XXV International Conference
Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL'2023),
24–27 October 2023, HSE University, Moscow, Russia

Discord discovery in time series,

or

Can we detect all anomalies
of an anomalously long time series
in an anomalously short time?

Mikhail Zymbler, Yana Kraeva

{mzym, kraevaya}@susu.ru

Big Data and Machine Learning Lab, South Ural State University, Chelyabinsk, Russia

This work was financially supported by the Russian Science Foundation (grant No. 23-21-00465)

Disclaimer*

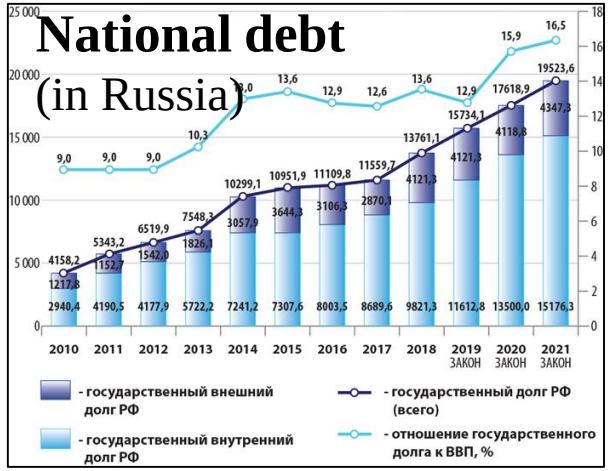
- This talk **does not touch deep learning** anomaly detection
- The authors acknowledge that deep learning plays an extremely important role in modern **DAMDID** in general and in **time series** analytics in particular
- However, the authors believe that in time series anomaly detection, a **simple math** is ***at least*** competitive with deep learning approaches

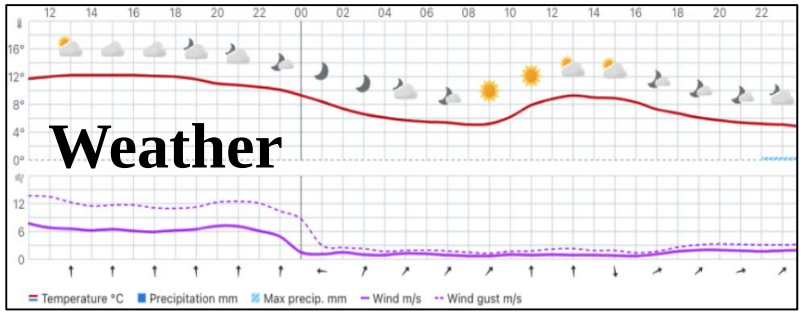
* During our research, neither a deep learning model nor a deep learning scientist were injured

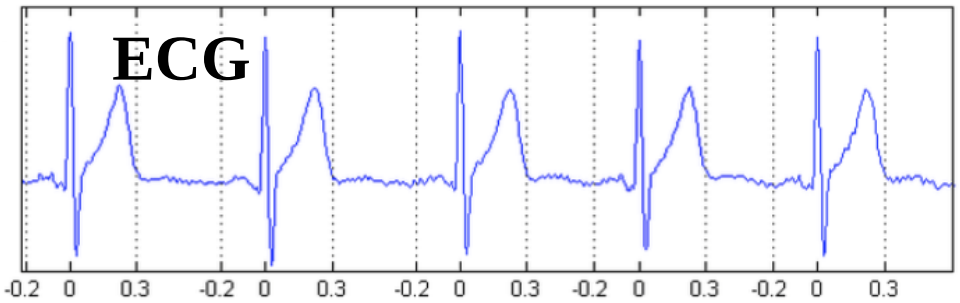
People measure everything over time in all domains

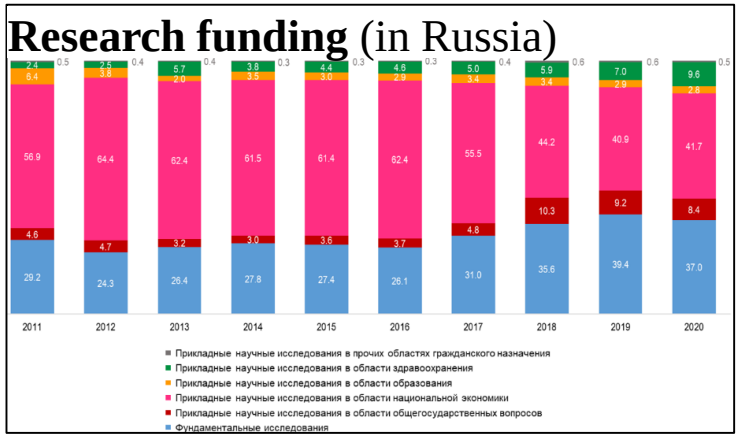
In random sample of 4000 images
in 15 newspapers from different countries
for 1974-1989,
more than 75% of images are time series*

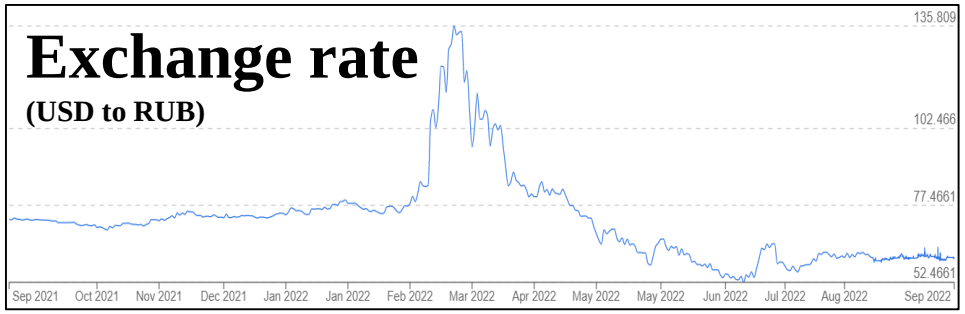
Tufte E. The visual display of quantitative information. Graphics Press, 2001. 200 p. Amazon

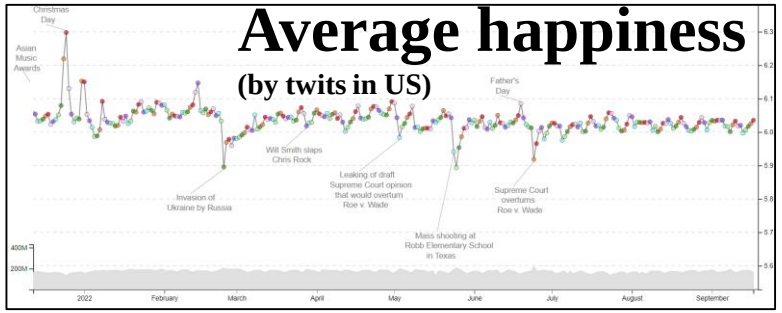






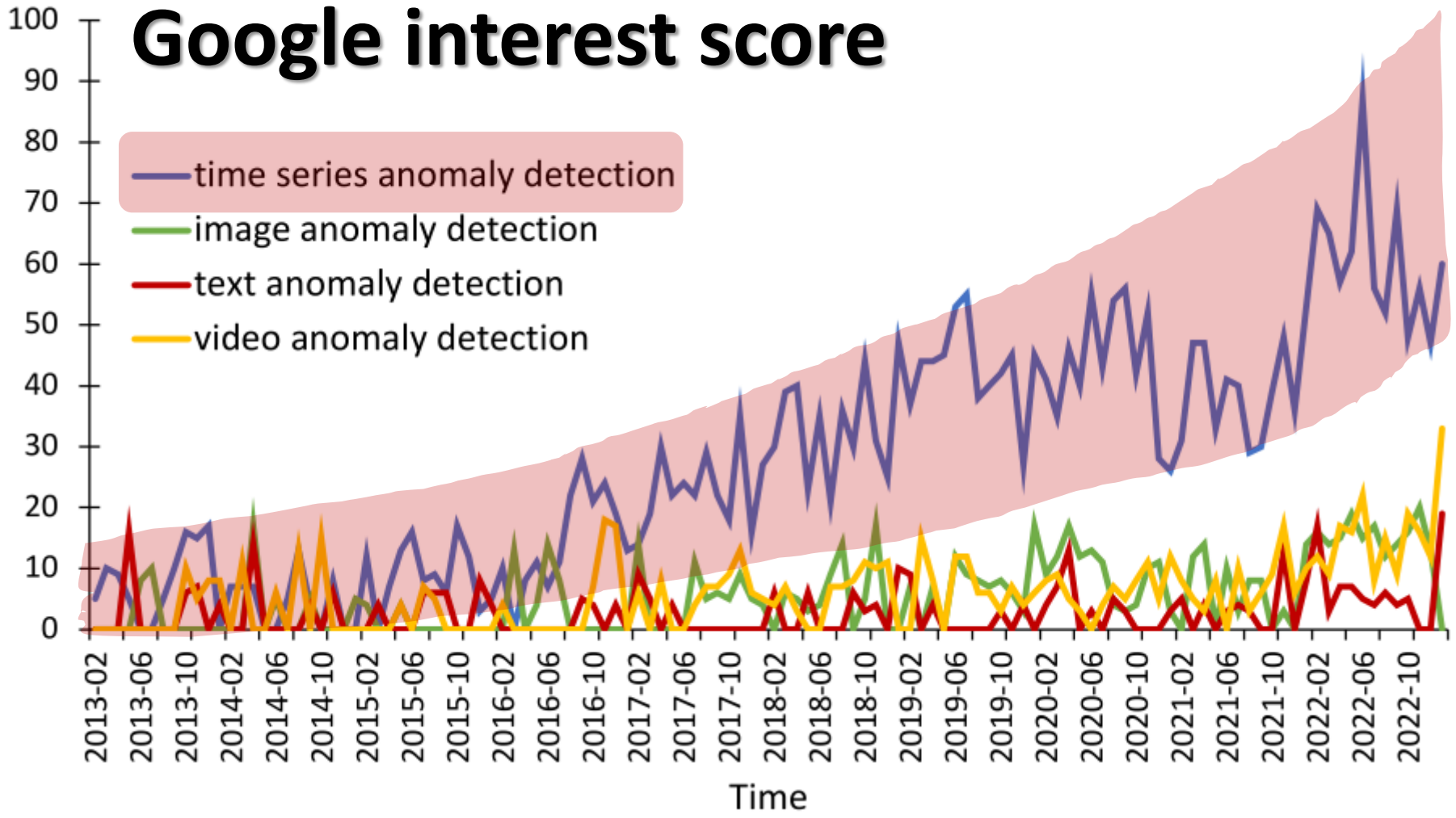






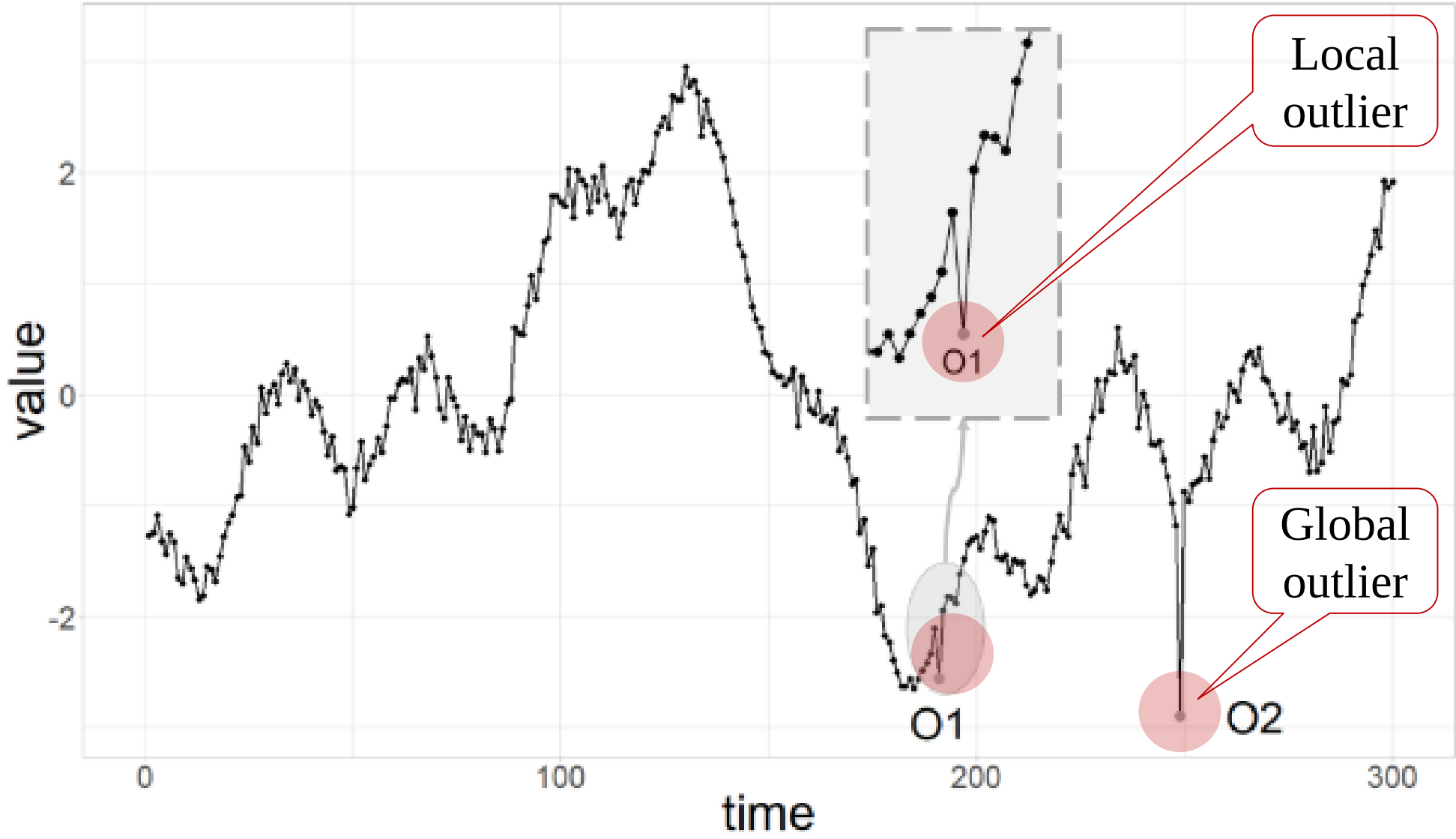
Your time series can be here

In time series, people are most interested in **anomalies**

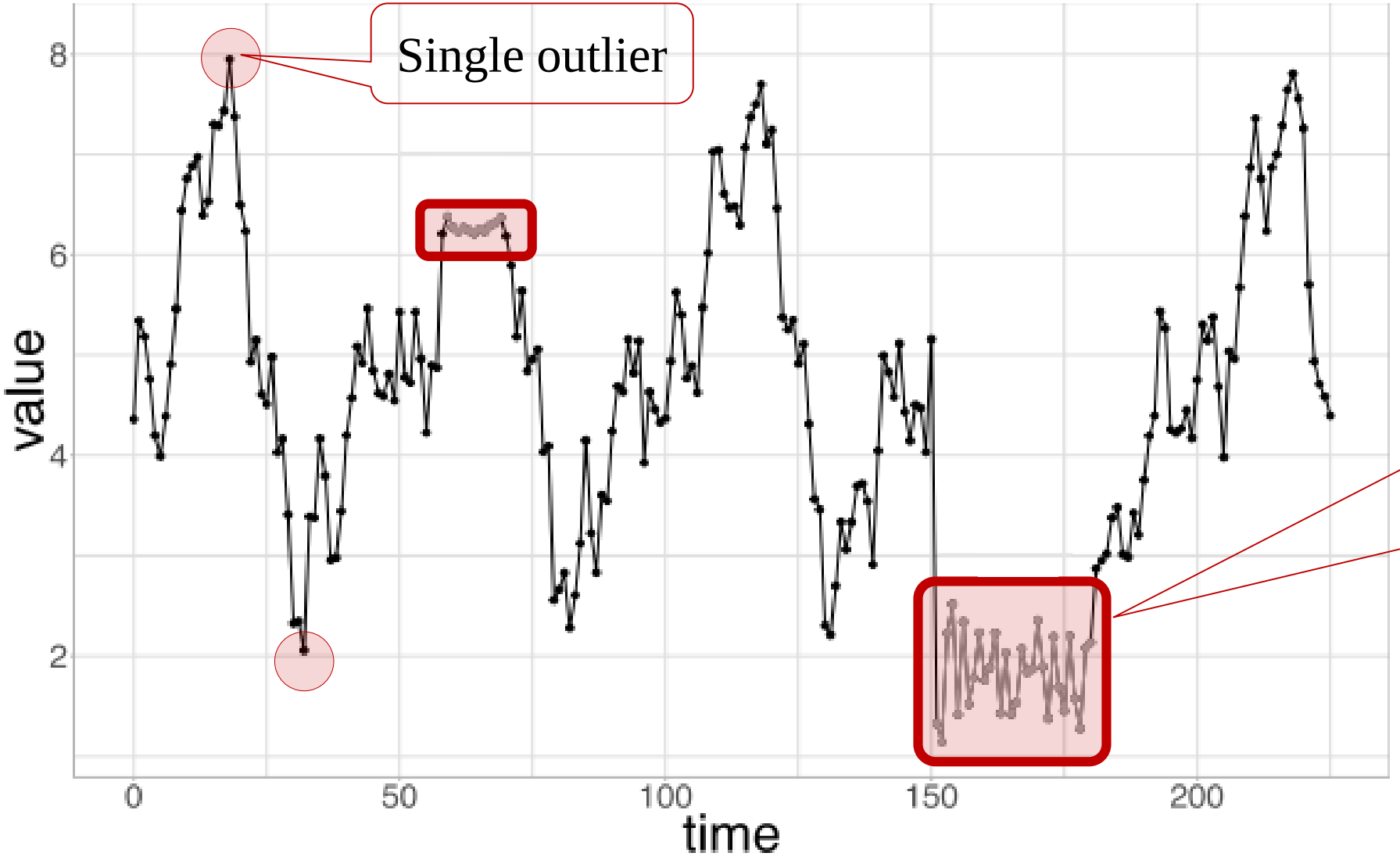


* Boniol P. *et al.* New trends in time-series anomaly detection. EDBT'2023. pp. 847-850. DOI: 10.48786/edbt.2023.80

Q: What anomalies we're interested in? ***A:*** **Not** peaks/valleys



Subsequence anomalies is the challenge



Collective behavior
of points
is anomalous,
but a point
may not be an outlier

Anomalies are informal and domain-dependent


Homer


Marge


Bart


Selma


Patty

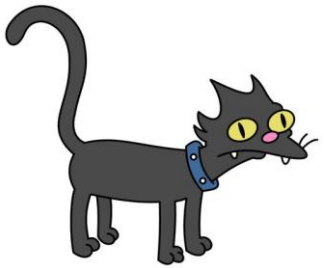

Barney


Quasimodo

Anomaly is an observation, which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism.

Hawkins D.M. Identification of outliers. Monographs on applied probability and statistics. Springer, 1980. DOI: 10.1007/978-94-015-3994-4.


Itchy

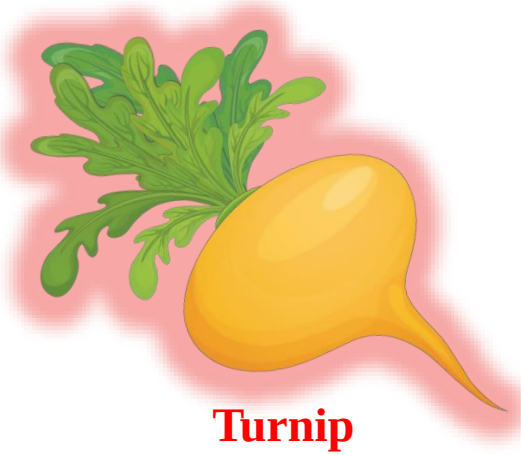

Scratchy


Santa's Little Helper


Lisa


Mona


Abe

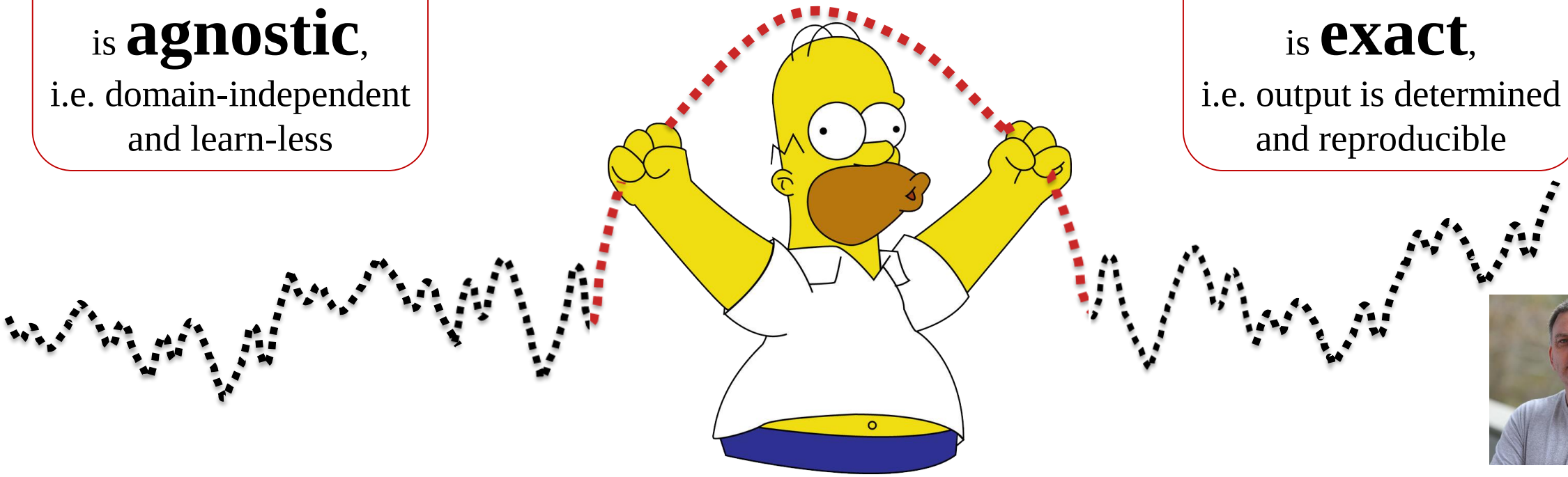

Turnip

Discord formalizes subsequence anomaly

Discord* is a subsequence of the given length whose distance to its nearest neighbor is greatest

This concept is **agnostic**, i.e. domain-independent and learn-less

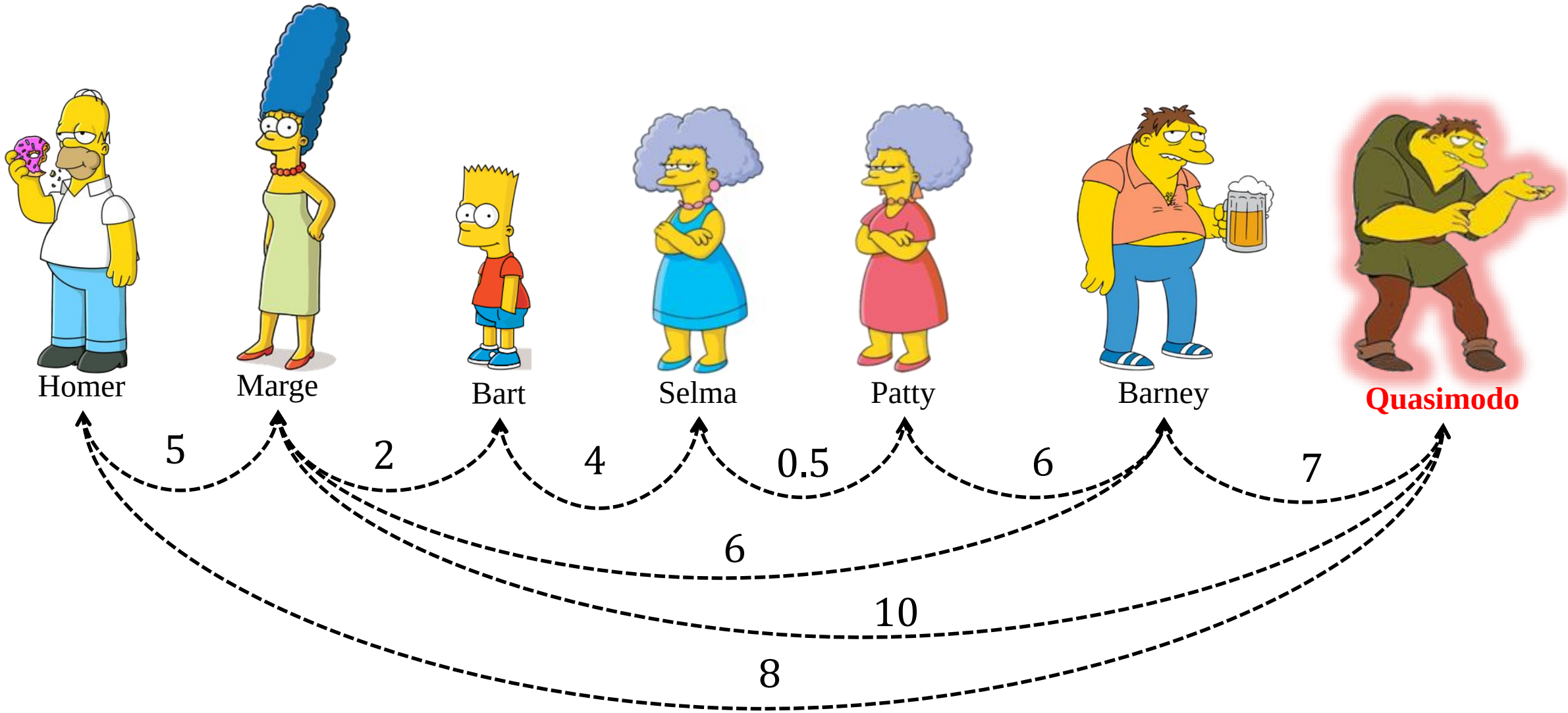
This concept is **exact**, i.e. output is determined and reproducible



