncertainties?

- f(TS$_u$) is a convolution: $f(\mathrm{TS}_u|H_0) = f(\mathrm{TS}|H_0) * f(U)$
  f(TS$_u$) more spread than f(TS)

- *p*-value computed from f(TS$_u$)

- weaker significance, power degradation



> 2D Gaussian with Uncertainties

B,T gaussian samples:

$$p_B = \mathcal{N}(\boldsymbol{\mu}_B, \Sigma_B) \qquad p_T = \mathcal{N}(\boldsymbol{\mu}_T, \Sigma_T)$$

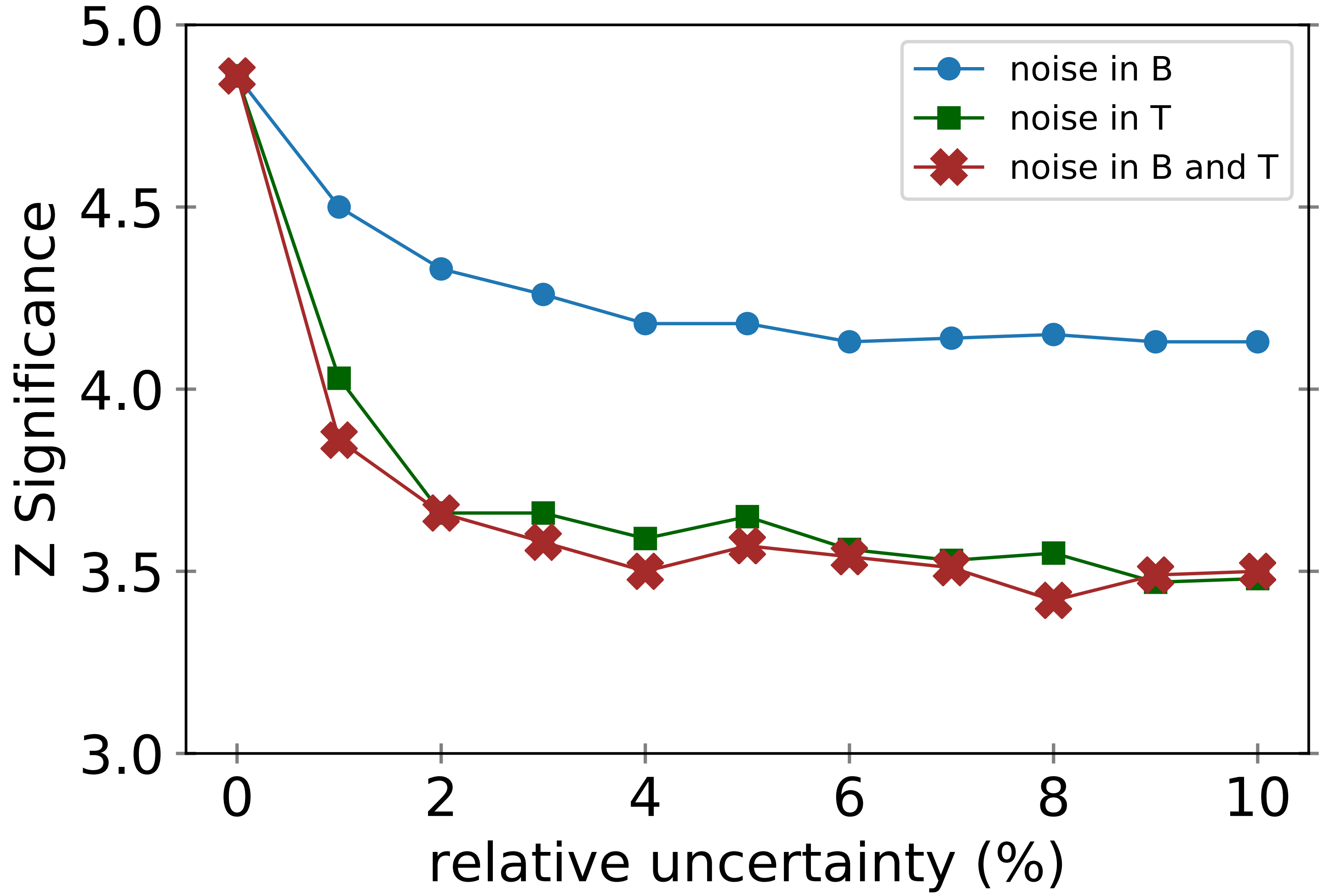
$$\boldsymbol{\mu}_B = \begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix} \qquad \boldsymbol{\mu}_T = \begin{pmatrix} 1.15 \\ 1.15 \end{pmatrix}$$