Eamonn Keogh

* Keogh E. *et al.* HOT SAX: Efficiently finding the most unusual time series subsequence. ICDM 2005. pp. 226-233. DOI: 10.1109/ICDM.2005.79

Discord concept: Distance reflects similarity



Discord concept

Distance matrix:
the close neighbors,
the similar they are

Homer	Marge	Bart	Selma	Patty	Barney	Quasimodo
0						
	0					
		0				
			0			
				0		
					0	
						0

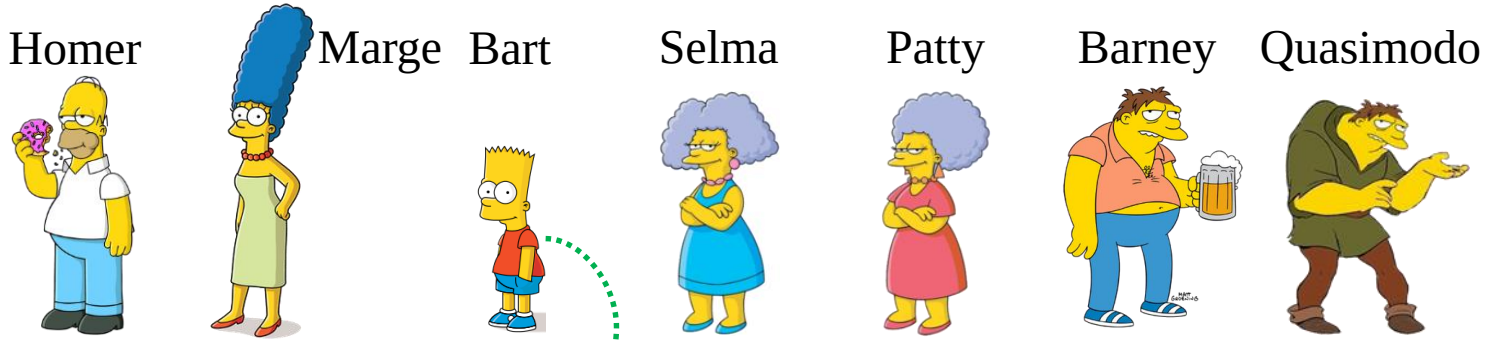
Discord concept

Distance matrix
with calculated distances
to neighbors

Homer	Marge	Bart	Selma	Patty	Barney	Quasimodo
0	5	2	4	4	6	8
5	0	2.5	3	3	6	10
2	2.5	0	4	4	6	9
4	3	4	0	0.5	5	8
4	3	4	0.5	0	5	8
6	6	6	5	5	0	7
8	10	9	8	8	7	0

Discord concept

Distance matrix
with
distances to their nearest neighbors
(i.e. column-wise minima)



0	5	**2**	4	4	6	8
5	0	2.5	3	3	6	10
2	**2.5**	0	4	4	6	9
4	3	4	0	**0.5**	**5**	8
4	3	4	**0.5**	0	**5**	8
6	6	6	5	5	0	**7**
8	10	9	8	8	7	0

Discord concept

Distance matrix
with the
farthest distance
to the nearest neighbor
(i.e. maximum
among
column-wise minima)

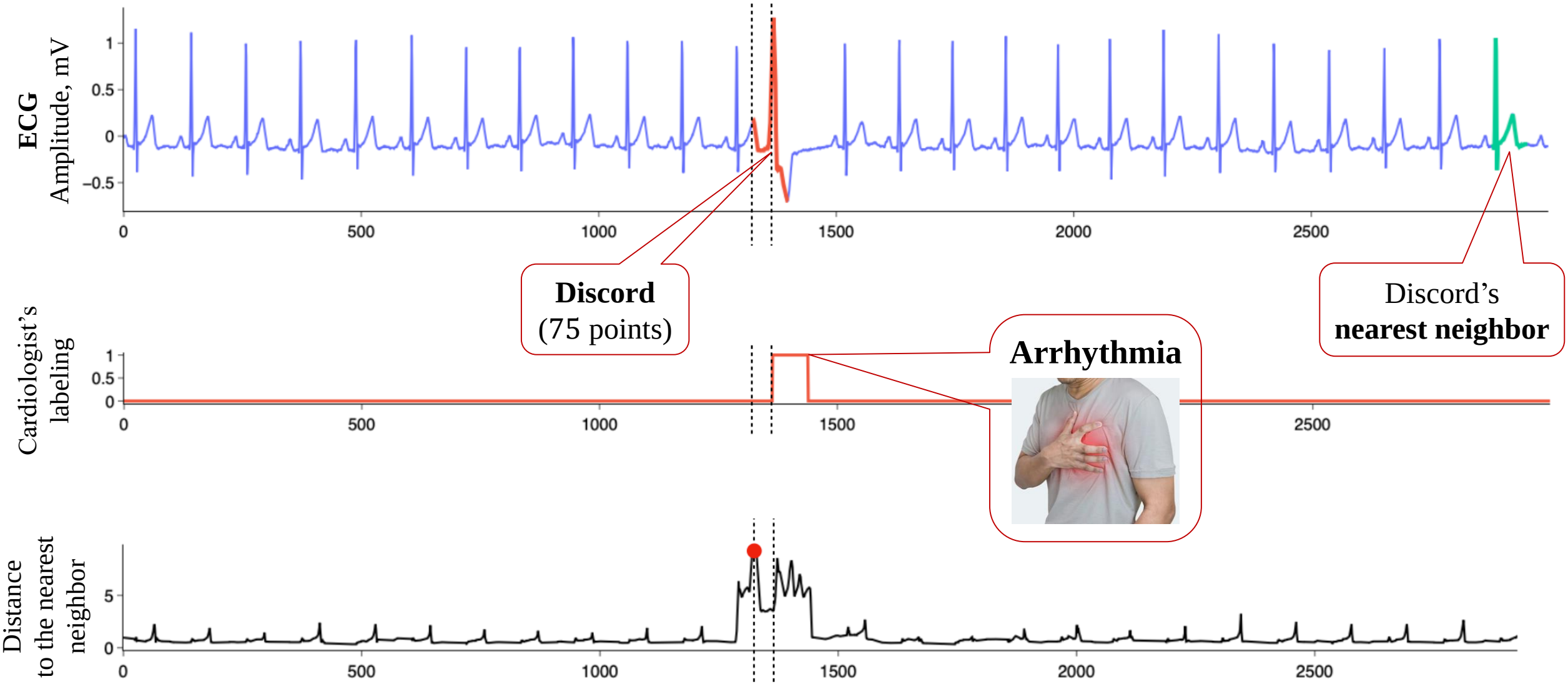
Homer	Marge	Bart	Selma	Patty	Barney	Quasimodo
0	5	2	4	4	6	8
5	0	2.5	3	3	6	10
2	2.5	0	4	4	6	9
4	3	4	0	0.5	5	8
4	3	4	0.5	0	5	8
6	6	6	5	5	0	**7**
8	10	9	8	8	7	0

Discord concept

Discord is an object with the **farthest nearest neighbor** (i.e. argument of the maximum among column-wise minima)

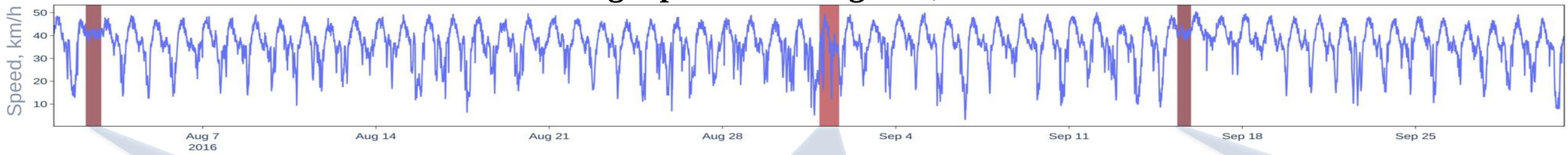
Homer	Marge	Bart	Selma	Patty	Barney	**Quasimodo**
0	5	2	4	4	6	8
5	0	2.5	3	3	6	10
2	2.5	0	4	4	6	9
4	3	4	0	0.5	5	8
4	3	4	0.5	0	5	8
6	6	6	5	5	0	**7**
8	10	9	8	8	7	0

Discords are not always identical to anomalies, …

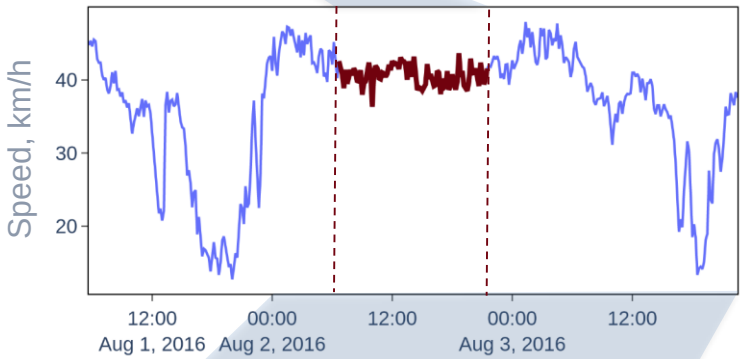


… but discords always reflect anomalies in real life

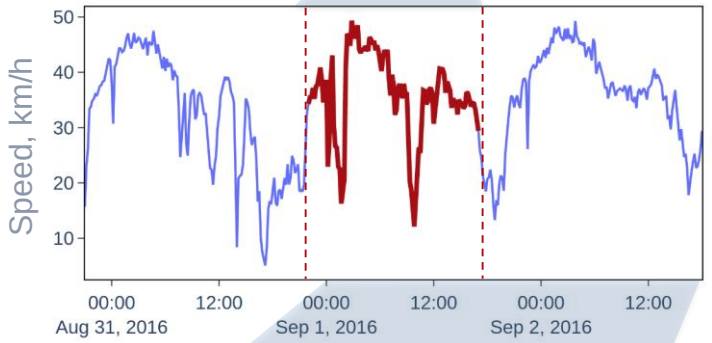
Average speed in Guangzhou, China*



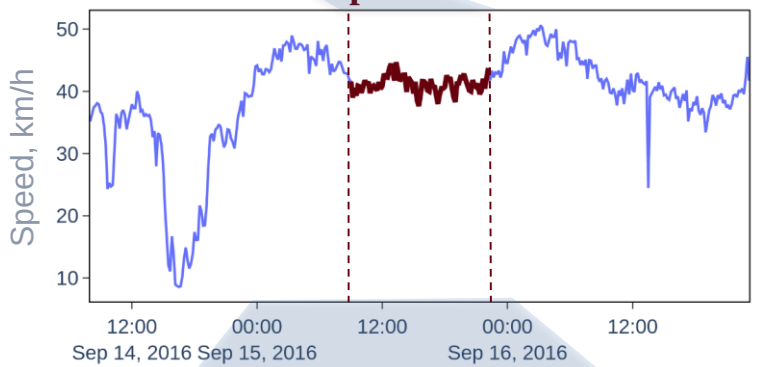
Top-2 discord



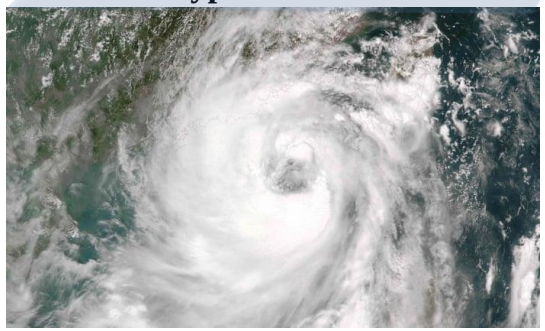
Top-3 discord



Top-1 discord



Typhoon Nida

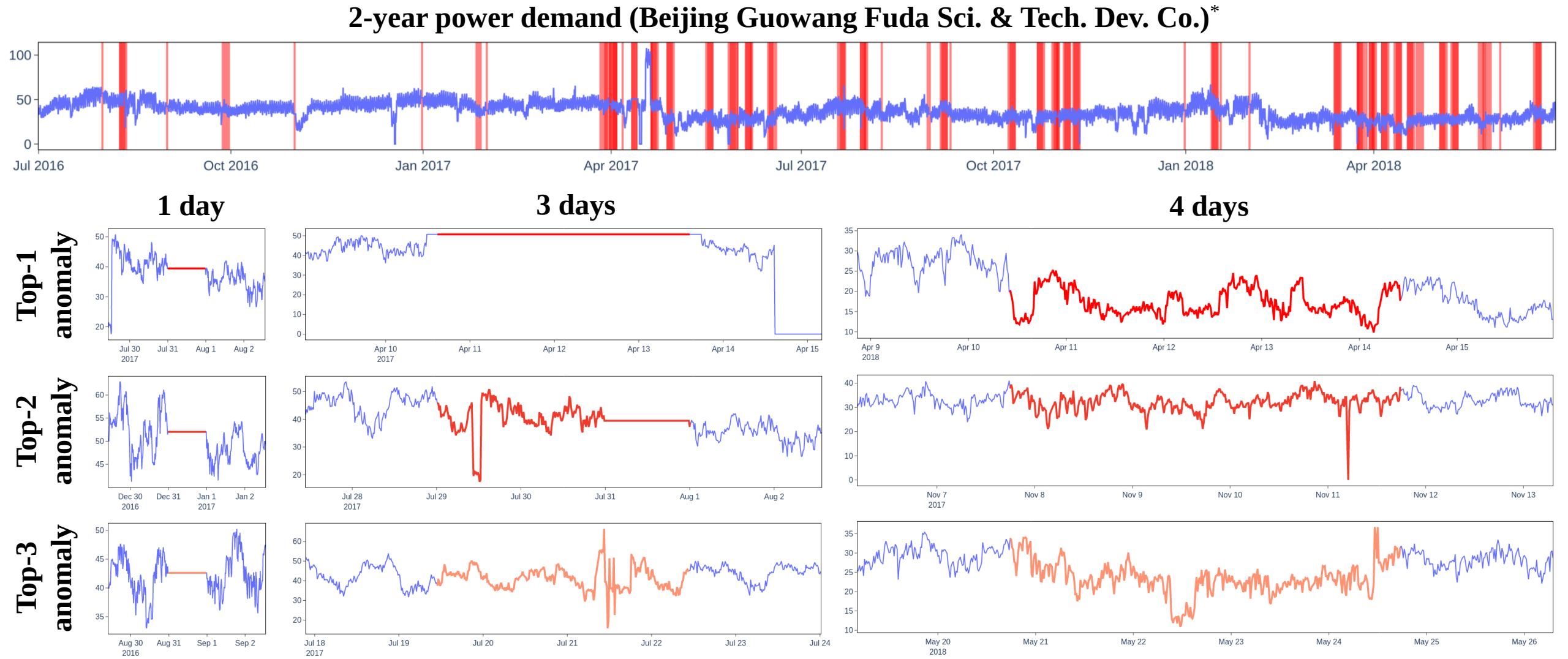


Day of Victory over Japan

Mid-Autumn Festival

* Chen X, Chen Y, He Z. Urban traffic speed dataset of Guangzhou, China. 2018. DOI: 10.5281/zenodo.1205229.

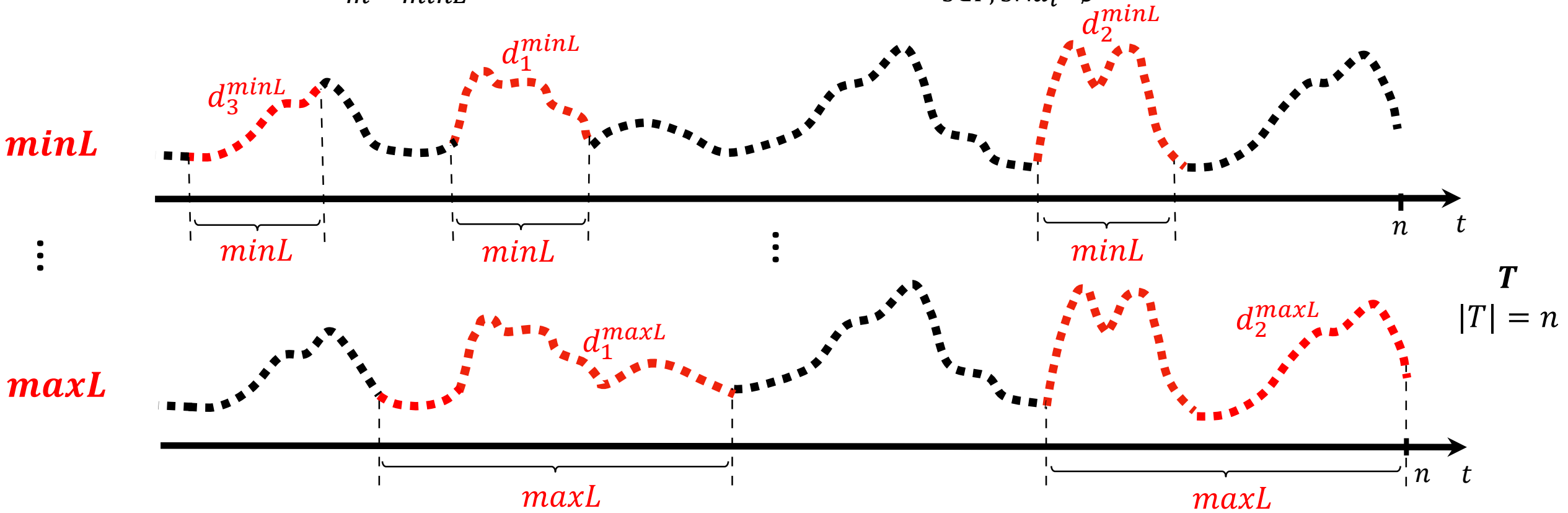
For long series, we need **all-length** subsequence anomalies!



* Zhou H. *et al.* Informer: beyond efficient transformer for long sequence time-series forecasting. AAAI 2021: 11106-11115. DOI: 10.1609/aaai.v35i12.17325.

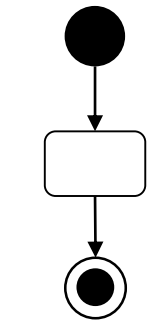
To discover all-length anomalies, we need **range discords**

- **Range discord*** is a discord that is at least a **threshold** r away from its nearest neighbor
- **We are given:** time series T, **range of discord length** $minL \ldots maxL$
- **We are to find:** $\mathcal{D} = \bigcup_{m=minL}^{maxL} D_m,\ D_m = \{d_1^m, d_2^m, \ldots\}$, where $\min_{s \in T,\ s \cap d_i = \emptyset} \text{dist}(d_i^m, s) \geq r$



* Yankov D. *et al.* Disk aware discord discovery: finding unusual time series in terabyte sized datasets. ICDM 2007. pp. 381-390. DOI: 10.1109/ICDM.2007.61.

Time series discord discovery algorithms: A roadmap

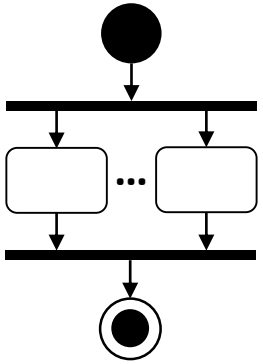


Serial algorithms

Discord[1] (2005)

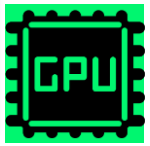
DRAG[2] (2007)

MERLIN[3] (2020)

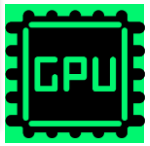


Parallel algorithms

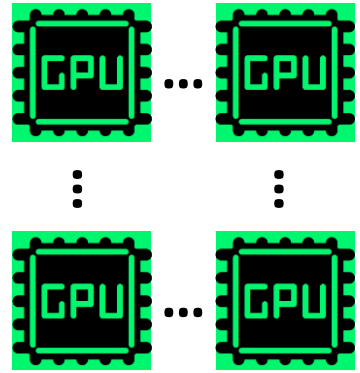
PD3[4] (2022)



PALMAD[5] (2023)



PADDi[6] (2023)



[1] Keogh E. *et al.* HOT SAX: Efficiently finding the most unusual time series subsequence. ICDM 2005. pp. 226-233. DOI: 10.1109/ICDM.2005.79

[2] Yankov D. *et al.* Disk aware discord discovery: finding unusual time series in terabyte sized datasets. ICDM 2007. pp. 381-390. DOI: 10.1109/ICDM.2007.61.

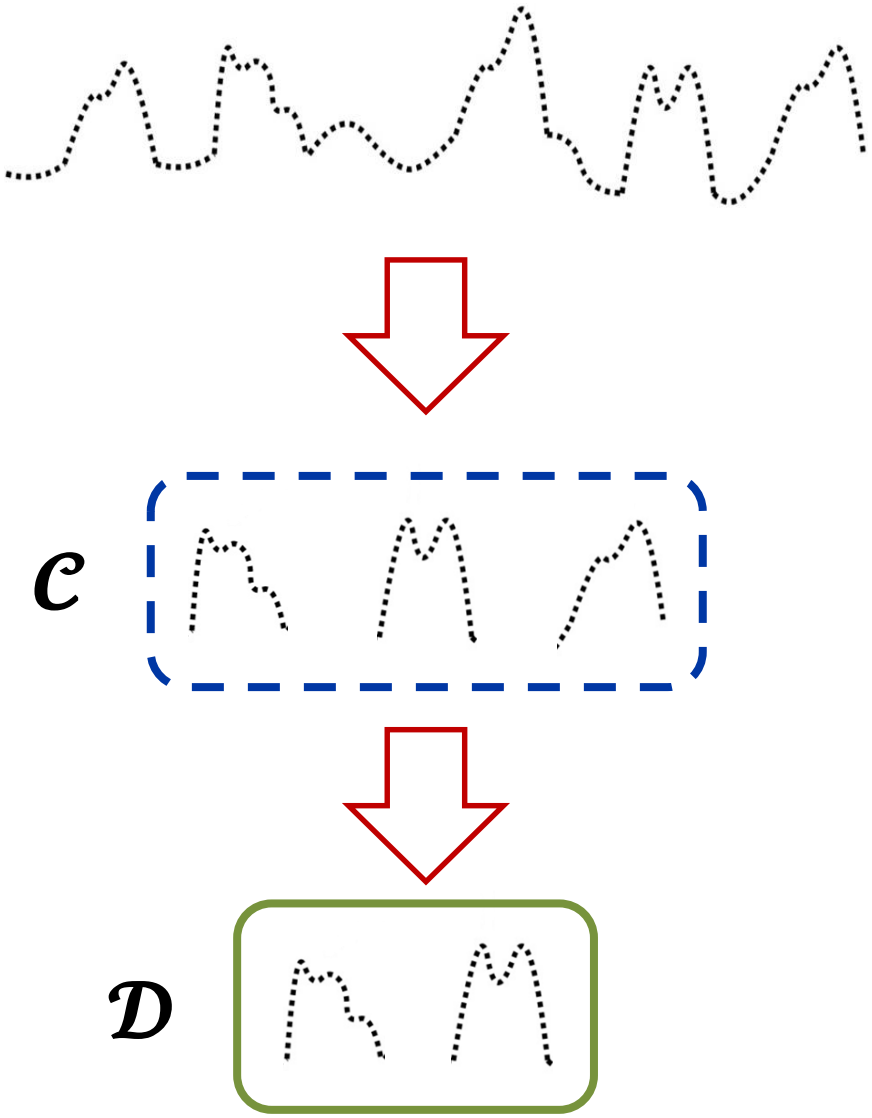
[3] Nakamura T. *et al.* MERLIN: parameter-free discovery of arbitrary length anomalies in massive time series archives. ICDM 2020. pp. 1190-1195. DOI: 10.1109/ICDM50108.2020.00147.