$$\Sigma_B = \Sigma_T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

gaussian uncorrelated errors
(diagonal covariance)
with fixed relative uncertainty

$$\sigma_i = \epsilon x_i$$

for each feature component *i*



> NN2ST: Summary

- ✓ general, model-independent
- ✓ fast, no optimization

  [$N_{B,T}$=20k, K=5, N_{perm}=1k, D=2: $t \sim$ **2 mins**
  $N_{B,T}$=20k, K=5, N_{perm}=1k, D=8: $t \sim$ **50 mins**]
- ✓ sensitive to **unspecified** signals
- ✓ useful when no variable can separate sig/bkg
- ✓ helps finding signal regions, optimal cuts, …
- ✓ flexible to incorporate uncertainties

- ✘ need to run for each sample pair
- ✘ permutation test is bottleneck

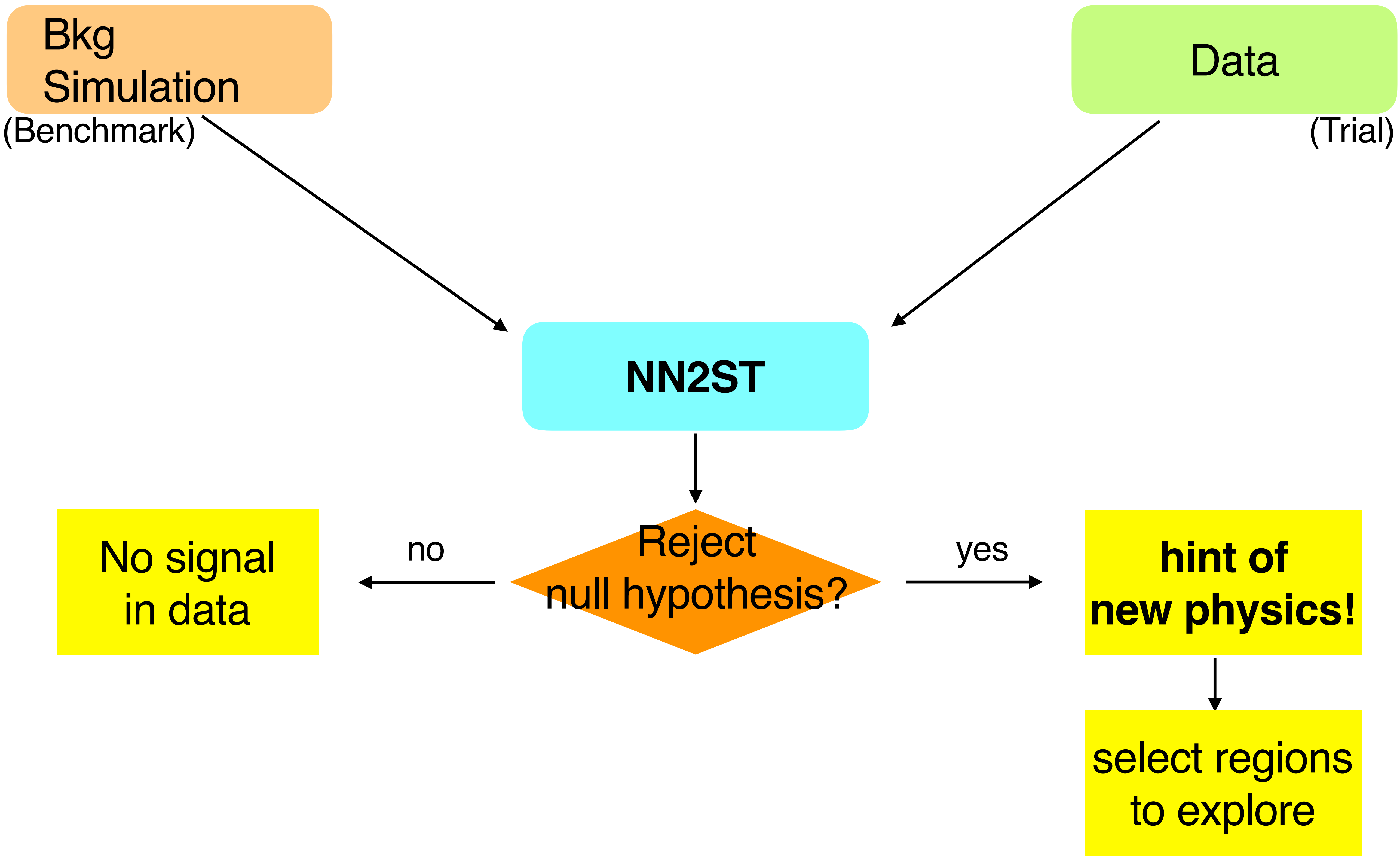
> Outline

1. Statistical test of dataset compatibility

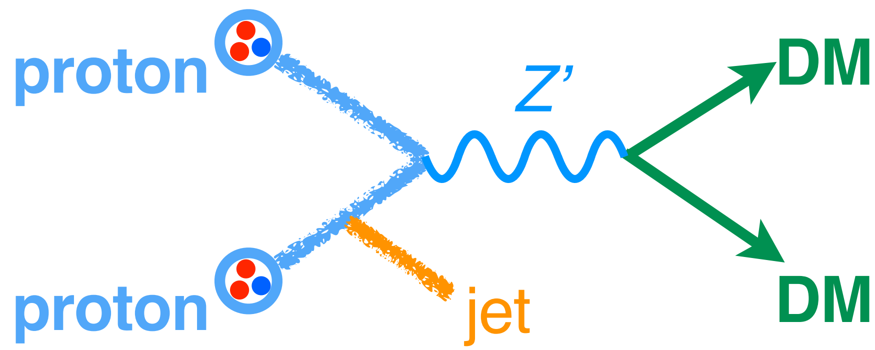
- Nearest-Neighbors Two-Sample Test
- Identify Discrepancies
- Include Uncertainties