[4] Kraeva Y., Zymbler M. A parallel discord discovery algorithm for a graphics processor. PRIA. 2023. 33(2). pp. 101-112. DOI: 10.1134/S1054661823020062.

[5] Zymbler M., Kraeva Y. High-performance time series anomaly discovery on graphics processors. Mathematics. 2023. 11(14). art. 3193. DOI: 10.3390/math11143193.

[6] Kraeva Y., Zymbler M. Anomaly detection in long time series on high-performance cluster with GPUs. NM&P. 2023. 24(3). pp. 291-304. DOI: 10.26089/NumMet.v24r320.

Discovery of fixed-length discords (DRAG algorithm)



1. Selection

Through one full scan of the time series, create a **set of candidates** to discords

2. Refinement

Through one full scan of the time series, **prune false positive** candidates

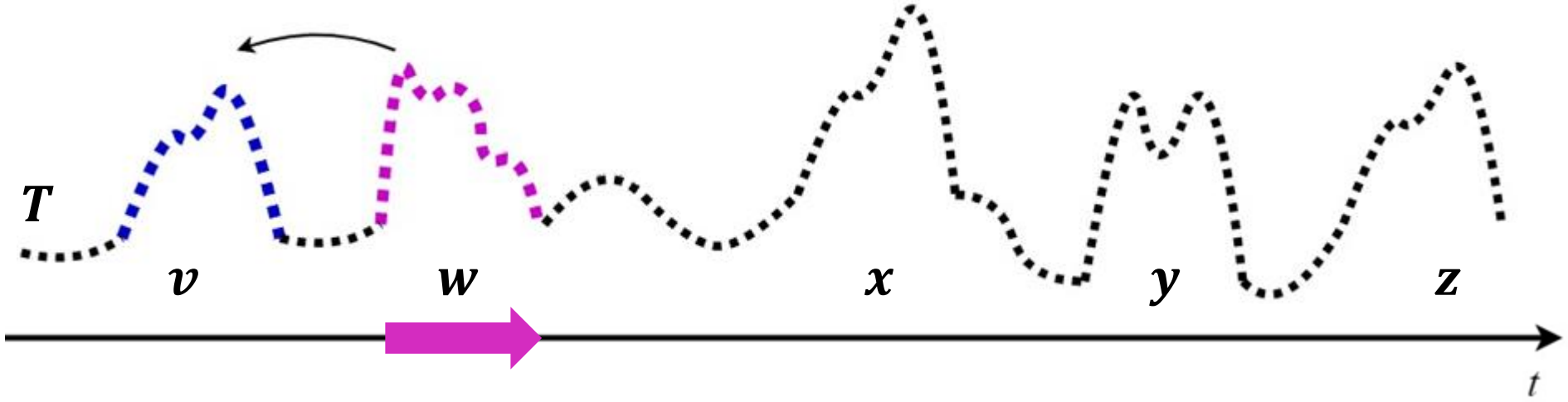
Candidate selection

$\mathcal{C} := \{T_{1,m}\}$
while not end of $\boldsymbol{T}$
    get next subsequence $\boldsymbol{s}$
    $isCandidate :=$ TRUE
    for each $\boldsymbol{c_i} \in \mathcal{C}$ **and** $\boldsymbol{s} \cap \boldsymbol{c_i} = \emptyset$
        if $\text{dist}(\boldsymbol{s}, \boldsymbol{c_i}) < r$ **then**
            $\mathcal{C} := \mathcal{C} \setminus \boldsymbol{c_i}$; $isCandidate :=$ FALSE
    if $isCandidate =$ TRUE **then** $\mathcal{C} := \mathcal{C} \cup \boldsymbol{s}$

$\mathcal{C} = \{\boldsymbol{v}\}$

$\text{dist}(\boldsymbol{w}, \boldsymbol{v}) \geq r$

$\mathcal{C} = \{\boldsymbol{v}, \boldsymbol{w}\}$



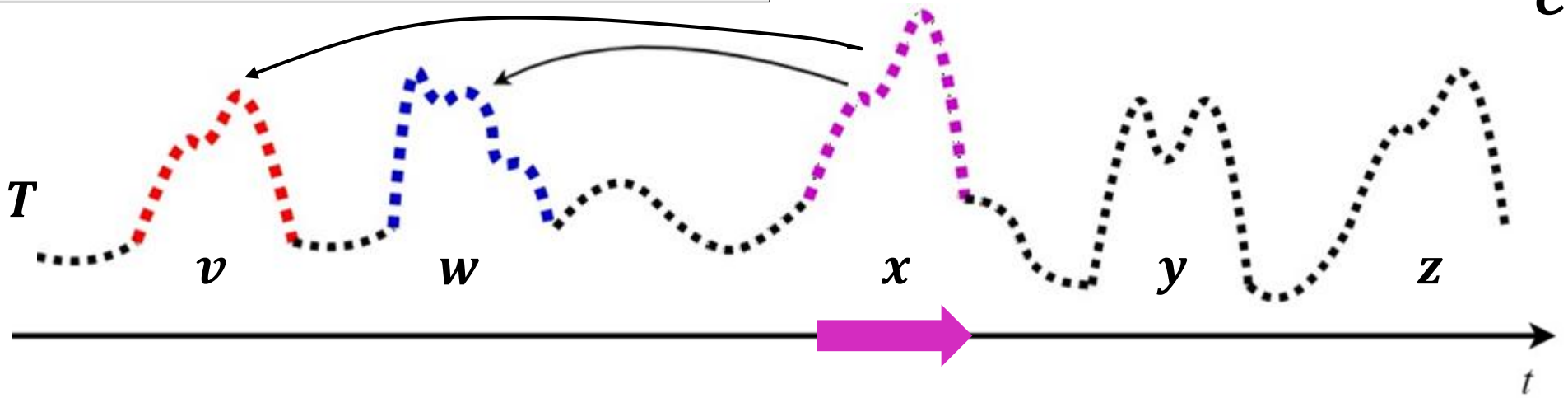
Candidate selection

$\mathcal{C} := \{T_{1,m}\}$
while not end of $\boldsymbol{T}$
    get next subsequence $\boldsymbol{s}$
    $isCandidate :=$ TRUE
    for each $\boldsymbol{c_i} \in \mathcal{C}$ **and** $\boldsymbol{s} \cap \boldsymbol{c_i} = \emptyset$
        if $\text{dist}(\boldsymbol{s}, \boldsymbol{c_i}) < r$ **then**
            $\mathcal{C} := \mathcal{C} \setminus \boldsymbol{c_i}$; $isCandidate :=$ FALSE
    if $isCandidate =$ TRUE **then** $\mathcal{C} := \mathcal{C} \cup \boldsymbol{s}$

$\mathcal{C} = \{\boldsymbol{v}, \boldsymbol{w}\}$

$\text{dist}(\boldsymbol{x}, \boldsymbol{v}) < r$
$\text{dist}(\boldsymbol{x}, \boldsymbol{w}) \geq r$

$\mathcal{C} = \{\boldsymbol{w}\}$



Candidate selection

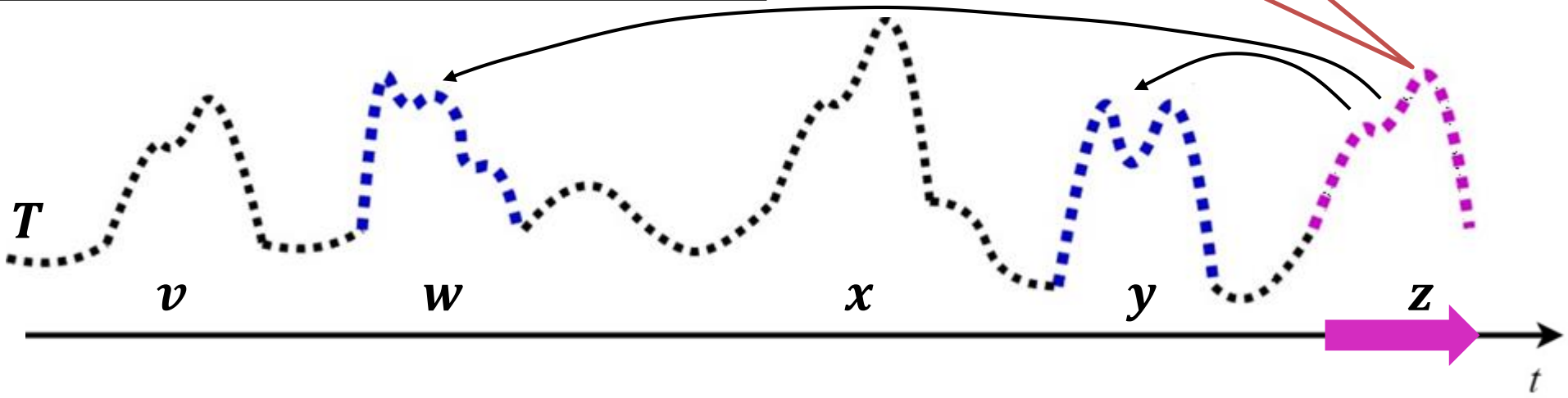
$\mathcal{C} := \{T_{1,m}\}$
while not end of $\boldsymbol{T}$
  get next subsequence $\boldsymbol{s}$
  $isCandidate :=$ TRUE
  for each $\boldsymbol{c_i} \in \mathcal{C}$ **and** $\boldsymbol{s} \cap \boldsymbol{c_i} = \emptyset$
    if $\text{dist}(\boldsymbol{s}, \boldsymbol{c_i}) < r$ **then**
      $\mathcal{C} := \mathcal{C} \setminus \boldsymbol{c_i}$; $isCandidate :=$ FALSE
  if $isCandidate =$ TRUE **then** $\mathcal{C} := \mathcal{C} \cup \boldsymbol{s}$

z is false positive since $\text{dist}(z, v) < r$ $\text{dist}(z, x) < r$. But v and x were excluded

$\mathcal{C} = \{w, y\}$

$\text{dist}(z, w) \geq r$
$\text{dist}(z, y) \geq r$

$\mathcal{C} = \{w, y, z\}$



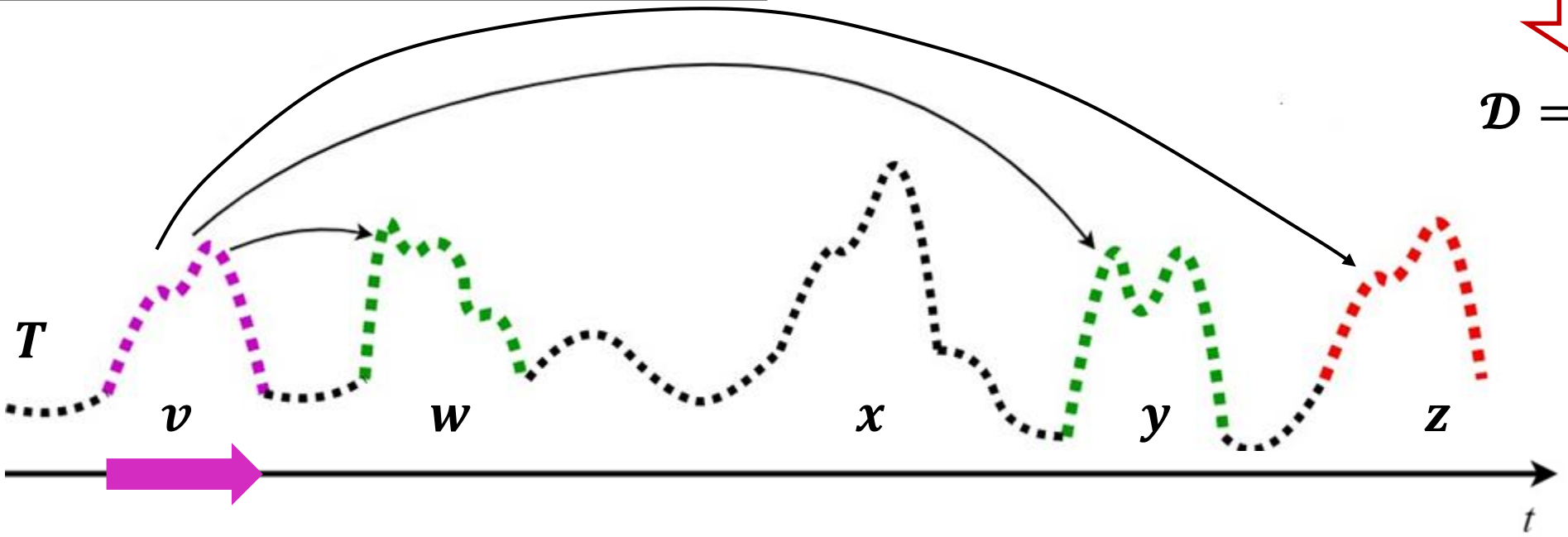
Discord refinement

$\mathcal{D} := \mathcal{C}$
while not end of T
    get next subsequence s
    for each $d_i \in \mathcal{D}$ **and** $s \cap d_i = \emptyset$
        if $\text{dist}(s, d_i) < r$ **then**
            $\mathcal{D} := \mathcal{D} \setminus d_i$

$\mathcal{D} = \{w, y, z\}$

$\text{dist}(v, w) \geq r$
$\text{dist}(v, y) \geq r$
$\text{dist}(v, z) < r$

$\mathcal{D} = \{w, y\}$



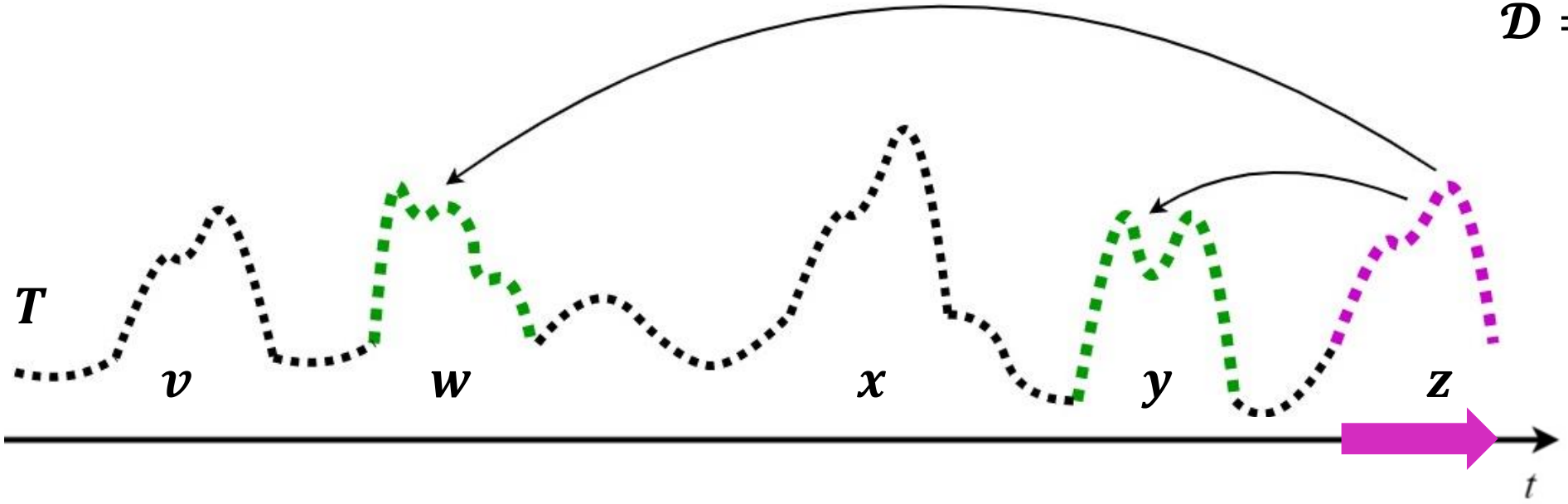
Discord refinement

$\mathcal{D} := \mathcal{C}$
while not end of T
    get next subsequence s
    for each $d_i \in \mathcal{D}$ **and** $s \cap d_i = \emptyset$
        if $\text{dist}(s, d_i) < r$ **then**
            $\mathcal{D} := \mathcal{D} \setminus d_i$

$\mathcal{D} = \{w, y\}$

$\text{dist}(z, w) \geq r$
$\text{dist}(z, y) \geq r$

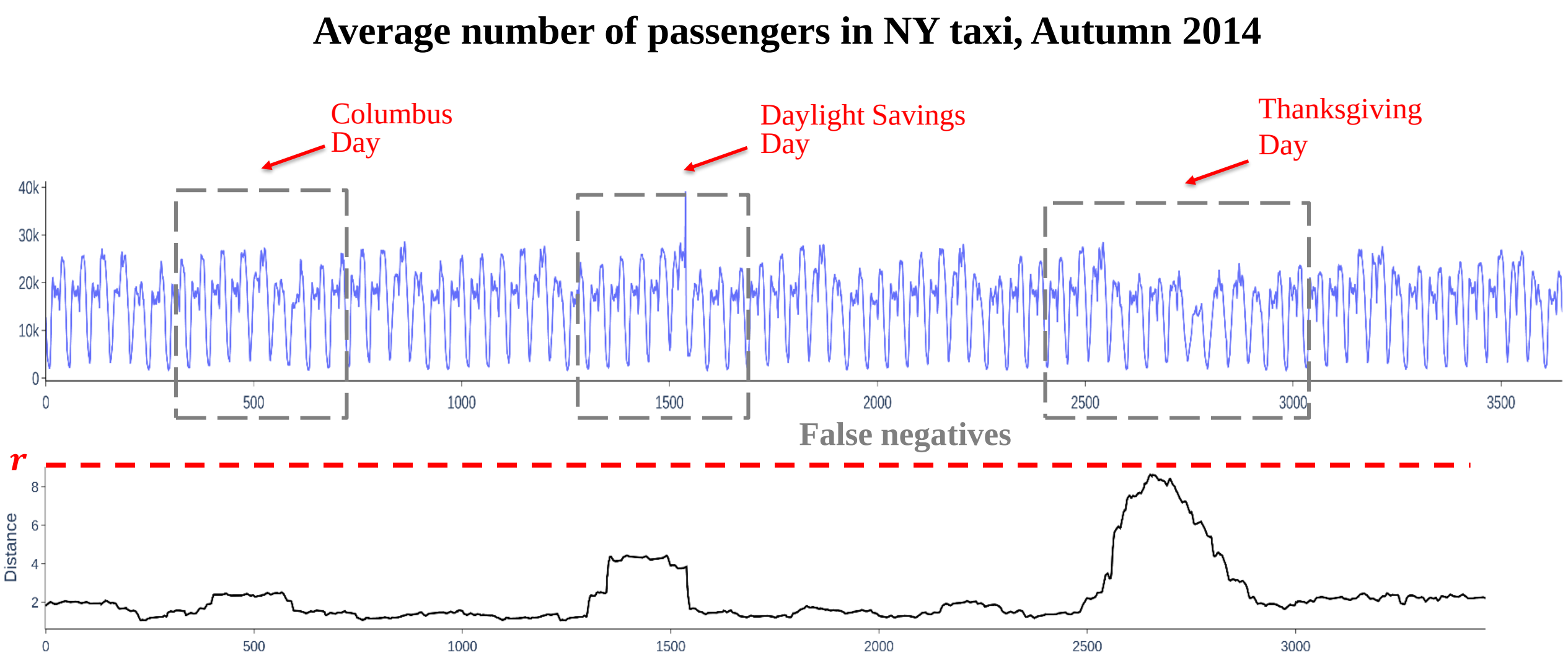
$\mathcal{D} = \{w, y\}$



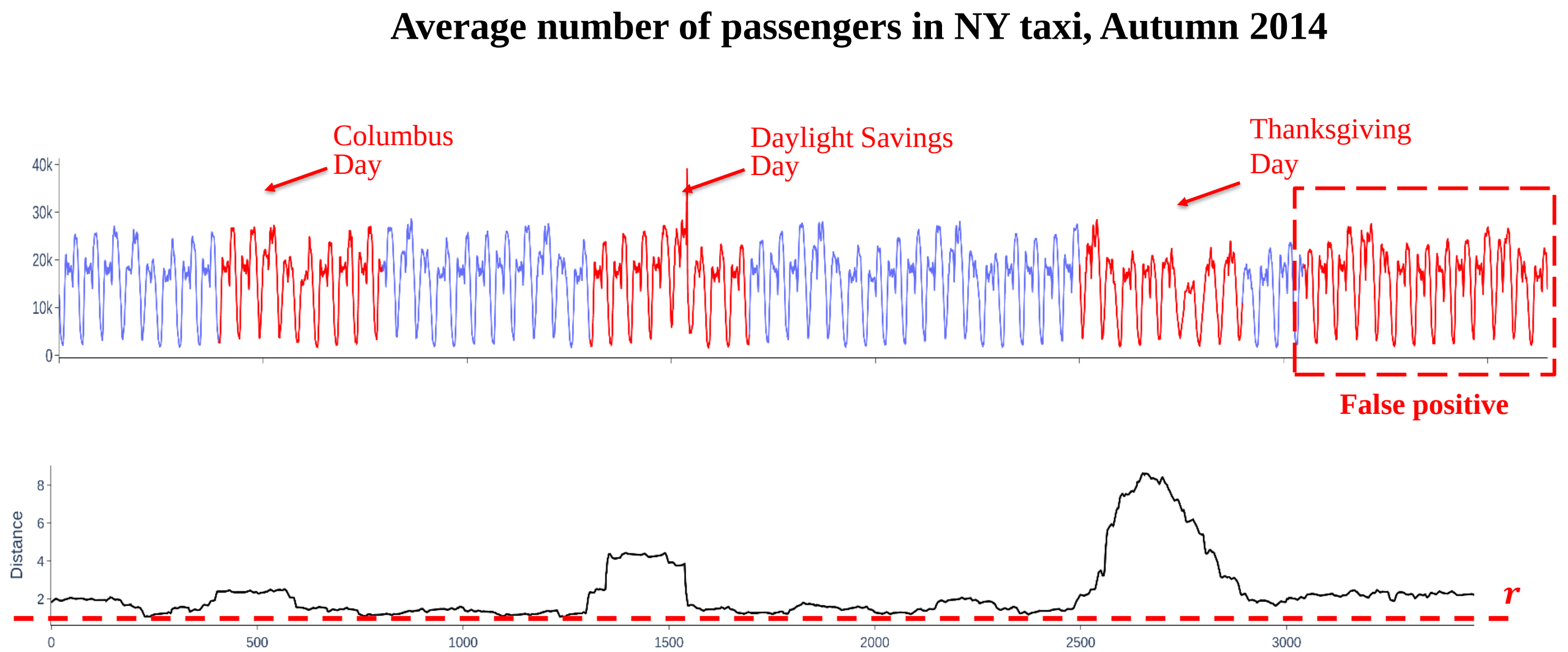
Q: What's wrong with DRAG? ***A:*** Nothing, it's perfect, but…

- We have to select the threshold r manually
- We have to select the discord length m manually