2. Applications to High-Energy Physics

> Our Method

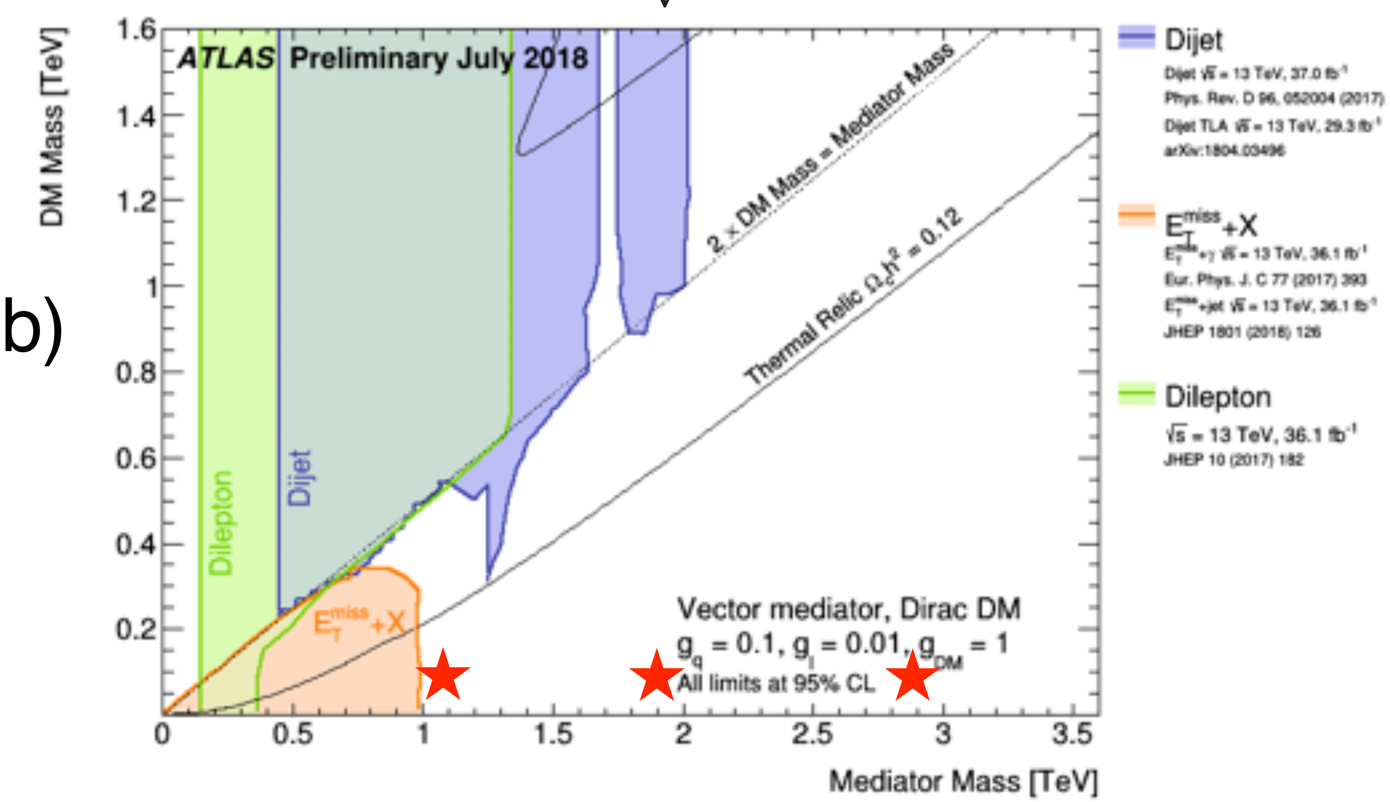


Bkg Simulation
(Benchmark)
Data
(Trial)
NN2ST
Reject null hypothesis?
no
No signal in data
yes
hint of new physics!
select regions to explore



DM + Z'
vector mediator

m_{DM} = 100 GeV
$m_{Z'}$ = 1.2, 2, 3 TeV
g_{DM} = 1, g_q = 0.1
$\sqrt{s} = 13$ TeV



- "proof-of-principle" study
- bkg: $Z \to \nu\bar{\nu} + (1,2)\, j$ ($\sigma_{\rm bkg}$=202.6 pb) sub-leading bkgs not included
- no full detector effects (generic Delphes profile)

Benchmark: **$\mathbf{BKG_1}$**
Trials: **$\mathbf{BKG_2}$ + SIG**
$K = 5$
$N_{perm} = 3000$

<u>8 features</u>:
- n. of jets
- p_T, η of 2 leading jets
- $E_T^{\rm miss}, H_T$
- $\Delta\phi_{E_T^{\rm miss}, j_1}$

> DM search @ LHC

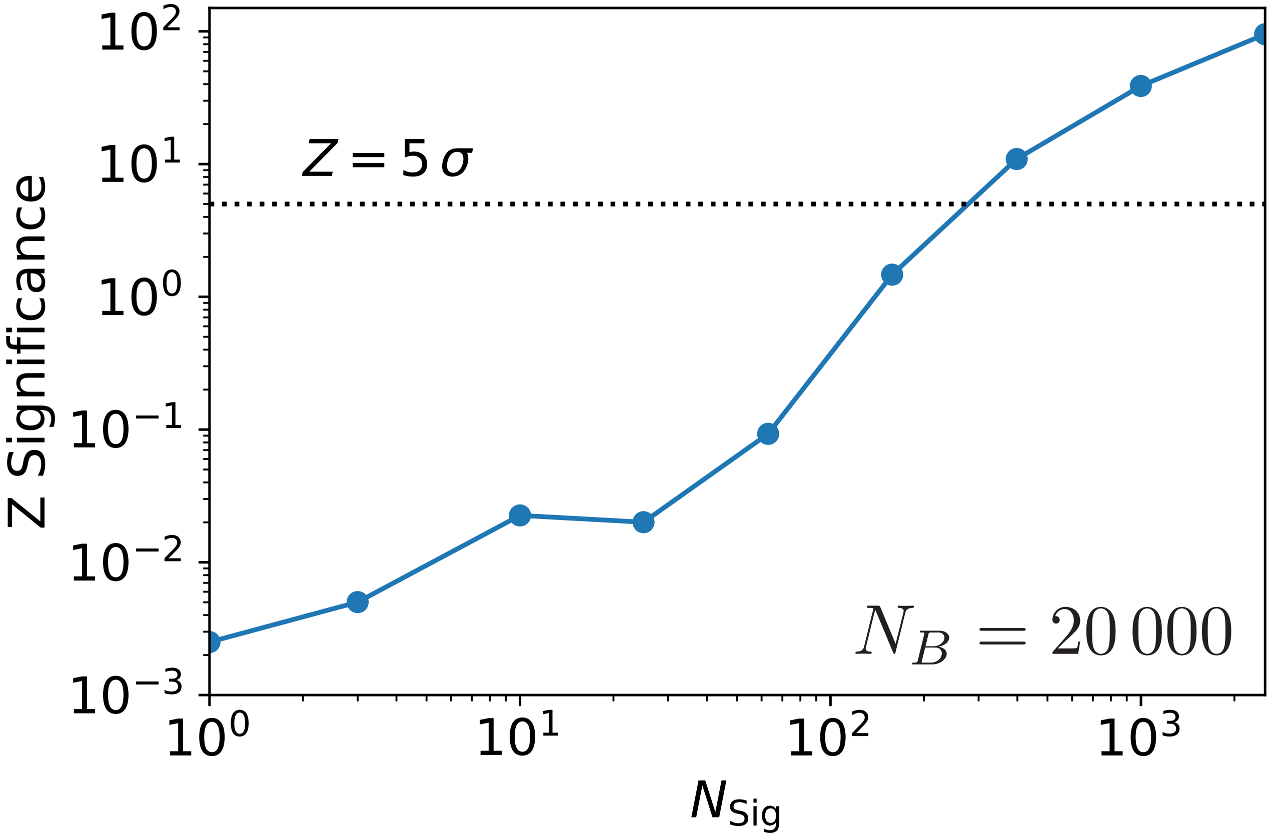
B: **BKG$_1$** (20k events)
T1: **BKG$_2$** (20k events) **+ SIG$_1$** (2010 events)
T2: **BKG$_2$** (20k events) **+ SIG$_2$** (375 events)
T3: **BKG$_2$** (20k events) **+ SIG$_3$** (59 events)