Manual selection: if r too big, we miss discords

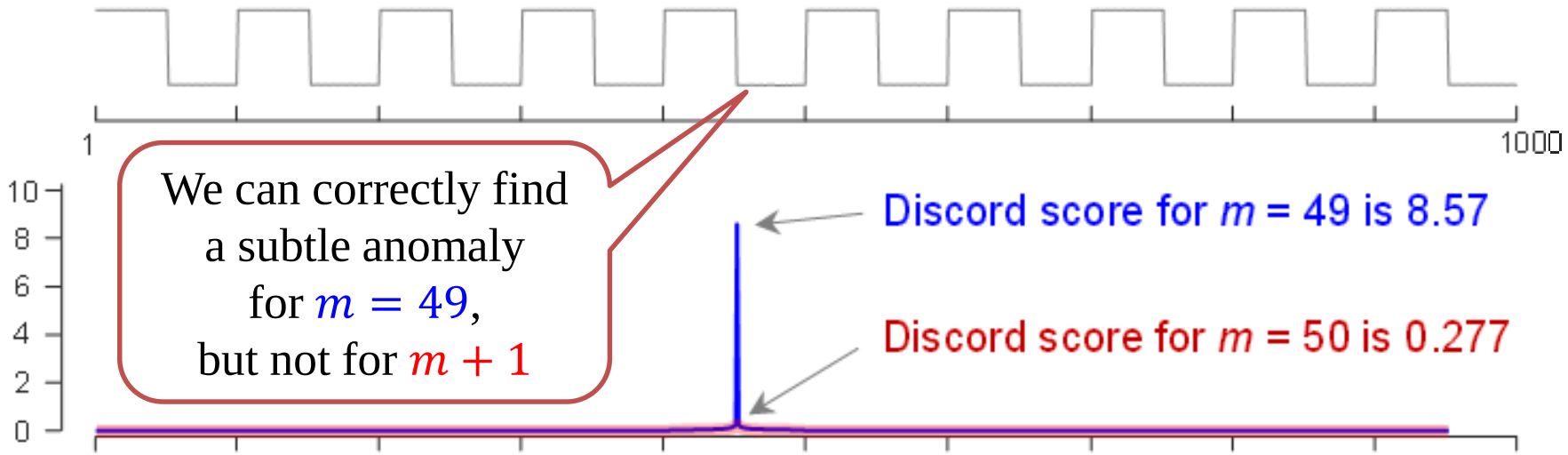


Manual selection: if r too small, we got false discords

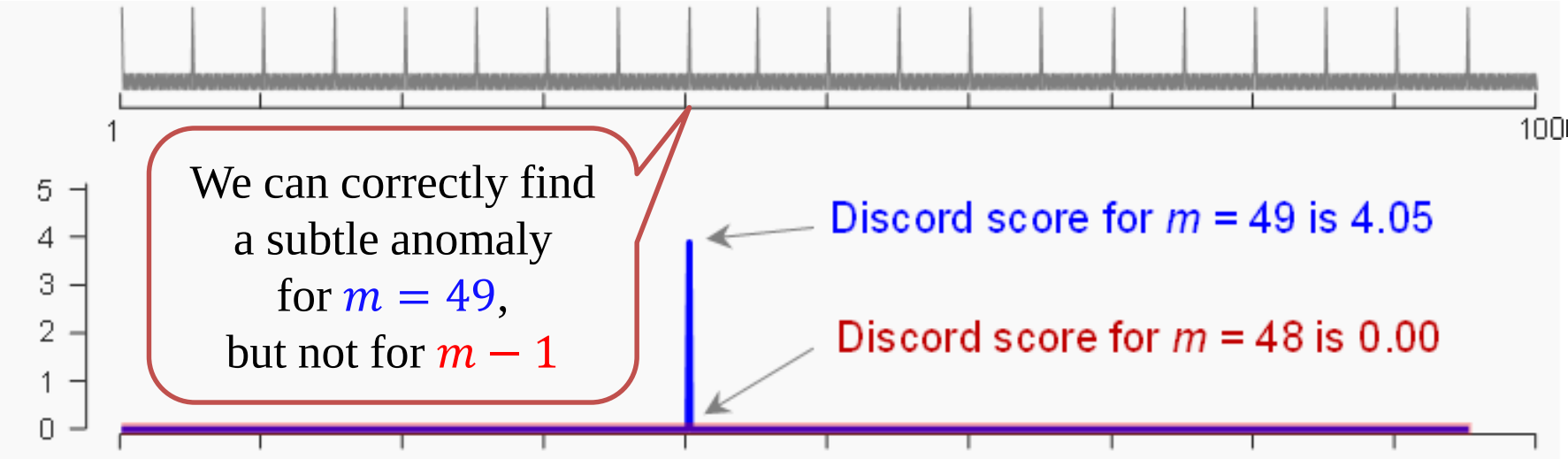


To discover ***all*** discords, we must test ***all*** m values

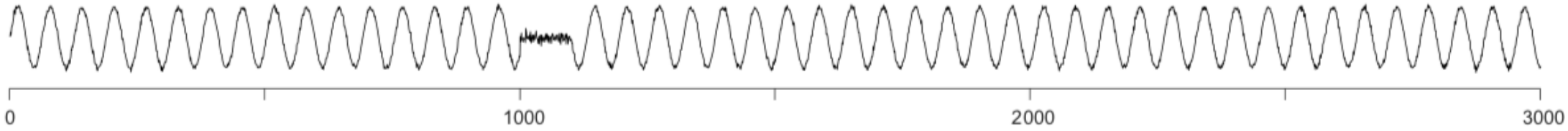
```
// Generate time series
for i ← 1 to 1000 do
    T_i ← i mod 2
end for
for i ← 1 to 1000 step 100 do
    T_{i:i+50} ← T_{i:i+50} + 100
end for
// Generate discord
T_{453:499} ← rand()/20
```

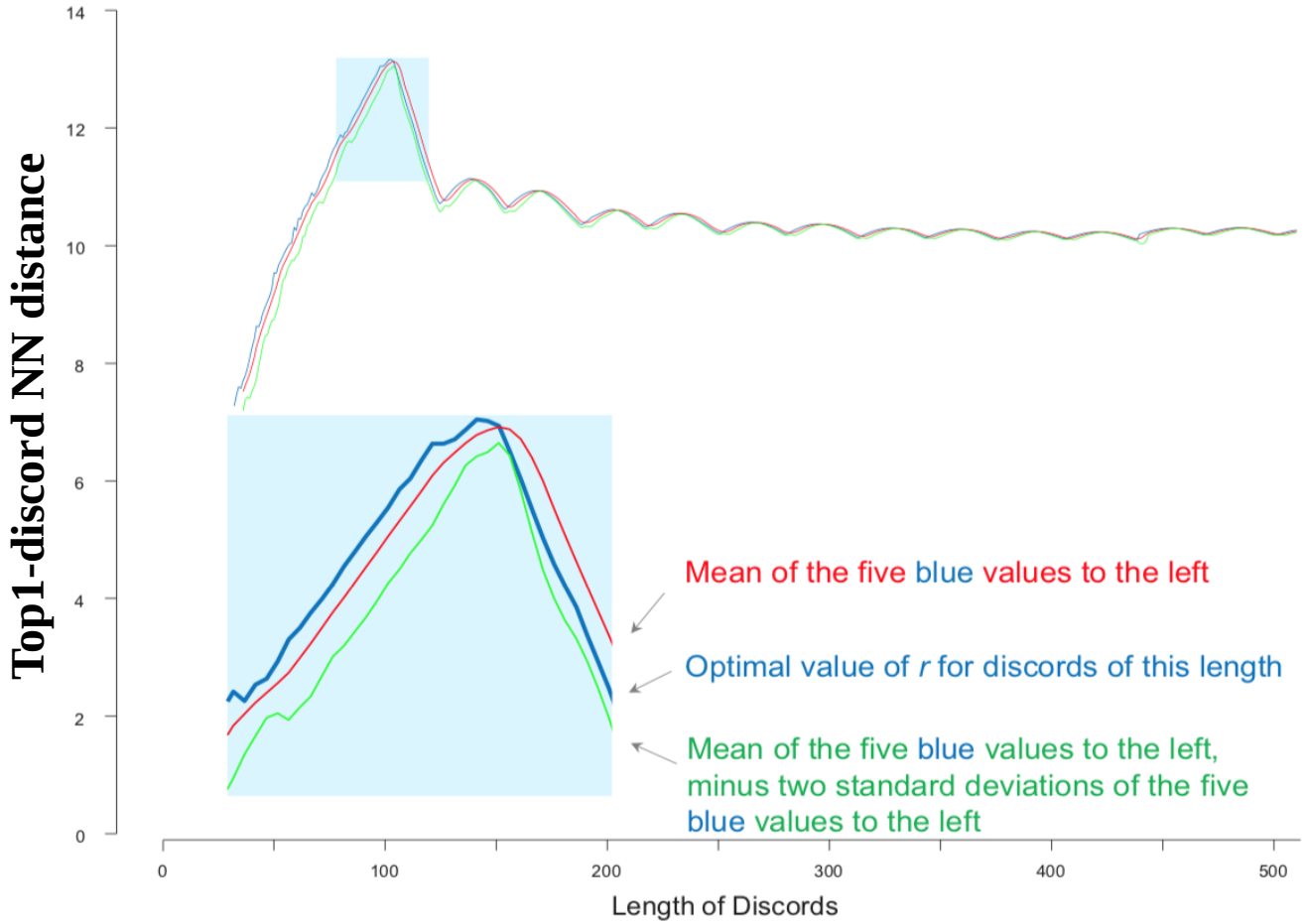



```
// Generate time series
for i ← 1 to 1000 do
    if i mod 50 ≠ 0 then
        T_i ← i mod 2
    else
        T_i ← 5
    end if
end for
// Generate discord
T_{430:431} ← 0
```



MERLIN algorithm: r is adaptively calculated

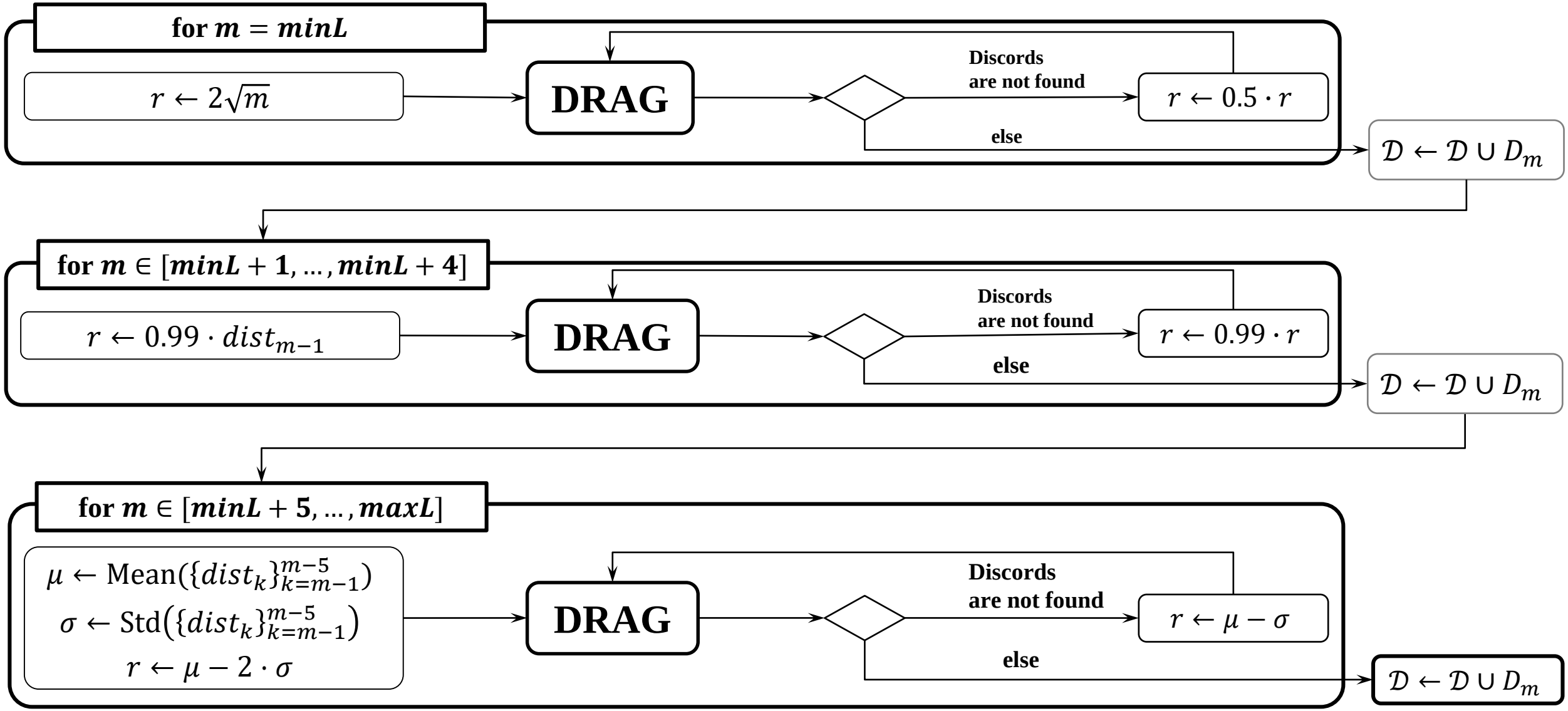




The threshold r must be calculated differently for different lengths m

Discord length, m	Threshold, r
$minL$	$r = 2\sqrt{minL}$
$minL + 1, \ldots, minL + 4$	$r = 0.99 \cdot nnDist_{m-1}$
$minL + 5, \ldots, maxL$	$r = \mu - 2\sigma$

MERLIN algorithm

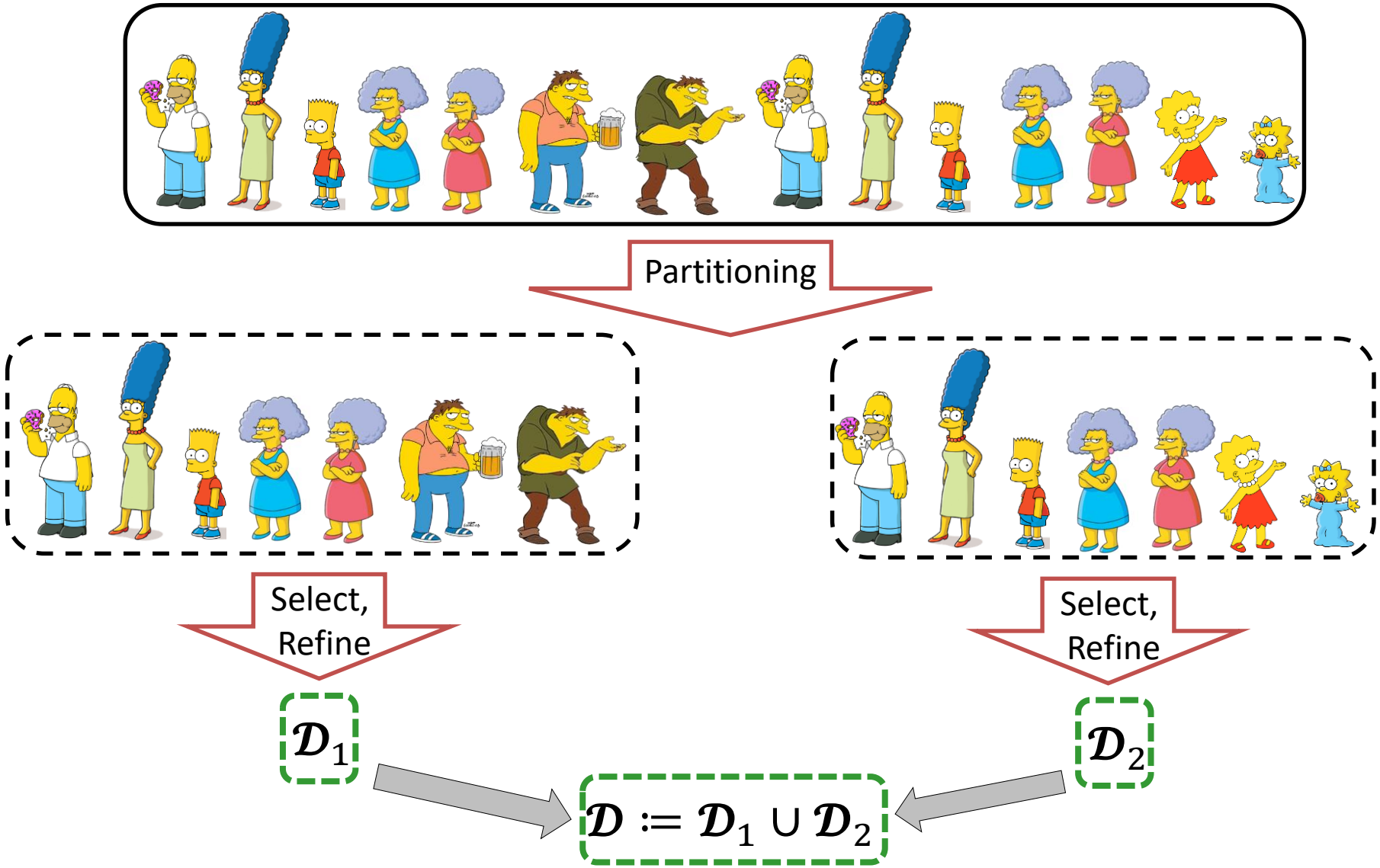


Q: What's wrong with MERLIN? ***A:*** Nothing, it's perfect, but…

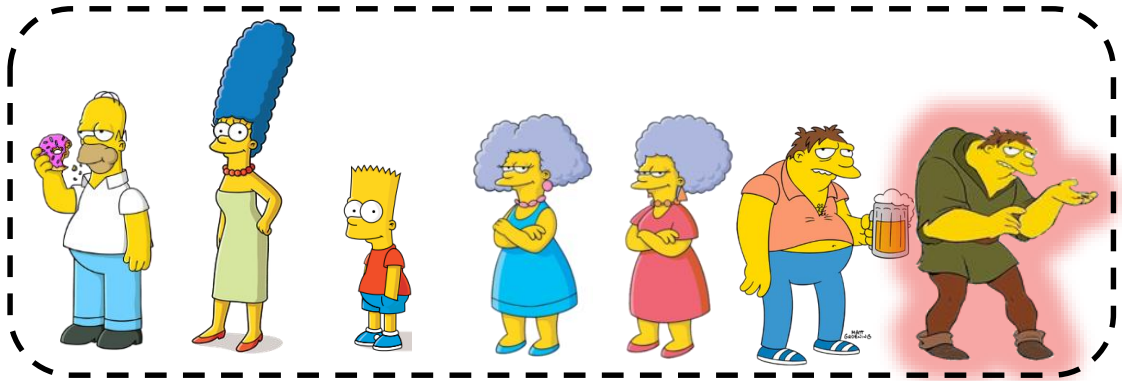
MERLIN (similarly to DRAG as its predecessor)

- is serial, not parallel
- has redundant calculations
  (the algorithm computes the Euclidean distances
  between candidates and subsequences
  along the moving sliding window)

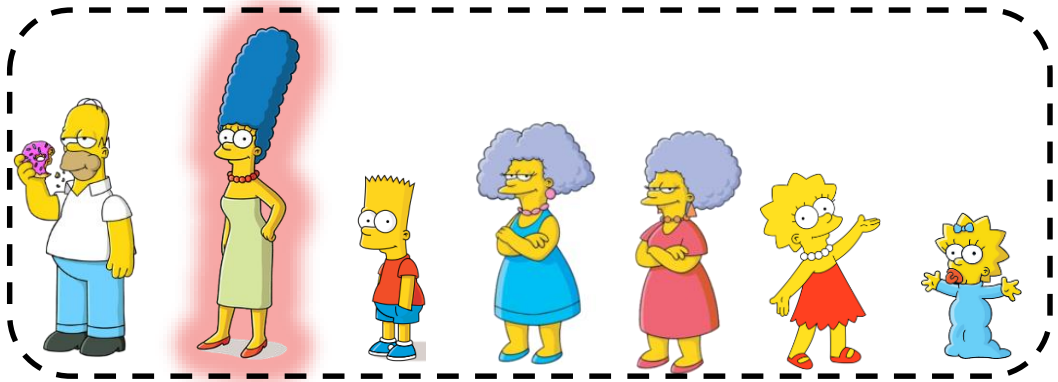
Can we discover discords in parallel?



Naïve parallel discord discovery does not work

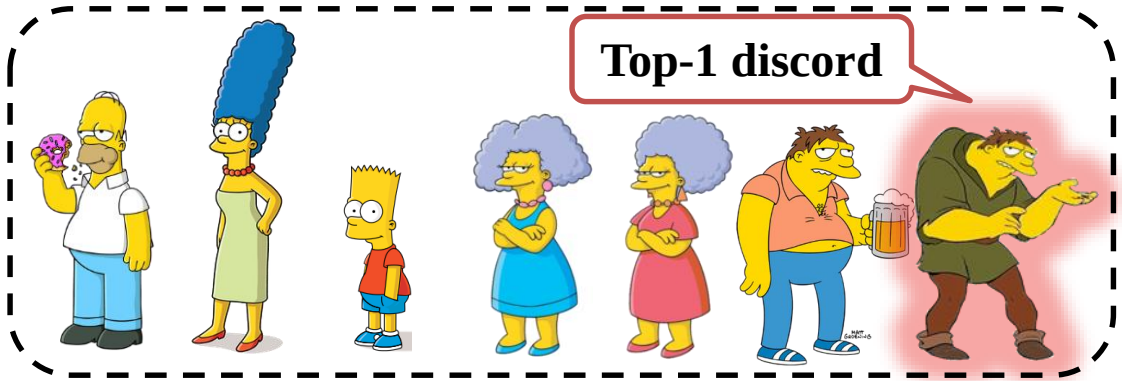


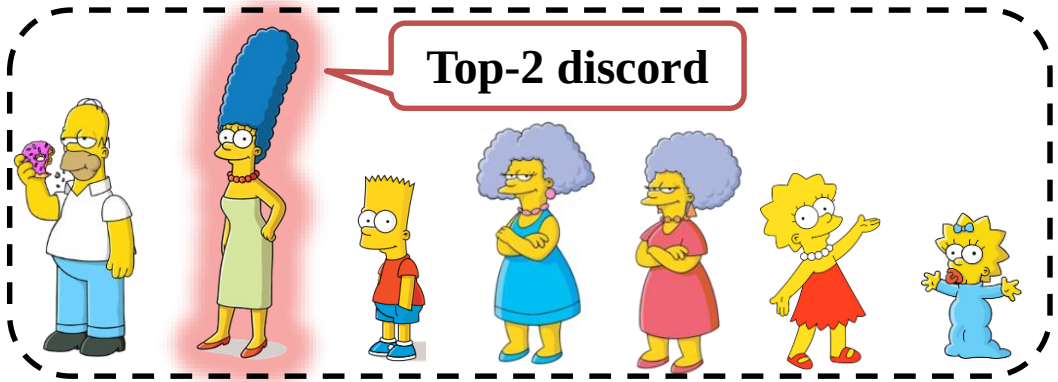
0	5	2	4	4	6	8
5	0	2.5	3	3	6	10
2	2.5	0	4	4	6	9
4	3	4	0	0.5	5	8
4	3	4	0.5	0	5	8
6	6	6	5	5	0	**7**
8	10	9	8	8	7	0



0	5	2	4	4	2.5	2.5
5	0	2.5	3	3	2	2
2	**2. 5**	0	4	4	0.5	0.5
4	3	4	0	0.5	4	4
4	3	4	0.5	0	4	4
2.5	2	0.5	4	4	0	0.5
2.5	2	0.5	4	4	0.5	0

Naïve parallel discord discovery does not work

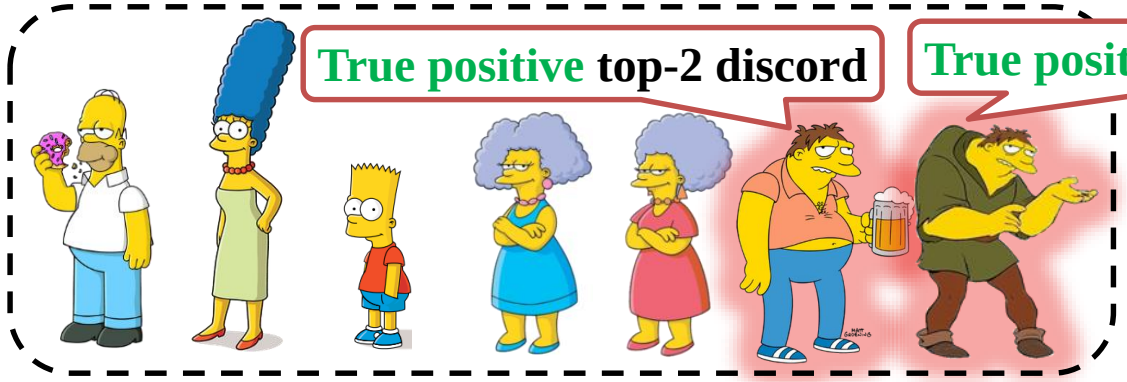


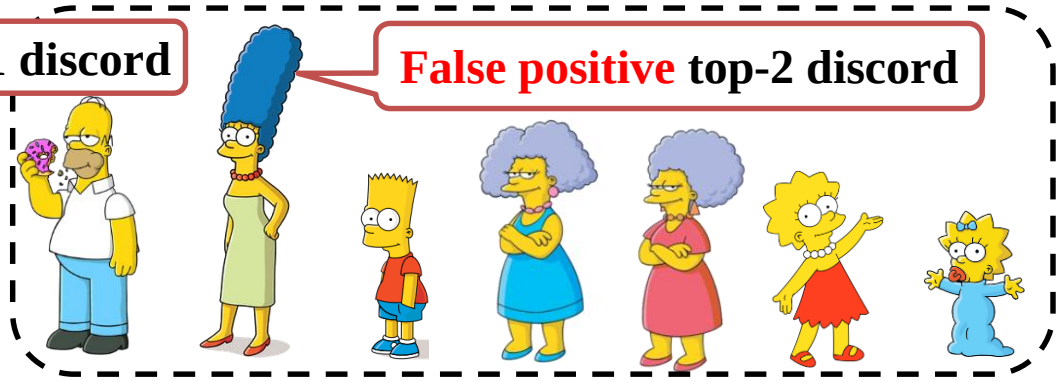


0	5	2	4	4	6	8
5	0	2.5	3	3	6	10
2	2.5	0	4	4	6	9
4	3	4	0	0.5	5	8
4	3	4	0.5	0	5	8
6	6	6	5	5	0	**7**
8	10	9	8	8	7	0