$$N_{\rm sig} = N_B \times \frac{\sigma_{\rm signal}}{\sigma_{\rm bkg}}$$

Sample	$M_{Z'}$	$\sigma_{\rm signal}$	$Z_{\text{no uncert.}}$	$Z_{\text{10\% rel uncert.}}$
T1	1.2 TeV	20.4 pb	40 σ	26 σ
T2	2 TeV	3.8 pb	13 σ	12 σ
T3	3 TeV	0.6 pb	2.7 σ	2.5 σ

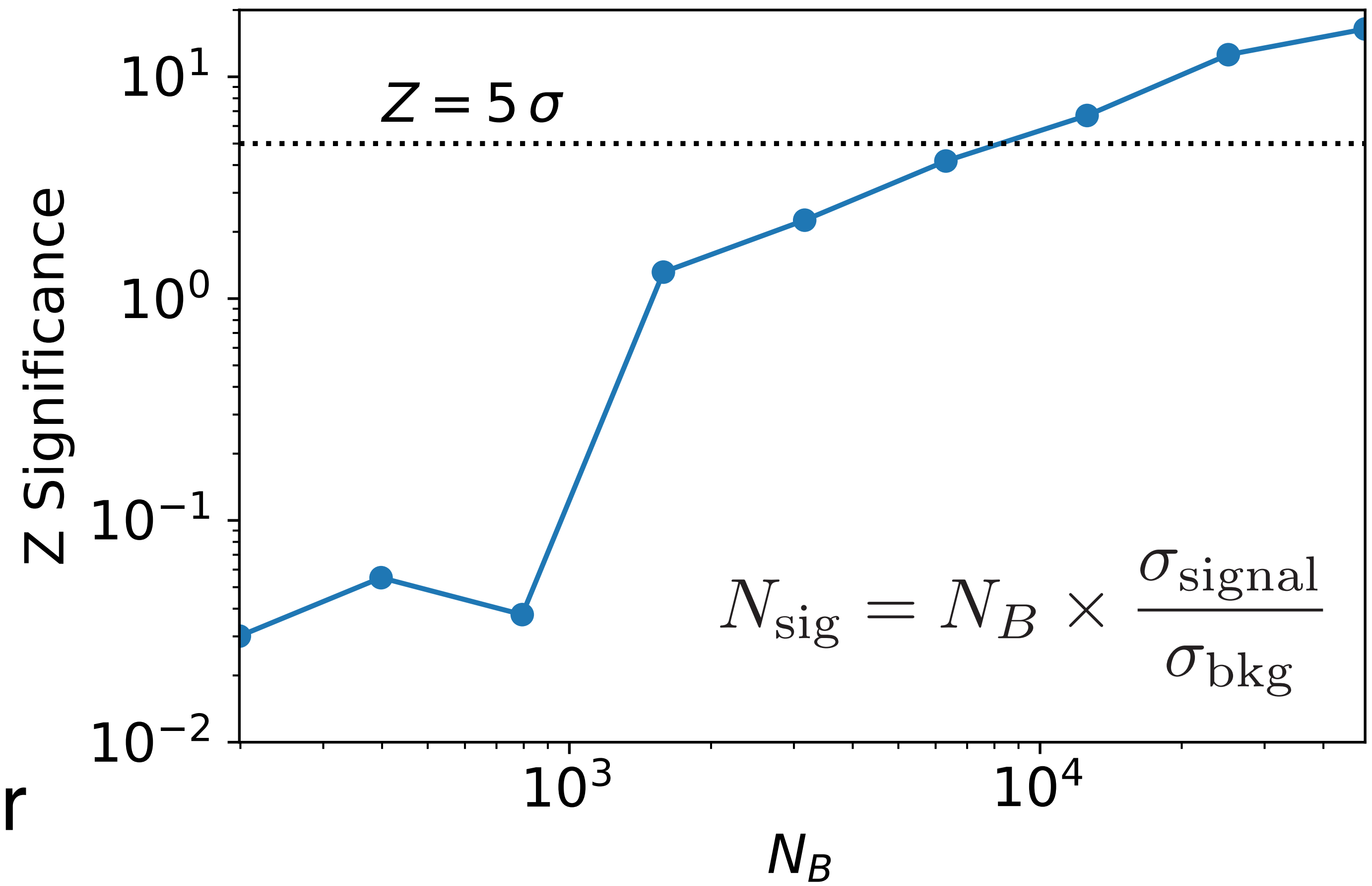
still not real-world

- systematics: expect further degradation of results
- **the method has value, it is worth exploring**



stronger signal
easier to discover

$$N_T = N_B + N_{\mathrm{sig}}$$



more data,
more power

Directions for future work:

- adaptive choice of K

- identifying discrepant regions in realistic situations
(with Z-score method)

- validation tool for bkg:
compatibility between MC sims. and data in control regions

- scalability

- … your suggestions?

> Take-Home Messages

1. New Statistical Test for BSM Physics

- assess degree of compatibility between 2 samples
- rooted on nearest neighbors, solid math foundations

2. NN2ST as a discovery tool

- powerful and model-independent
- lots of applications

3. NN2ST to guide searches

- identify regions of discrepancies

BACK UP

how to choose *K*? **Model Selection!**

True: $r(\boldsymbol{x}) = \dfrac{p_T(\boldsymbol{x})}{p_B(\boldsymbol{x})}$ Estimated: $\hat{r}(\boldsymbol{x}) = \dfrac{\hat{p}_T(\boldsymbol{x})}{\hat{p}_B(\boldsymbol{x})}$

Define the mean-square error:

$$\begin{aligned} L(r,\hat{r}) &= \frac{1}{2}\int \left[\hat{r}(\boldsymbol{x}') - r(\boldsymbol{x}')\right]^2 p_B(\boldsymbol{x}')d\boldsymbol{x}' \\ &= \frac{1}{2}\int \hat{r}(\boldsymbol{x}')^2 p_B(\boldsymbol{x}')d\boldsymbol{x}' - \int \hat{r}(\boldsymbol{x})p_T(\boldsymbol{x})d\boldsymbol{x} + \frac{1}{2}\int r(\boldsymbol{x}')^2 p_B(\boldsymbol{x}')d\boldsymbol{x}' \end{aligned}$$

Estimate loss: $\hat{L}(r,\hat{r}) = \dfrac{1}{2N_B}\displaystyle\sum_{\boldsymbol{x}'\in\mathcal{B}} \hat{r}(\boldsymbol{x}')^2 - \dfrac{1}{N_T}\sum_{\boldsymbol{x}\in\mathcal{T}} \hat{r}(\boldsymbol{x})$

Select optimal *K* minimizing the loss.

Alternatively: Point-Adaptive k-NN (PAk) [1802.10549]