0	5	2	4	4	2.5	2.5
5	0	2.5	3	3	2	2
2	**2. 5**	0	4	4	0.5	0.5
4	3	4	0	0.5	4	4
4	3	4	0.5	0	4	4
2.5	2	0.5	4	4	0	0.5
2.5	2	0.5	4	4	0.5	0

Naïve parallel discord discovery does not work

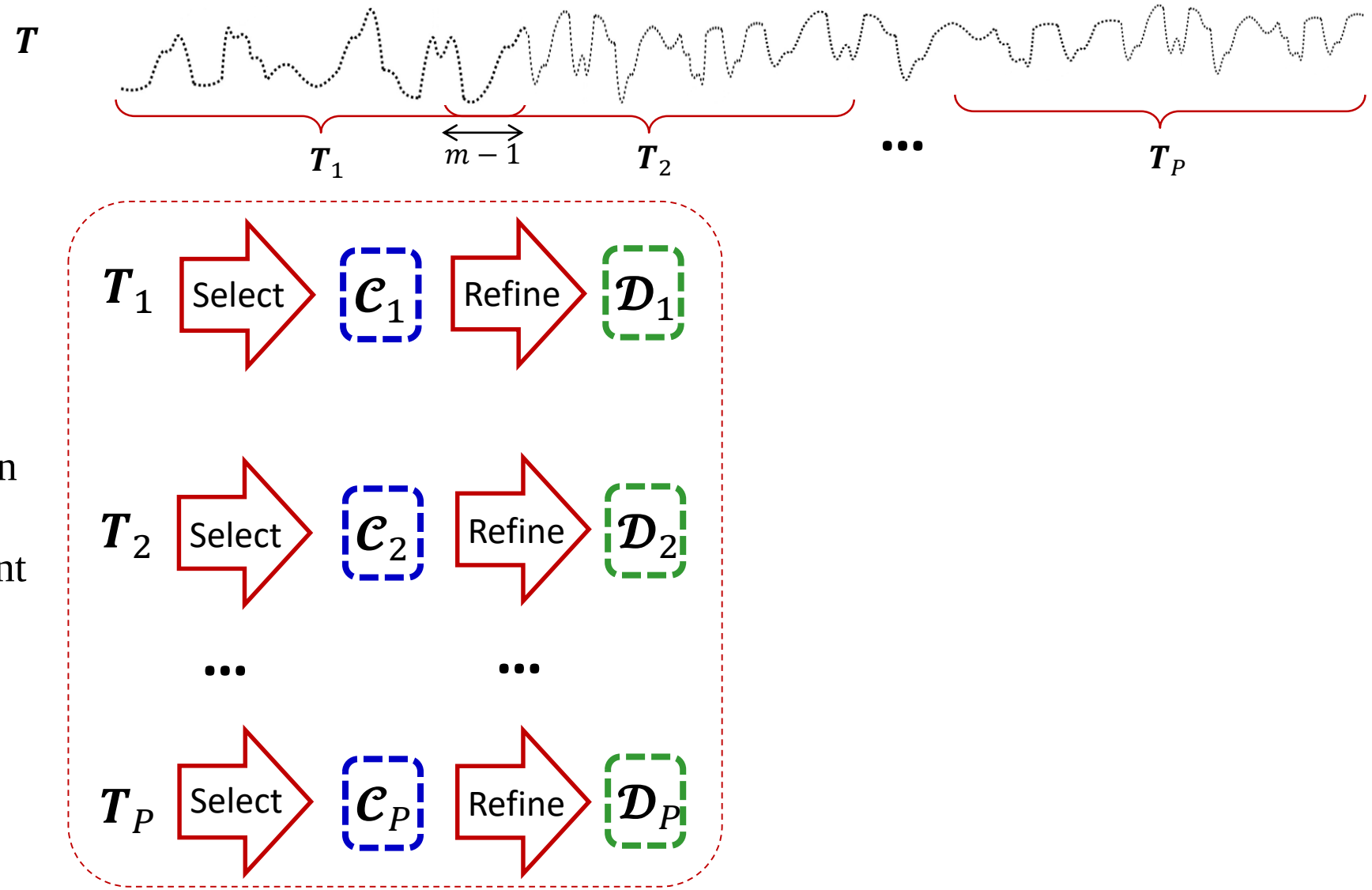




0	5	2	4	4	6	8
5	0	2.5	3	3	6	10
2	2.5	0	4	4	6	9
4	3	4	0	0.5	**5**	8
4	3	4	0.5	0	**5**	8
6	6	6	5	5	0	7
8	10	9	8	8	7	0

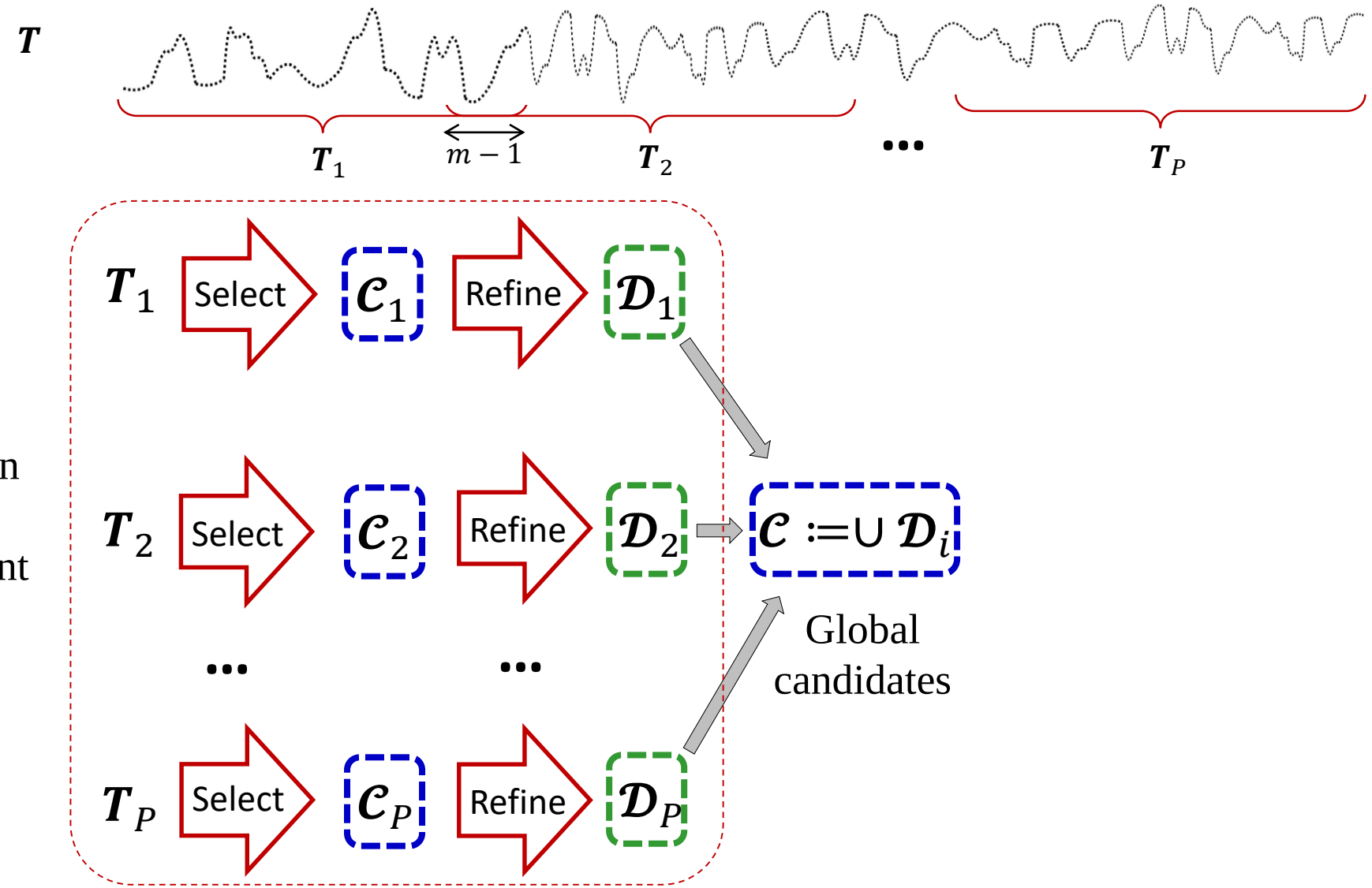
0	5	2	4	4	2.5	2.5
5	0	2.5	3	3	2	2
2	**2. 5**	0	4	4	0.5	0.5
4	3	4	0	0.5	4	4
4	3	4	0.5	0	4	4
2.5	2	0.5	4	4	0	0.5
2.5	2	0.5	4	4	0.5	0

Parallel discovery: Local discords must be globally refined



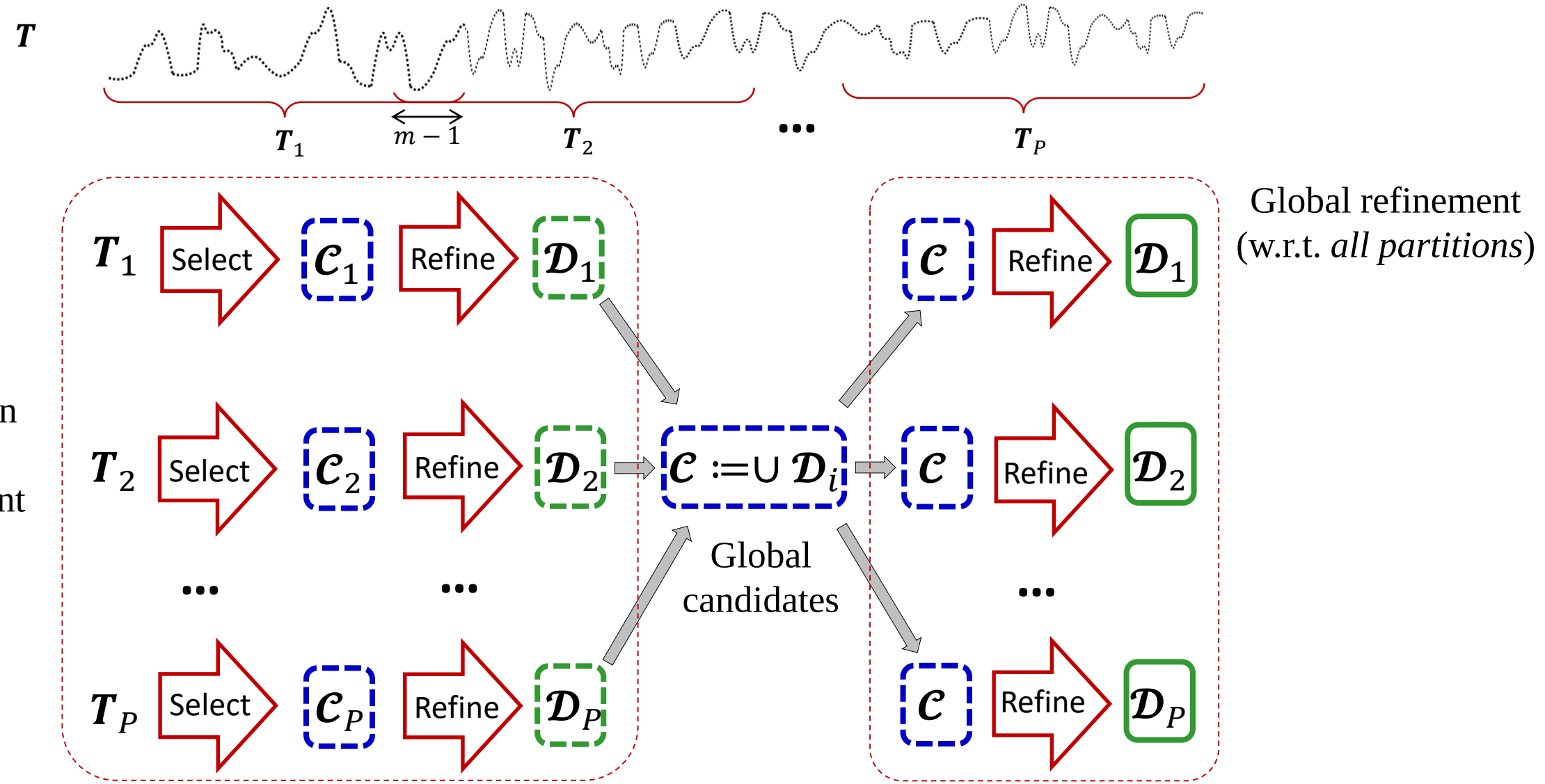
Local selection and local refinement (w.r.t. *each partition*)

Parallel discovery: Local discords must be globally refined



Local selection and local refinement (w.r.t. *each partition*)

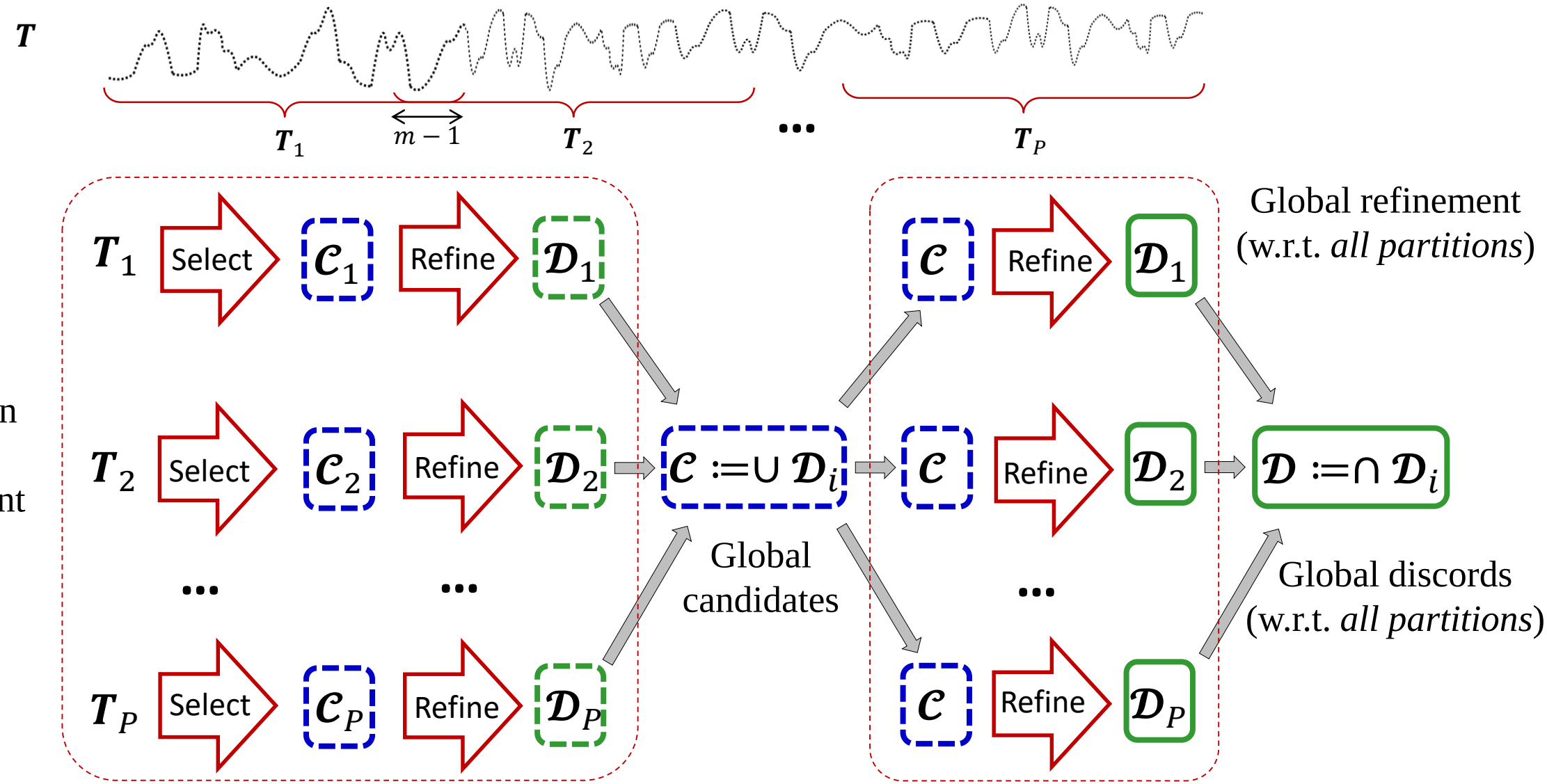
Parallel discovery: Local discords must be globally refined



Local selection and local refinement (w.r.t. *each partition*)

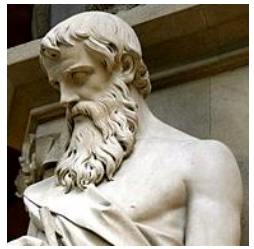
Global refinement (w.r.t. *all partitions*)

Parallel discovery: Local discords must be globally refined



Q: What dist(·,·) function do we use? ***A:*** z-norm ED^2

$$\mathbf{ED}(X,Y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$$



Euclid (325-265 BC)

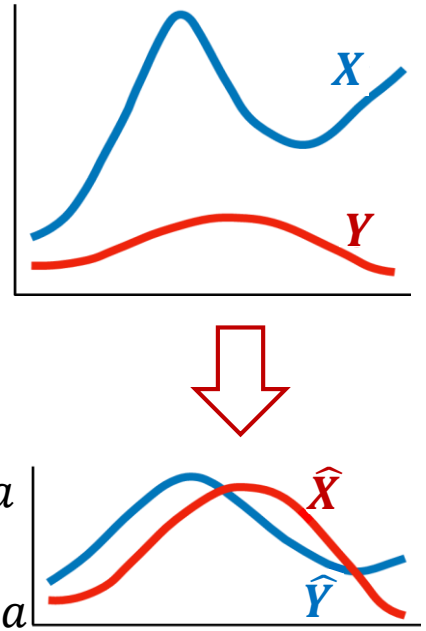
z-normalization

$$\hat{T} = (\hat{t}_1, \ldots, \hat{t}_m), \quad \hat{t}_i = \frac{t_i - \mu}{\sigma}$$

$$\mu = \frac{1}{n}\sum_{i=1}^{m} t_i,$$

$$\sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m} t_i^2 - \mu^2}$$

$$\mu_{\hat{T}} = 0, \sigma_{\hat{T}} = 1$$

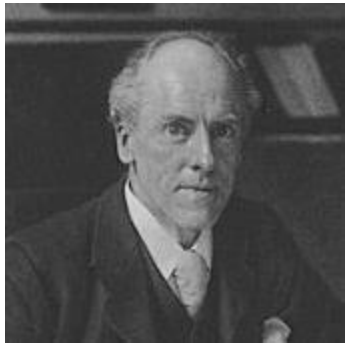


$$\mathbf{ED}^2_{\mathbf{norm}}(X,Y) = \mathrm{ED}^2(\hat{X},\hat{Y}) = {}^{*}$$

$$= 2m\left(1 - \boldsymbol{corr}(X,Y)\right) =$$

$$= 2m\left(1 - \frac{X \cdot Y - m\mu_X\mu_Y}{m\sigma_X\sigma_Y}\right)$$

Pearson correlation of subsequences

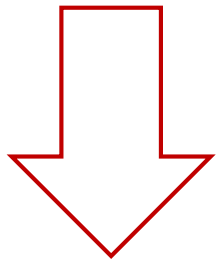


Karl Pearson
1857-1936

* Mueen A. *et al.* Fast approximate correlation for massive time-series data. SIGMOD 2010. pp. 171-182. DOI: 10.1145/1807167.1807188

We can calculate distances even faster

$$\mathrm{ED}^2_{\mathrm{norm}}\left(T_{i,m}, T_{j,m}\right) = 2m\left(1 - \frac{T_{i,m} \cdot T_{j,m} - m\boldsymbol{\mu_i}\boldsymbol{\mu_j}}{m\boldsymbol{\sigma_i}\boldsymbol{\sigma_j}}\right)$$



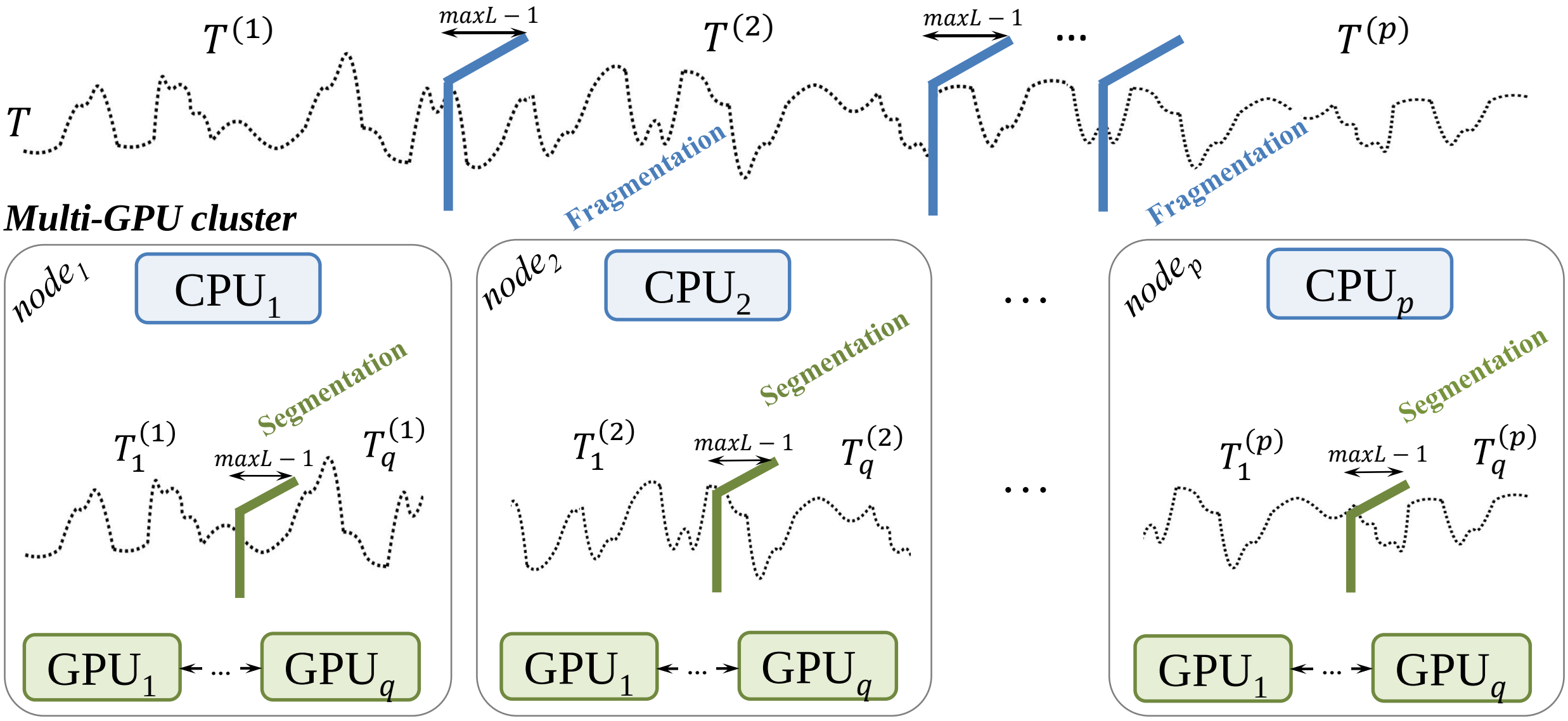
If we calculate distances between subsequences with lengths in the $minL..maxL$ range, then we can use recurrent formulas*

$$\boldsymbol{\mu_{T_{i,m+1}}} = \frac{1}{m+1}\left(m\boldsymbol{\mu_{T_{i,m}}} + t_{i+m}\right),$$

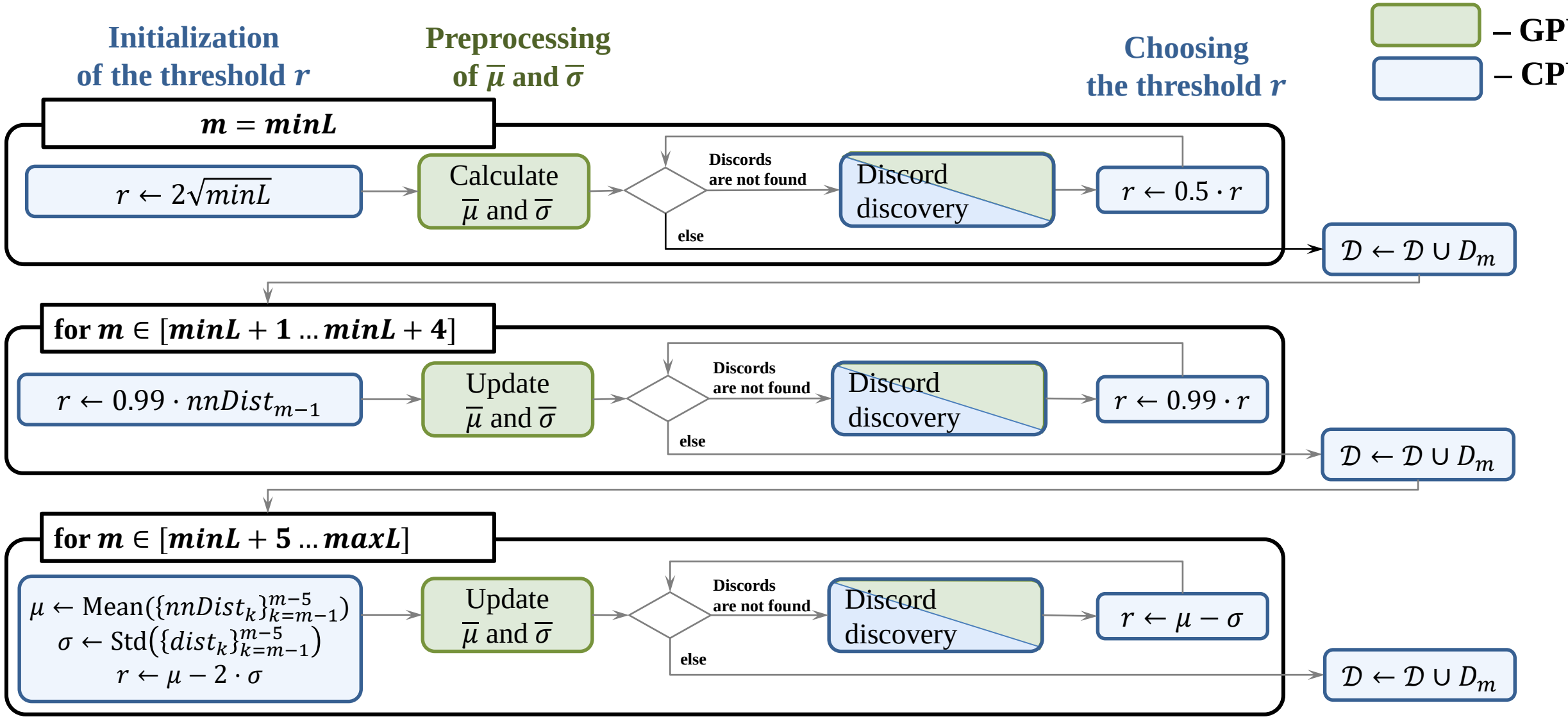
$$\boldsymbol{\sigma^2_{T_{i,m+1}}} = \frac{m}{m+1}\left(\boldsymbol{\sigma^2_{T_{i,m}}} + \frac{1}{m+1}\left(\boldsymbol{\mu_{T_{i,m}}} - t_{i+m}\right)^2\right)$$

* Zymbler M., Kraeva Y. High-performance time series anomaly discovery on graphics processors. Mathematics. 2023. 11(14). art. 3193. DOI: 10.3390/math11143193.

Two-level data parallelism (PADDi algorithm)



General scheme of calculations

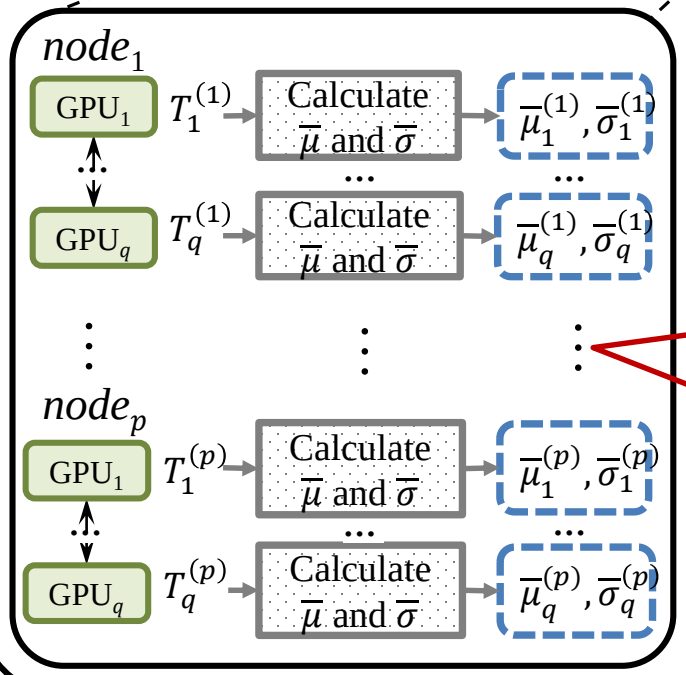


Preprocessing

for each $m \in [minL; maxL]$

Initialization of r → **Preprocessing** → Discords are not found → Discord discovery → Update r

else → $\mathcal{D} \leftarrow \mathcal{D} \cup D_m$



$$\mathrm{ED}^2_{\mathrm{norm}}\left(T_{i,m}, T_{j,m}\right) = 2m\left(1 - \frac{T_{i,m} \cdot T_{j,m} - m\boldsymbol{\mu_i}\boldsymbol{\mu_j}}{m\boldsymbol{\sigma_i}\boldsymbol{\sigma_j}}\right)$$

$\boldsymbol{m = minL}$ $\qquad$ $\boldsymbol{minL < m \leq maxL}$

Calculate: $O(m)$ complexity

$$\boldsymbol{\mu}_{T_{i,m}} = \frac{1}{m}\sum_{k=0}^{m-1} t_{i+k},$$

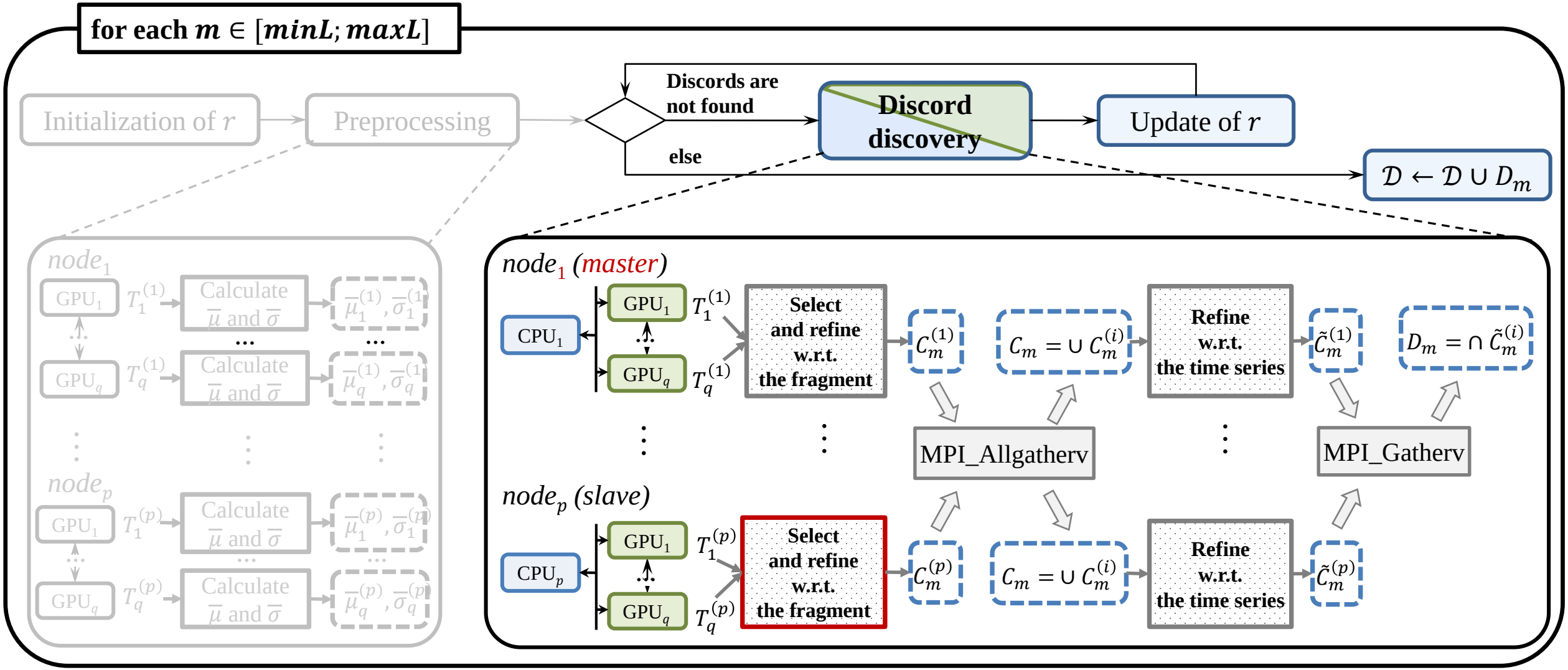
$$\boldsymbol{\sigma}^2_{T_{i,m}} = \frac{1}{m}\sum_{k=0}^{m-1} t^2_{i+k} - \mu^2_{T_{i,m}}$$

Update: $O(1)$ complexity

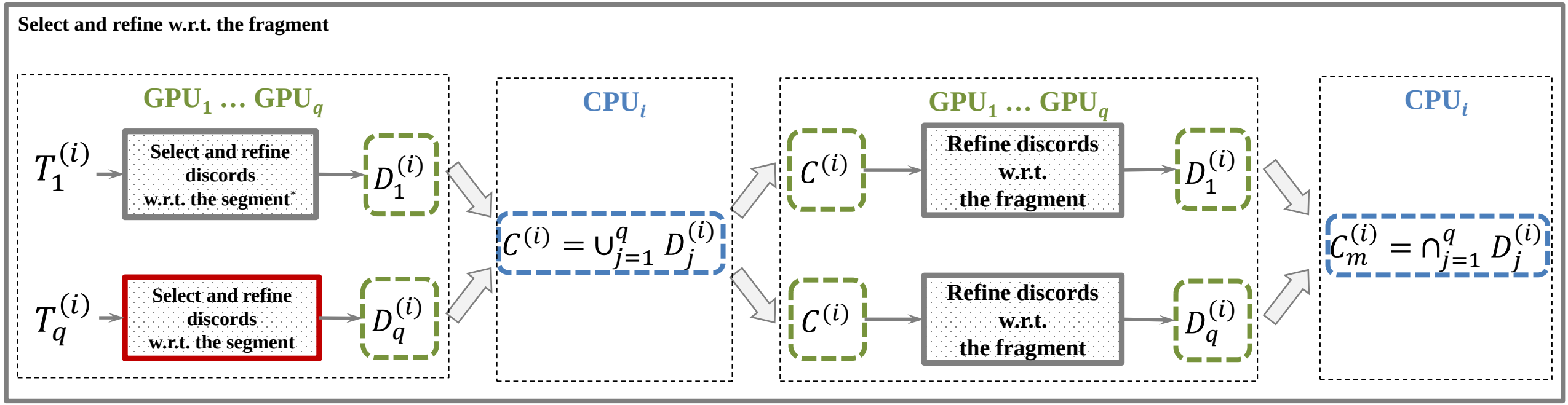
$$\boldsymbol{\mu}_{T_{i,m+1}} = \frac{1}{m+1}\left(m\boldsymbol{\mu}_{T_{i,m}} + t_{i+m}\right),$$

$$\boldsymbol{\sigma}^2_{T_{i,m+1}} = \frac{m}{m+1}\left(\boldsymbol{\sigma}^2_{T_{i,m}} + \frac{1}{m+1}\left(\boldsymbol{\mu}_{T_{i,m}} - t_{i+m}\right)^2\right)$$

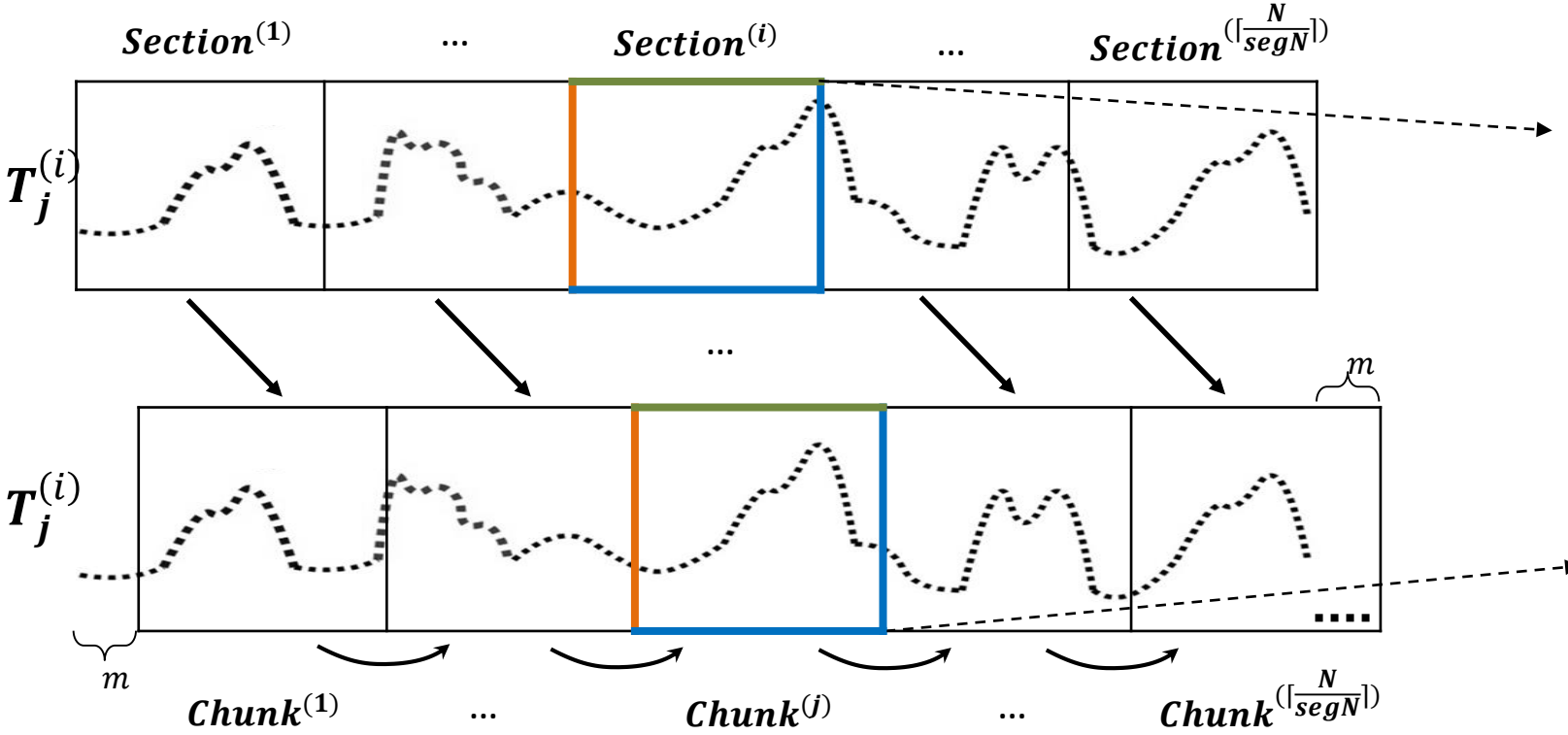
Discord discovery

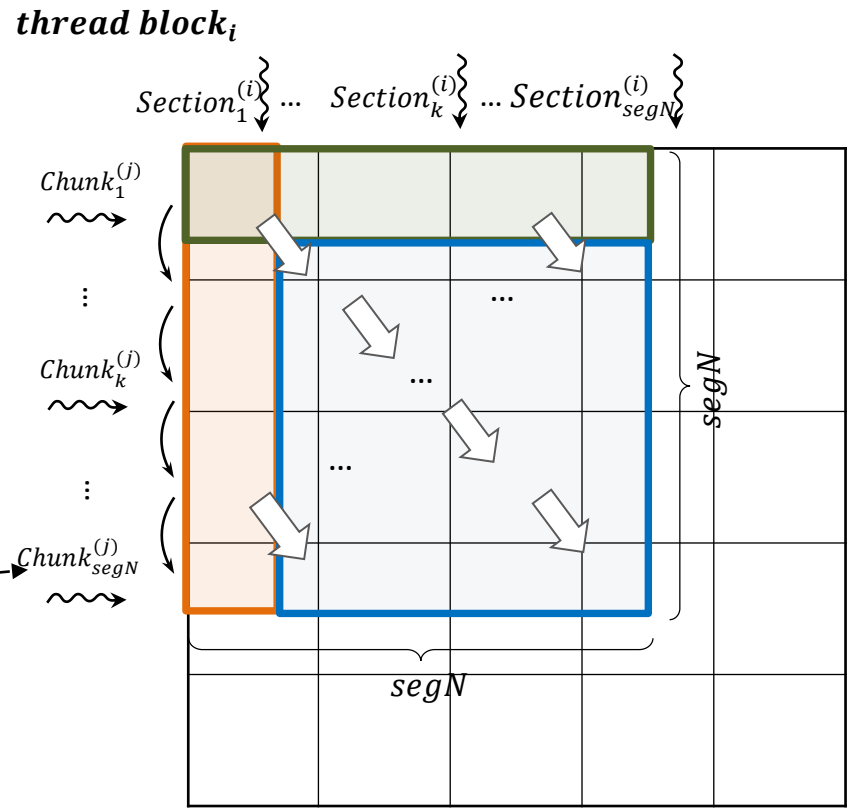


Candidate selection and discord refinement w.r.t. the fragment



Candidate selection and discord refinement w.r.t. the segment





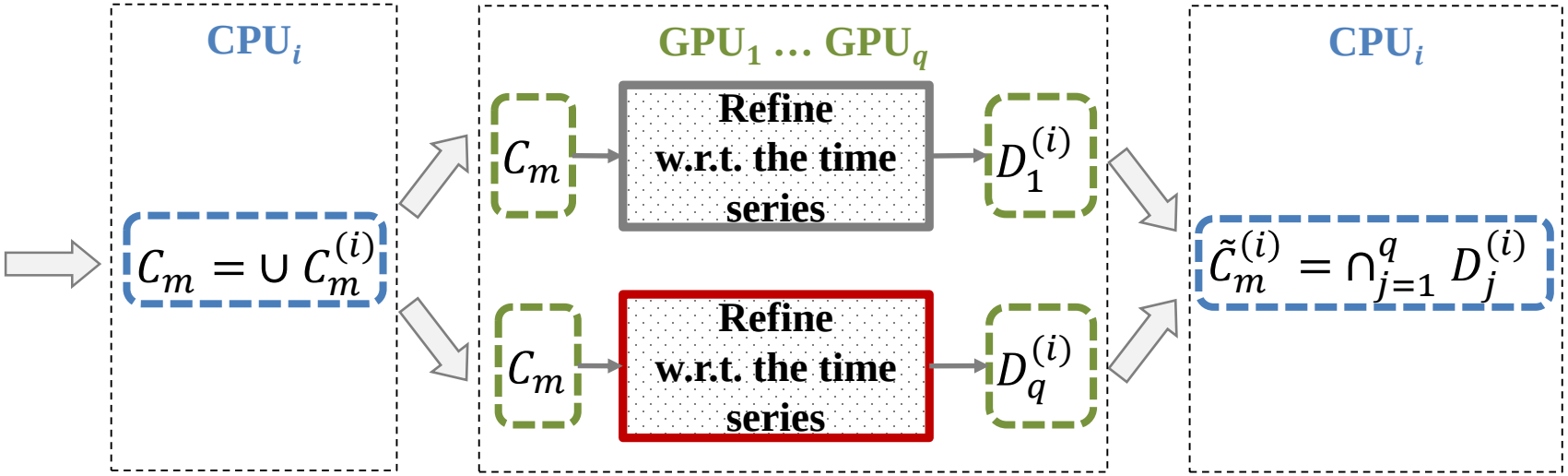
$$\mathrm{ED}^2_{\mathrm{norm}}\left(T_{i,m}, T_{j,m}\right) = 2m\left(1 - \frac{T_{i,m} \cdot T_{j,m} - m\mu_i\mu_j}{m\sigma_i\sigma_j}\right)$$

$\boldsymbol{O(1)}$ **instead of** $\boldsymbol{O(m)}$!

$$\boldsymbol{QTrow^{(i)}(tid)} = \sum_{k=1}^{m} T_{tid}^{(i)}(k) \cdot Chunk_1^{(j)}(k) \qquad \boldsymbol{QTcol^{(i)}(tid)} = \sum_{k=1}^{m} T_1^{(i)}(k) \cdot Chunk_{tid}^{(j)}(k)$$

$$\boldsymbol{QTrow^{(i)}(tid)} = QTrow^{(i)}(tid-1) - T_{tid-1}^{(i)}(1) \cdot Chunk_{tid-1}^{(j)}(1) + T_{tid}^{(i)}(m) \cdot Chunk_{tid}^{(j)}(m)$$

Discord refinement w.r.t. the time series

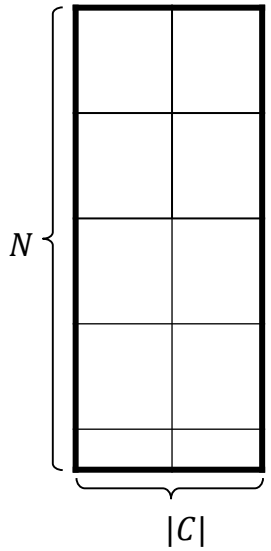


Discord refinement

Distance matrix

$Dist \in \mathbb{R}^{N \times |C|}$

$$Dist_{i,j} = 2m\left(1 - \frac{T_{i,m} \cdot T_{j,m} - m\mu_i\mu_j}{m\sigma_i\sigma_j}\right)$$



Segment $T_j^{(i)}$

Segment subsequences

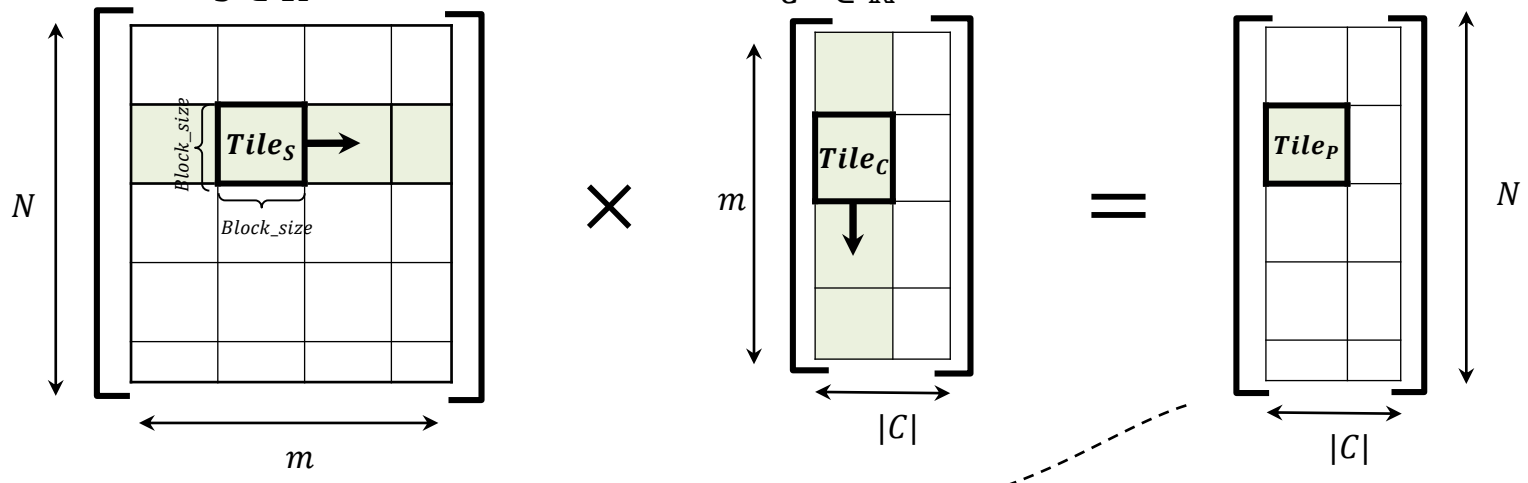
$S \in \mathbb{R}^{N \times m}$

Candidates

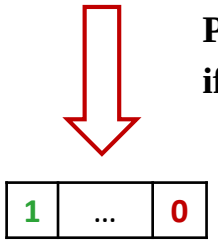
$C^T \in \mathbb{R}^{|C| \times m}$

Matrix of scalar products

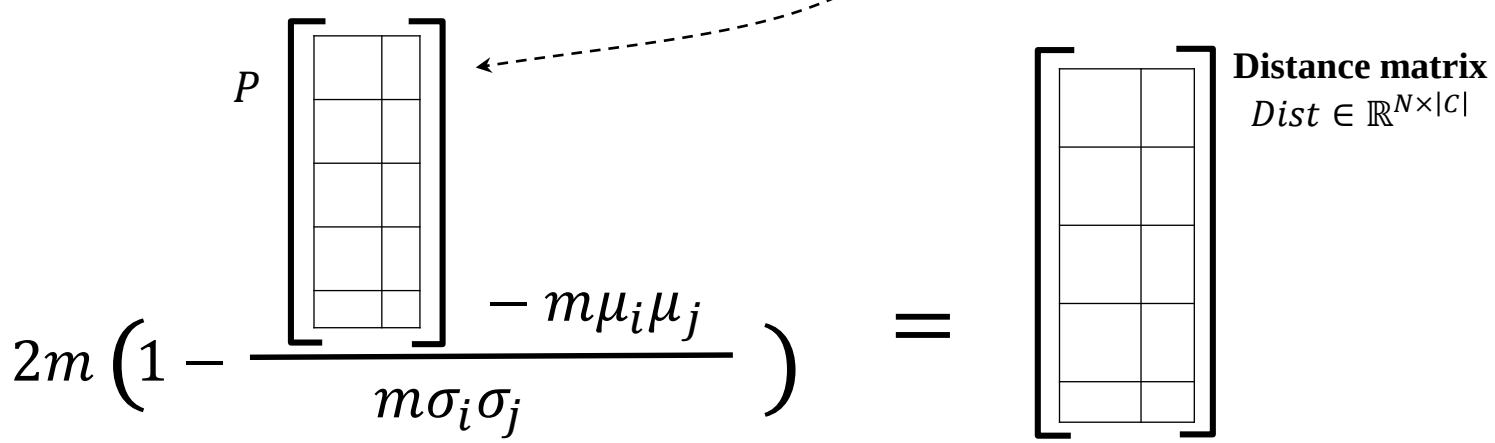
$P \in \mathbb{R}^{N \times |C|}$



Discord
Bitmap



Prune the candidate C_j,
if $\min_{1<i<N}(Dist_{i,j}) < r^2$



Distance matrix

$Dist \in \mathbb{R}^{N \times |C|}$

Experimental evaluation

- **HPC clusters**
  - ***Lobachevsky (University of Nizhny Novgorod)***

    64×{2×Sandy Bridge E5-2660 + **3×Kepler K20X** (2 688 cores @0.73 GHz, 1.3 TFLOPS)}
  - ***Lomonosov-2 (Moscow State University)***

    64×{1×Xeon Gold 6126 + **2×Tesla P100** (3 584 cores @1.19 GHz, 4 TFLOPS)}
- **Data**

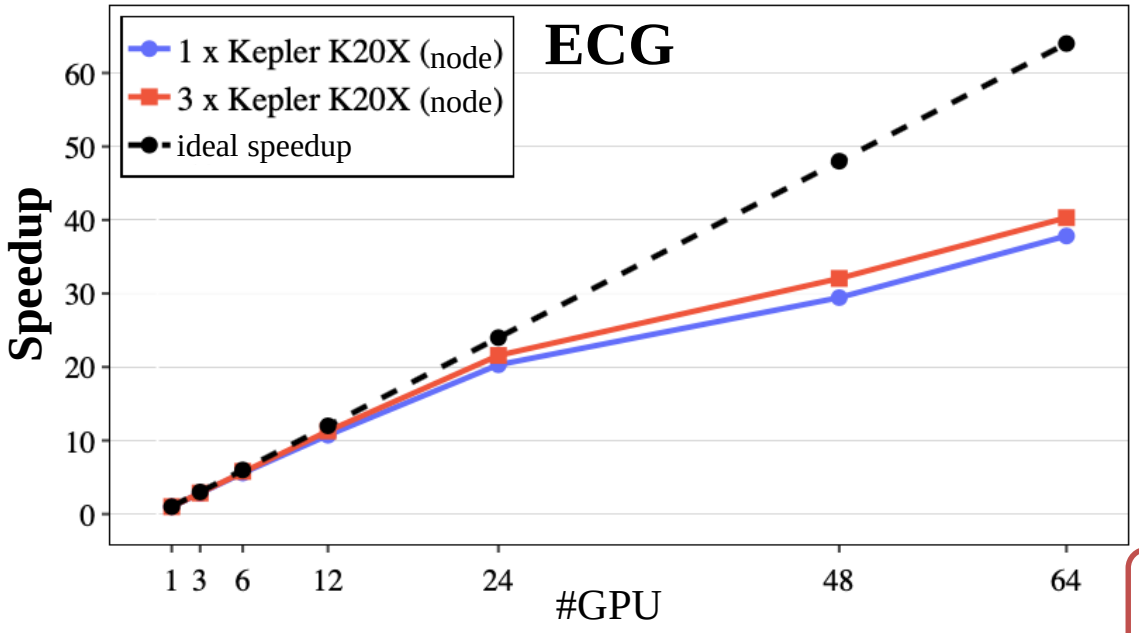
Time series	Length, n	Discord length range, $minL..maxL$	Subject domain
ECG[1)]	$2 \cdot 10^6$	$64..128$	ECG of an adult patient
Power[2)]			Annual household power consumption

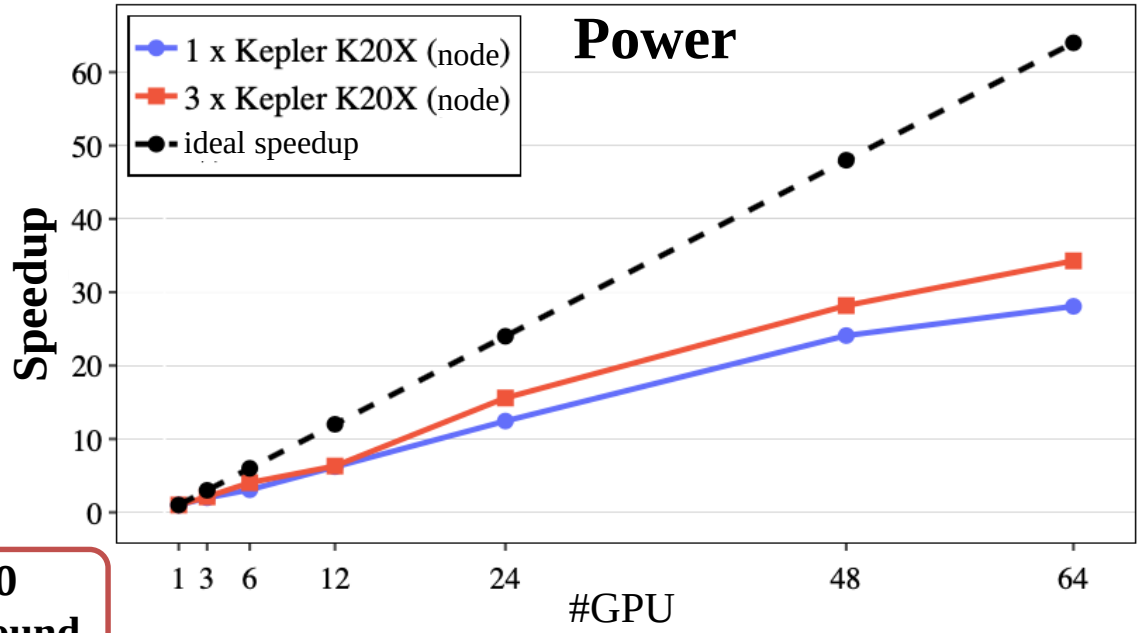
- **Metrics**
  - ***Performance***: average running time over 10 launches (without I/O)
  - ***Speedup***: $s(p) = {}^{t_1}/_{t_p}$, where t_1 and t_p are performance on one and p GPUs, respectively

[1)] Goldberger A. *et al*. PhysioBank, PhysioToolkit, and PhysioNet components of a new research resource for complex physiologic signals. Circulation. 2000. 101(23). pp. 215-220. DOI: 10.1161/01.CIR.101.23.e215.

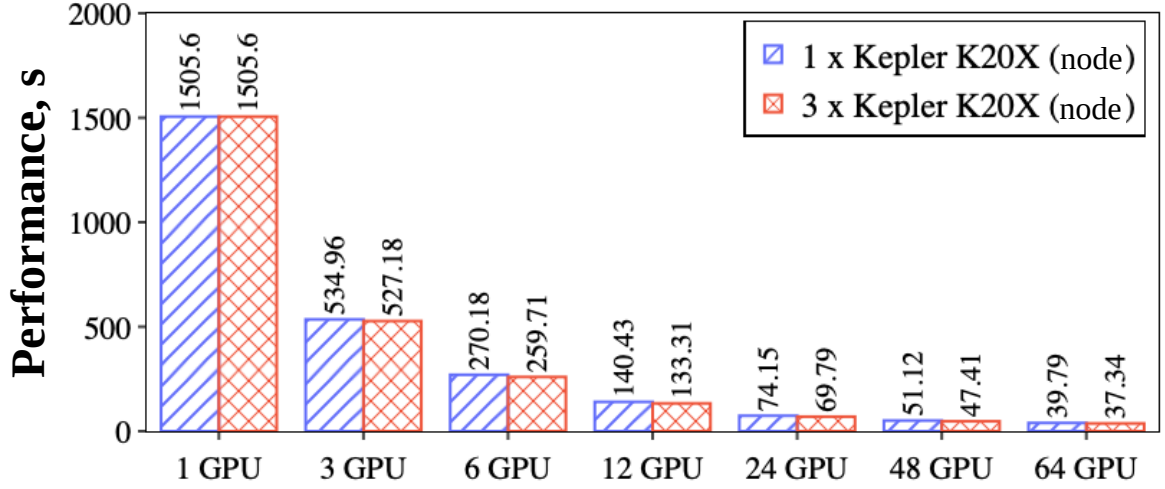
[2)] Individual household electric power consumption. URL: https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption/.

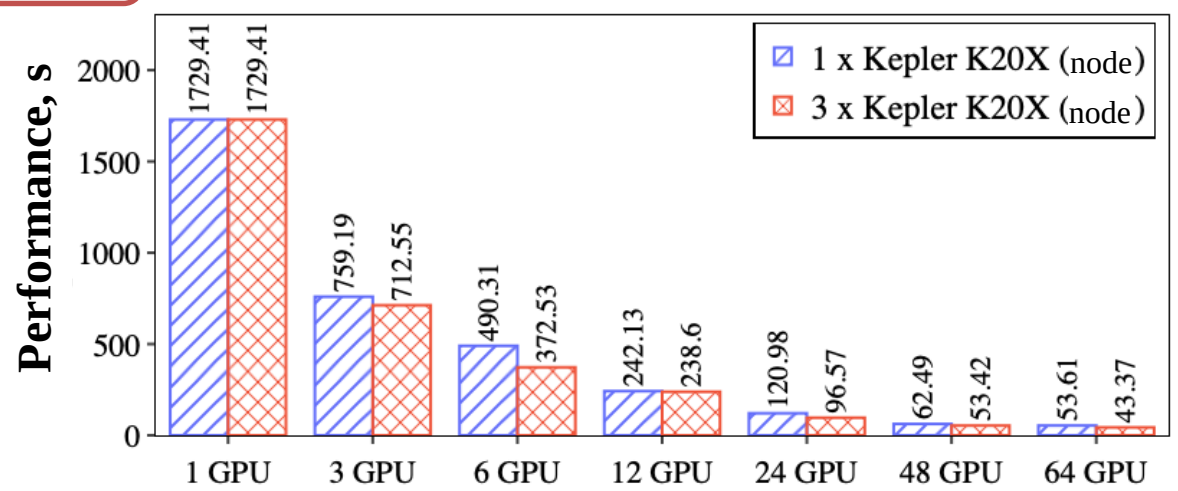
Speedup and performance (Lobachevsky)



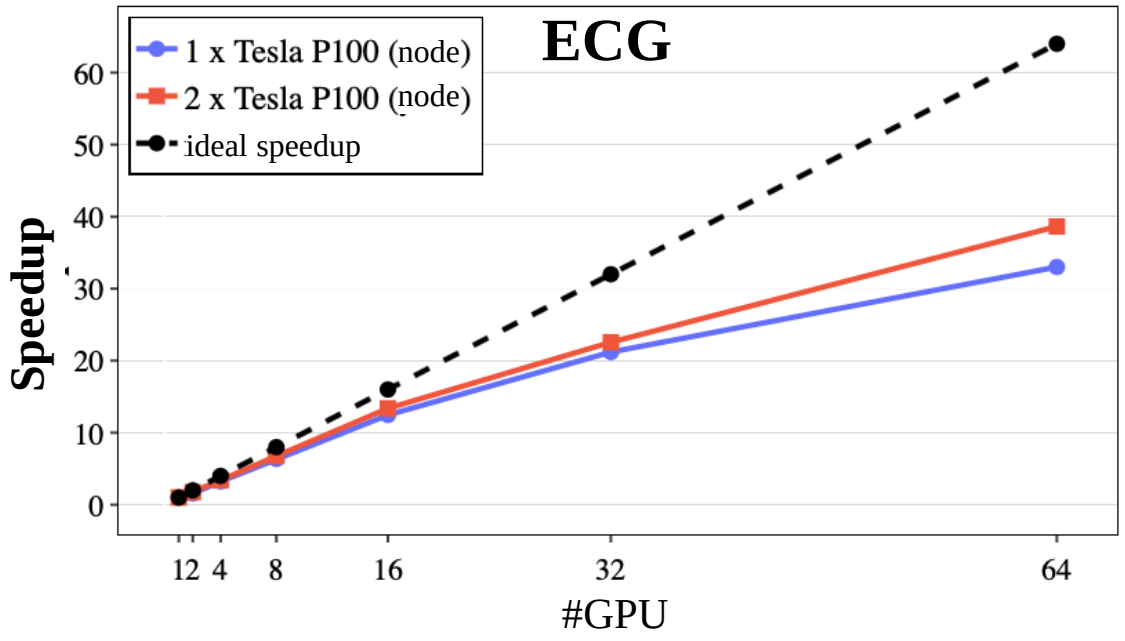


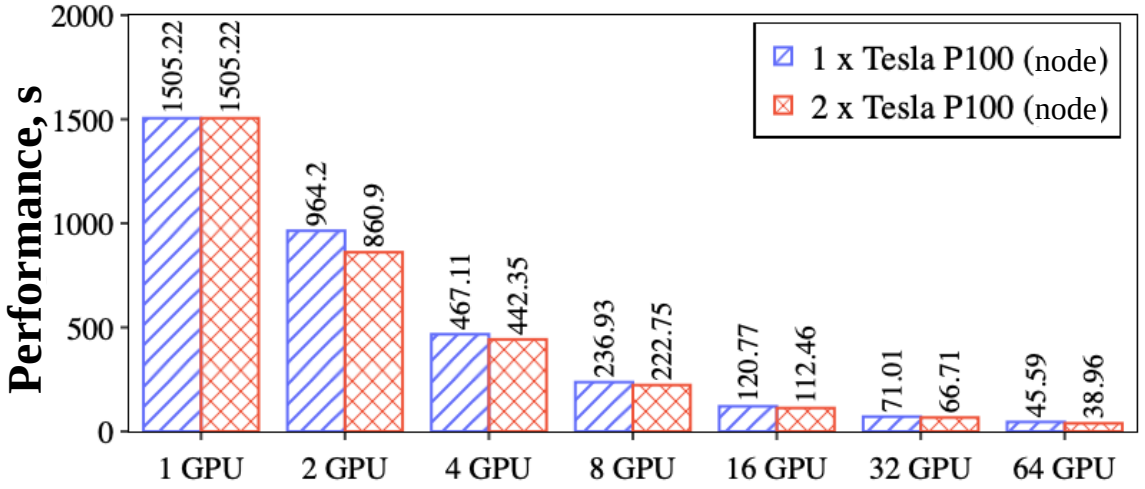
≈ 1500 discords found



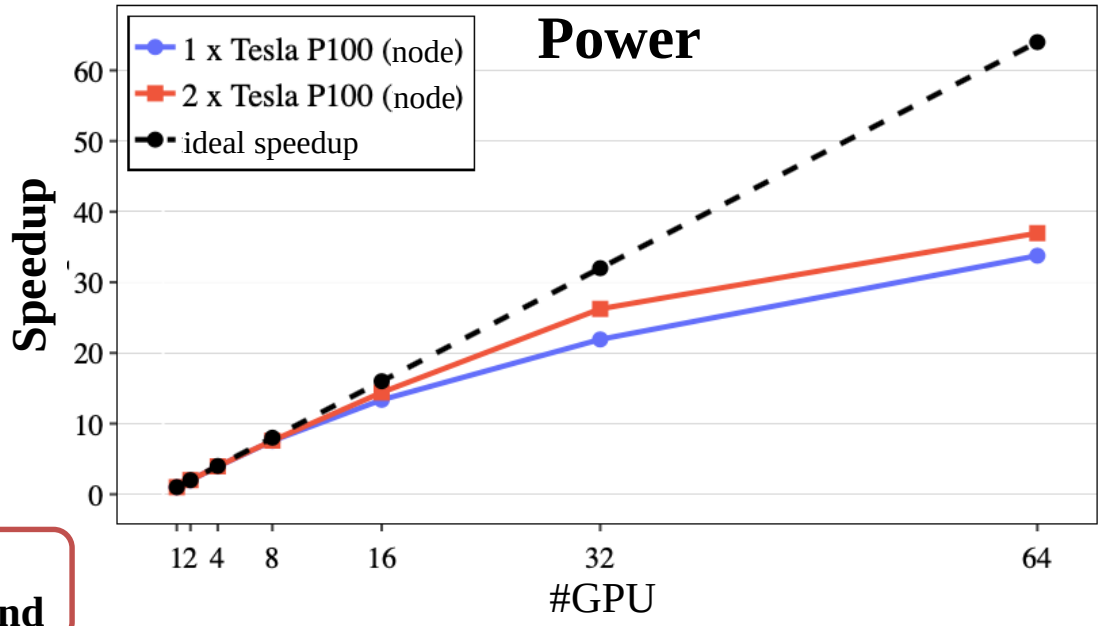


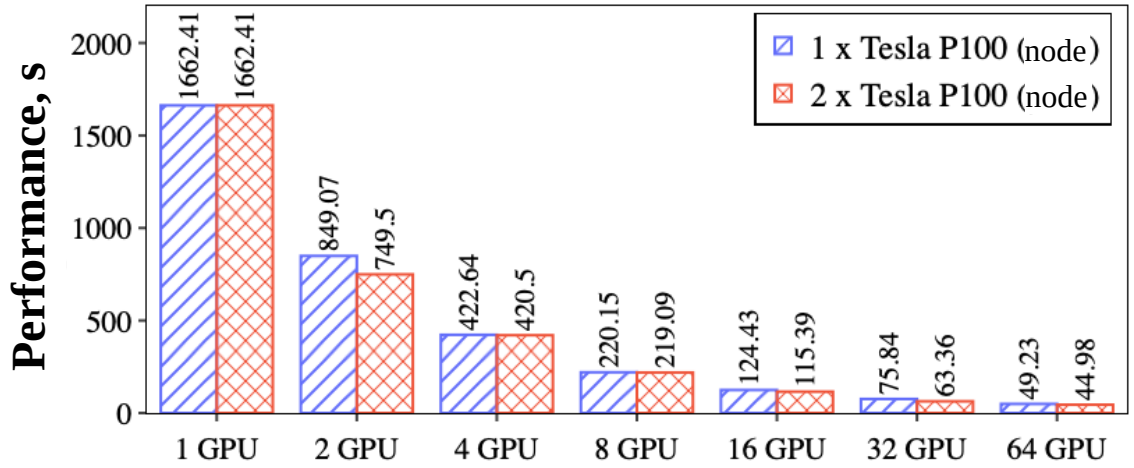
Speedup and performance: Lomonosov-2





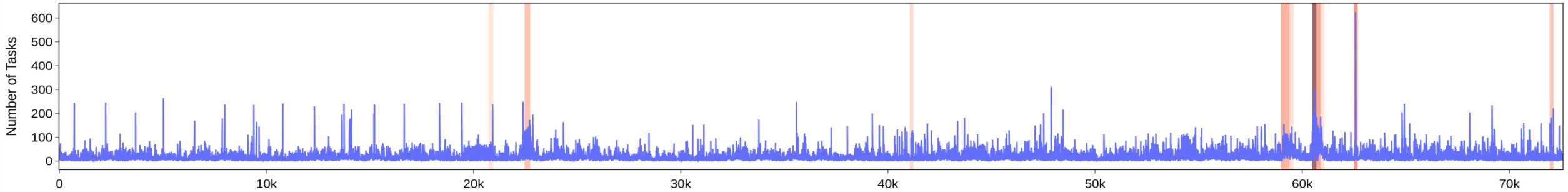
≈ 1500 discords found





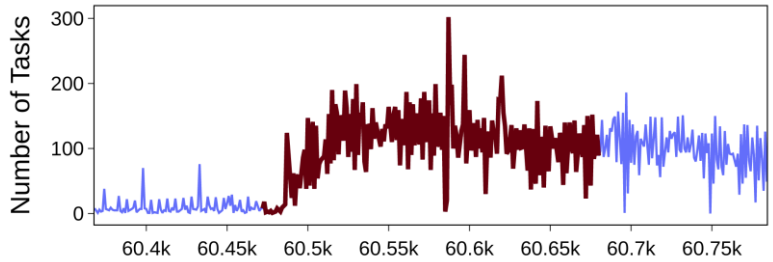
Case study: Alibaba Cloud HPC cluster workload

Tasks running in Alibaba PAI (Platform for Artificial Intelligence), July-August 2020*

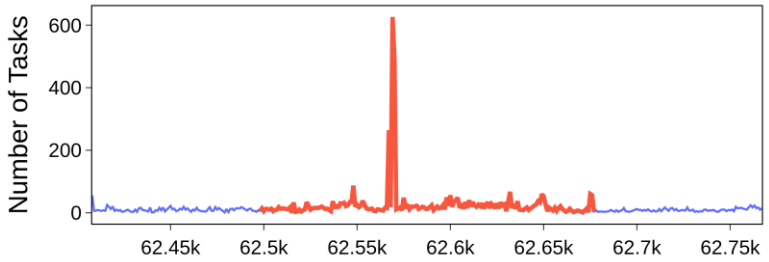


Top discords (range is 3-7 min)

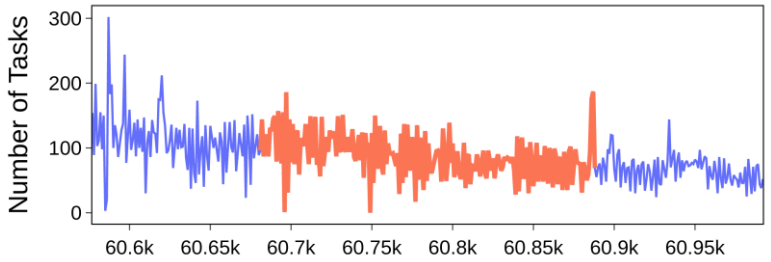
Top-1: $m = 209$ (3 min 29 sec)



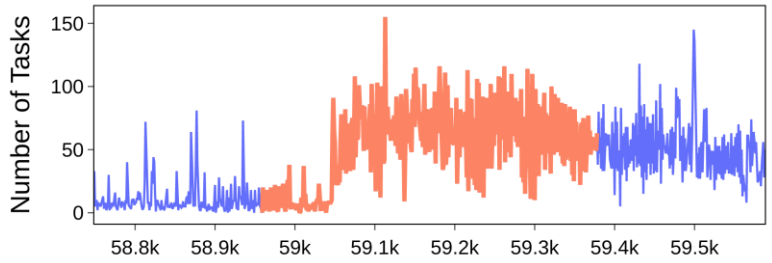
Top-2: $m = 180$ (3 min)



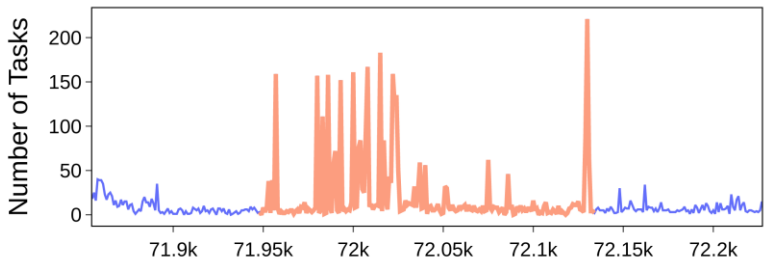
Top-3: $m = 208$ (3 min 28 sec)



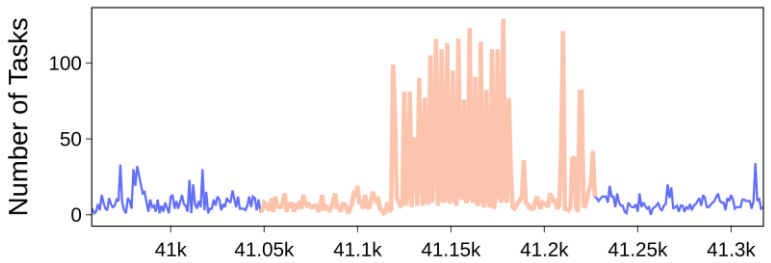
Top-4: $m = 420$ (7 min)



Top-5: $m = 283$ (4 min 43 sec)



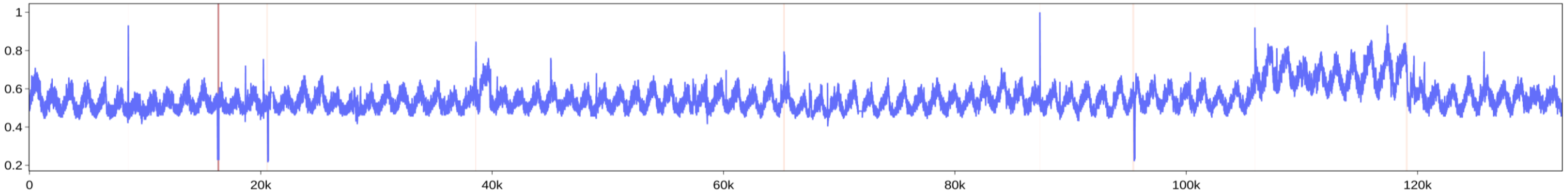
Top-6: $m = 187$ (3 min 7 sec)



* Weng Q. *et al.* MLaaS in the wild: workload analysis and scheduling in large-scale heterogeneous GPU clusters. NSDI 2022. pp. 945-960. https://www.usenix.org/conference/nsdi22/presentation/weng

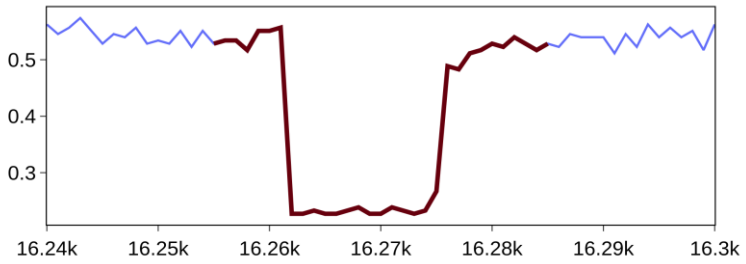
Case study: ebay HPC cluster workload

Pooled Server Metric of the eBay application server*

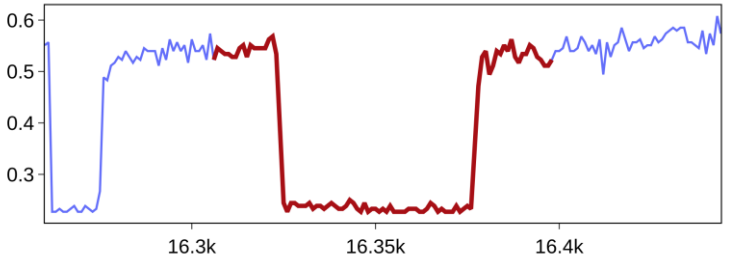


Top discords (range is 10-200)

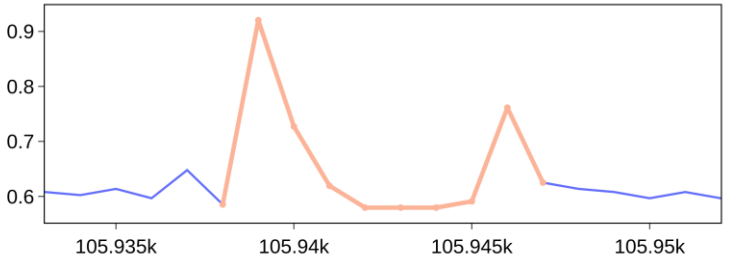
Top-1: $m = 31$



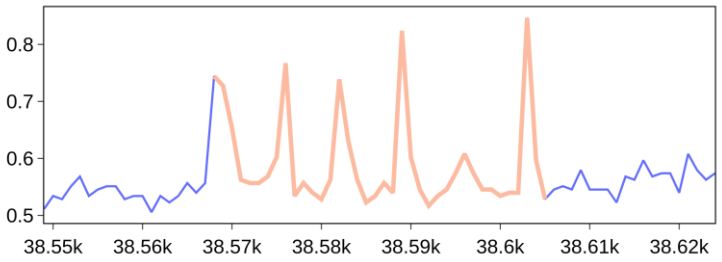
Top-2: $m = 93$



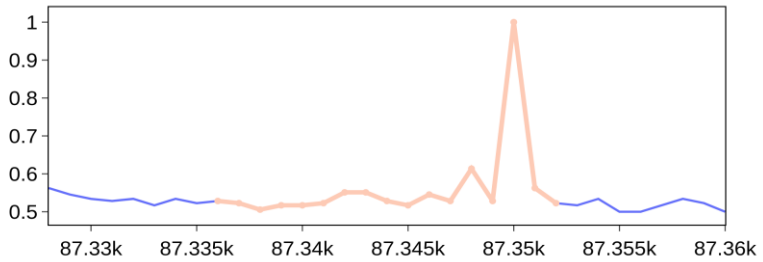
Top-3: $m = 10$



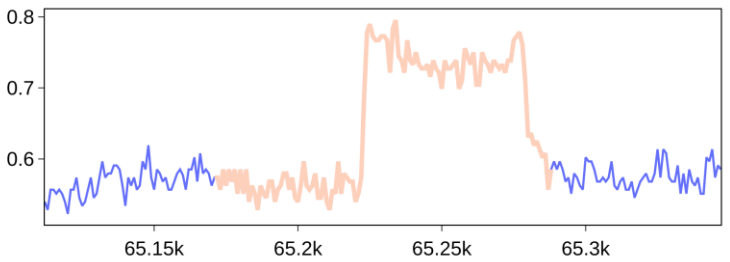
Top-4: $m = 38$



Top-5: $m = 17$



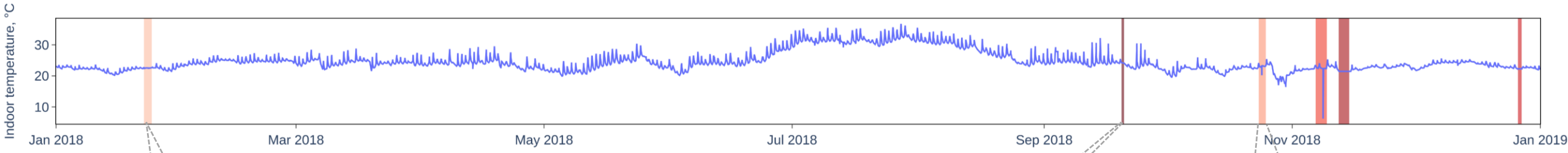
Top-6: $m = 118$



* Abdulaal A. *et al.* Practical approach to asynchronous multivariate time series anomaly detection and localization. KDD'21. pp. 2485-2494. DOI: 10.1145/3447548.3467174.

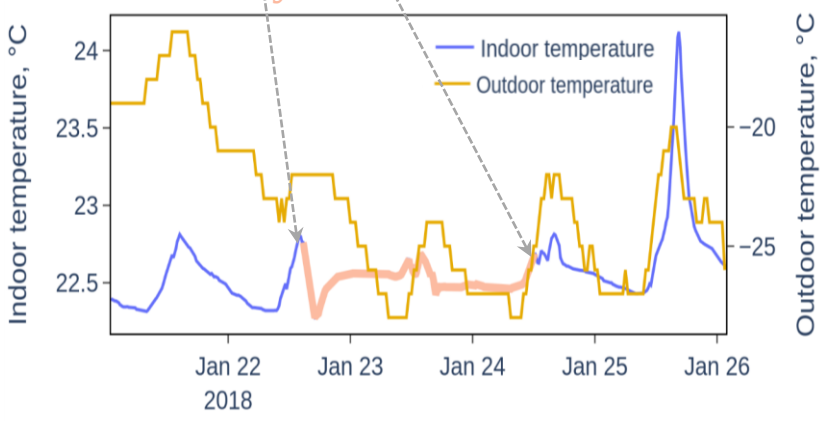
Case study: Smart heating control

Temperature in the SUSU lecture hall in 2018 (frequency is 4 times per hour)*

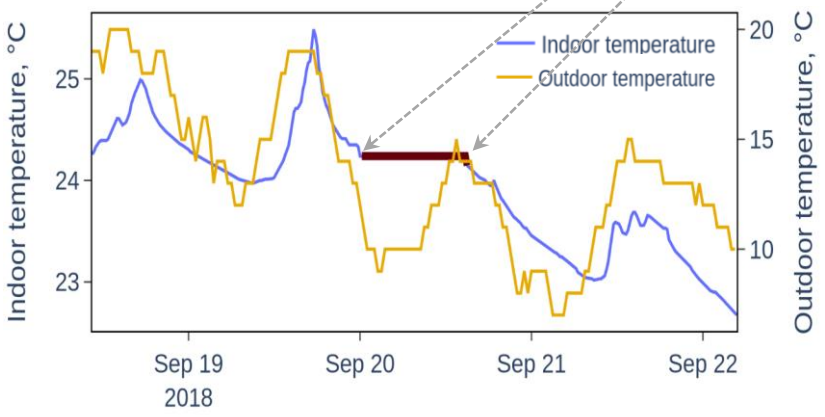


Top discords (range is 12-48 hours)

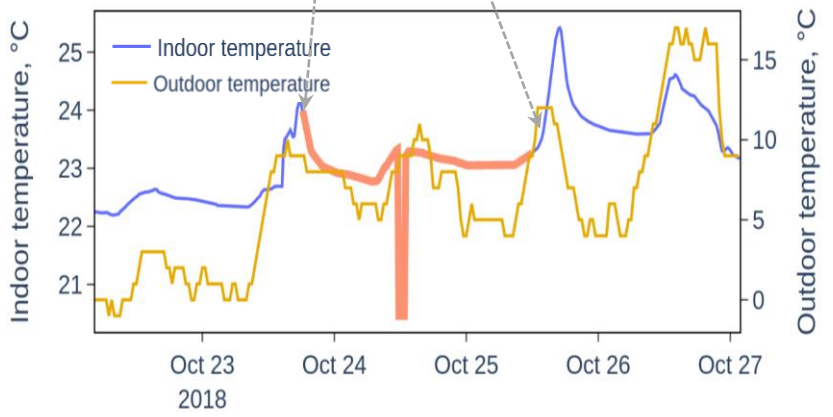
1 day 21 hours 45 min



15 hours



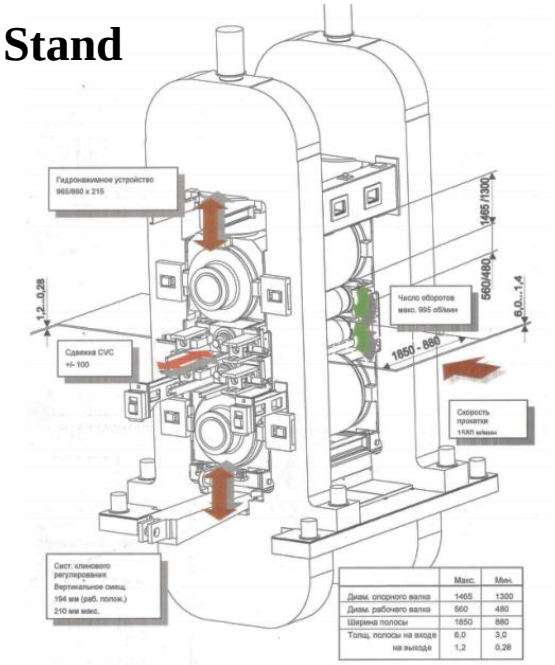
1 day 18 hours



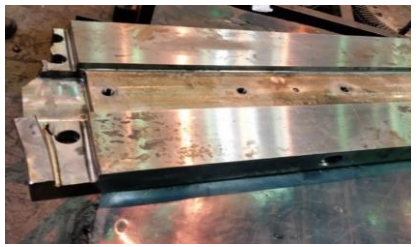
* Zymbler M., Kraeva Ya. *et al*. Cleaning sensor data in smart heating control system. GloSIC 2020. pp. 375-381. DOI: 10.1109/GloSIC50886.2020.9267813.

Case study: Cold rolling mill stand

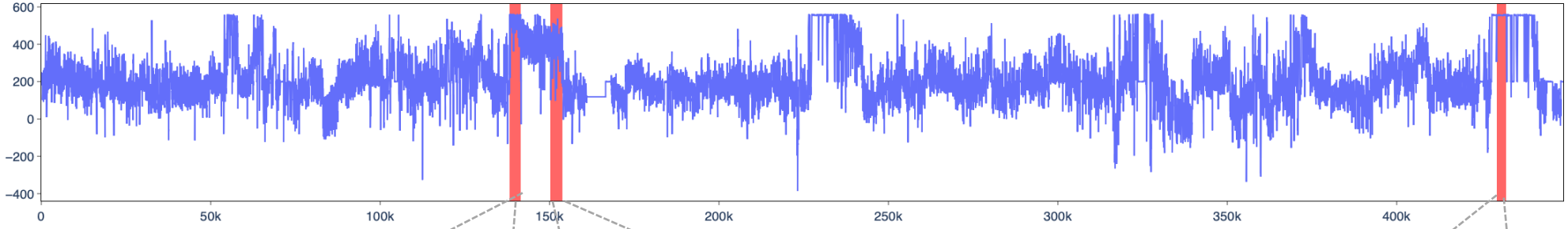
Stand



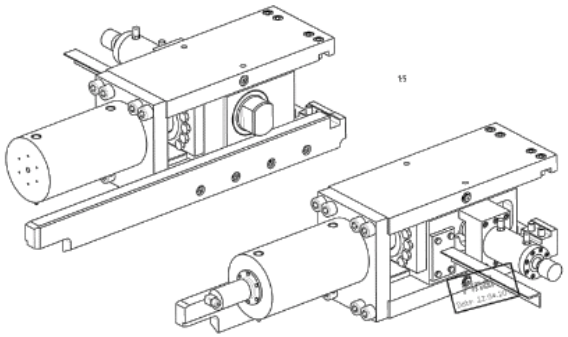
Destruction of CVC plates



Actual rolling bending force

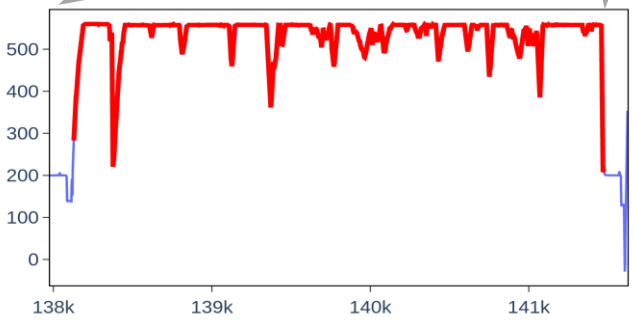


CVC (continuously variable curvature) system

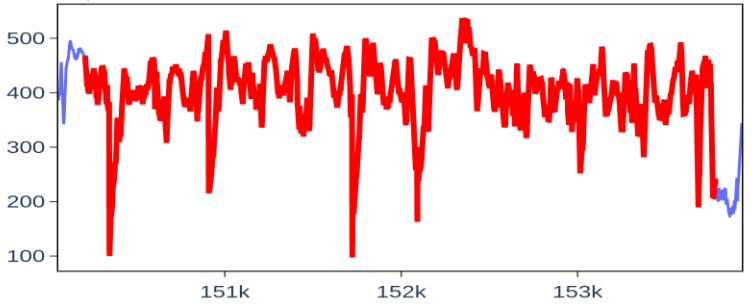


Top discords (range is 0.5-1.5 hours)

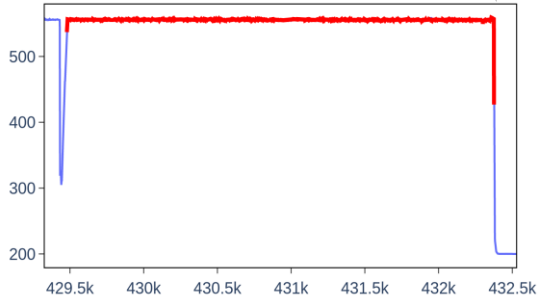
55 min



1 hour

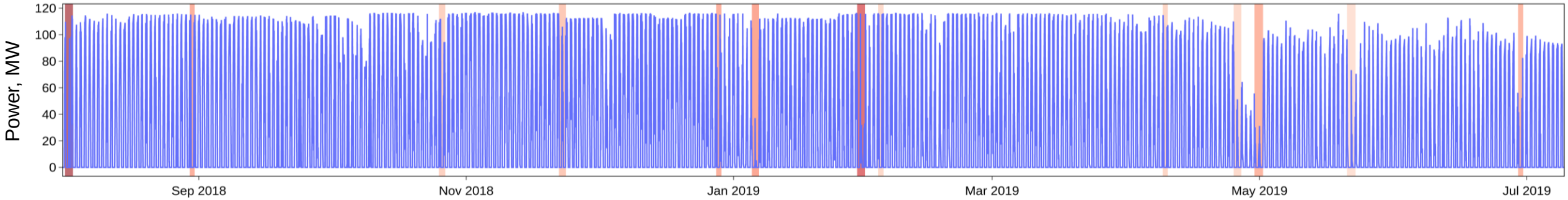


48 min



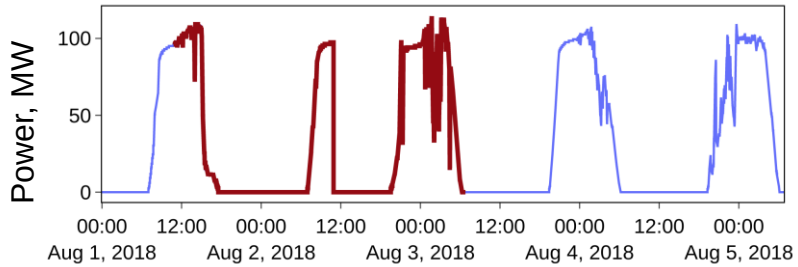
Case study: Solar energy

Solar energy production by AEMO (Australian Energy Market Operator) in Aug 2018 – Jul 2019 (frequency is once per 4 s)*

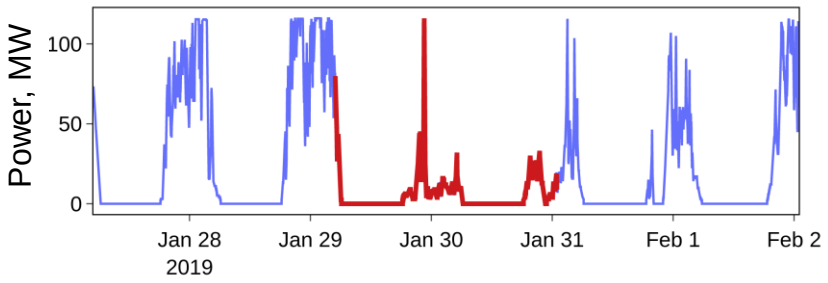


Top discords (range is 1-2 days)

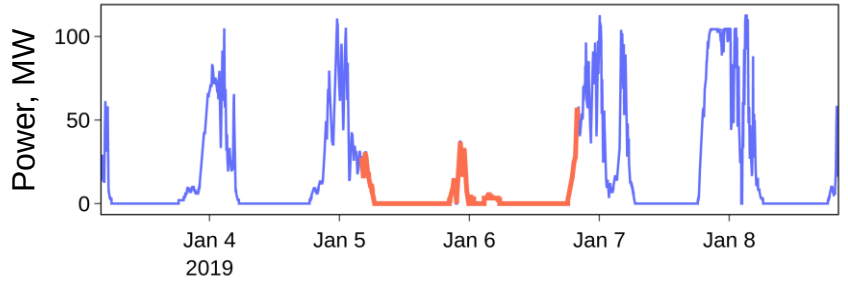
$m = 39\,600$ (1 day 20 hours)



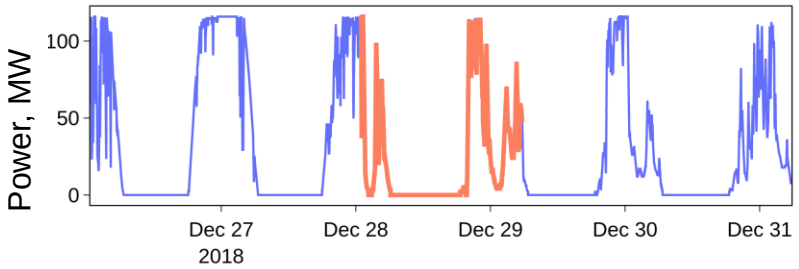
$m = 39\,600$ (1 day 20 hours)



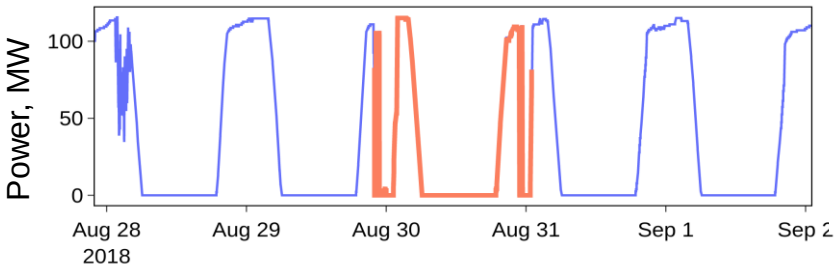
$m = 36\,000$ (1 day 16 hours)



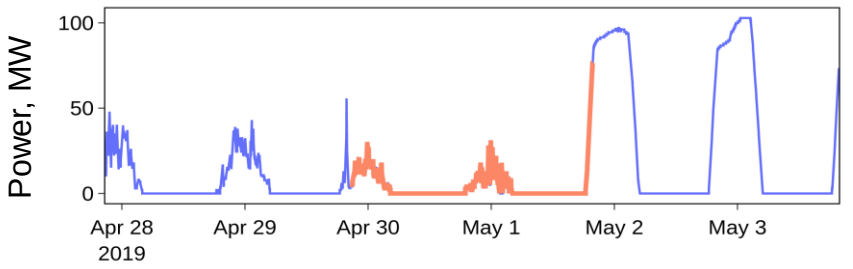
$m = 26\,100$ (1 day 5 hours)



$m = 24\,300$ (1 day 3 hours)



$m = 42\,300$ (1 day 23 hours)



* Godahewa R. *et al*. Solar Power Dataset (4 Seconds Observations). Zenodo, 2020. DOI: 10.5281/zenodo.4656027.

Do you have time series to find anomalies in? Then we're coming to you!

- **Discord** formalizes time series subsequence anomaly
- Serial discord discovery (E. Keogh *et al.*)
  - **DRAG** discovers single-length discords
  - **MERLIN** discovers all-length discords
- **Scalable parallel discord discovery (our research)**
  - **PD3** discovers single-length discords on GPU
  - **PALMAD** discovers all-length discords on GPU
  - **PADDi** discovers all-length discords on multi-GPU cluster

Thank you for your attention, we're ready to answer your questions

Mikhail Zymbler
Dr.Sci.,
Assoc. Prof.

mzym@susu.ru
https://mzym.susu.ru

Yana Kraeva
PhD student,
RF President stipend holder

kraevaya@susu.ru