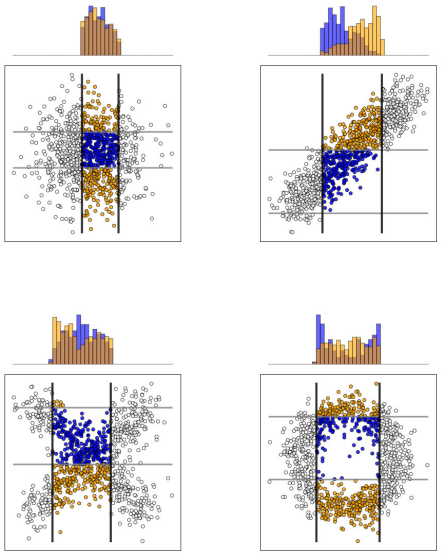
Karlsruhe Institute of Technology

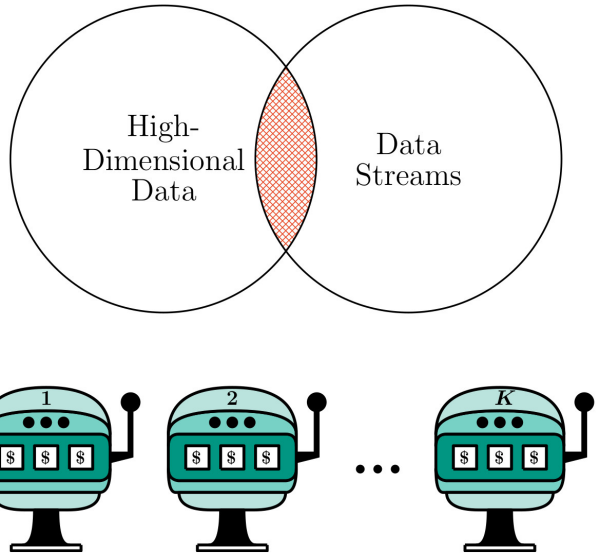
Estimating Dependency, Monitoring and Knowledge Discovery in High-Dimensional Data Streams

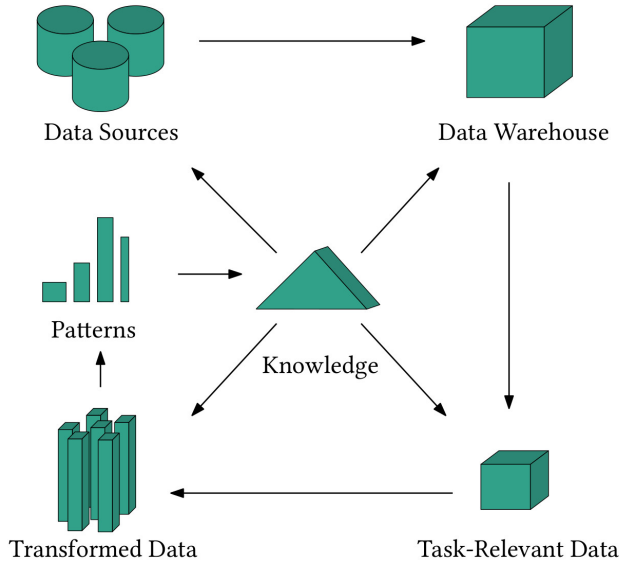
Doctoral Defense

Edouard Fouché | July 15, 2020

ADVISOR: PROF. DR.-ING. KLEMENS BÖHM







Motivations

Data Mining – or *Knowledge Discovery from Data (KDD)* – is the process of extracting knowledge from massive data sets.

- Data Mining has massive impact on our daily lives.

Modern tasks often involve data that:

- is *high-dimensional* (many attributes)
- comes as a *stream* (continuously collected)
- or both ! (high-dimensional + stream)

Motivations

Data Mining – or *Knowledge Discovery from Data (KDD)* – is the process of extracting knowledge from massive data sets.

- Data Mining has massive impact on our daily lives.

Modern tasks often involve data that:

- is *high-dimensional* (many attributes)
- comes as a *stream* (continuously collected)
- or both ! (high-dimensional + stream)

Motivations

Data Mining – or *Knowledge Discovery from Data (KDD)* – is the process of extracting knowledge from massive data sets.

- Data Mining has massive impact on our daily lives.

Modern tasks often involve data that:

- is *high-dimensional* (many attributes)
- comes as a *stream* (continuously collected)
- or both ! (high-dimensional + stream)

Motivations

Data Mining – or *Knowledge Discovery from Data (KDD)* – is the process of extracting knowledge from massive data sets.

- Data Mining has massive impact on our daily lives.

Modern tasks often involve data that:

- is *high-dimensional* (many attributes)
- comes as a *stream* (continuously collected)
- or both ! (high-dimensional + stream)

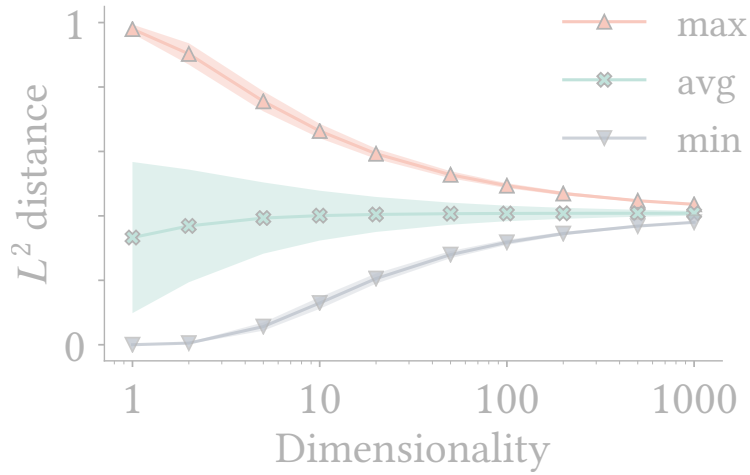
Challenges

We must deal with *High-Dimensional Data Streams (HD-DS)*.
→ They come with two sets of challenges:

High-Dimensionality

"*curse of dimensionality*" [Bel57]

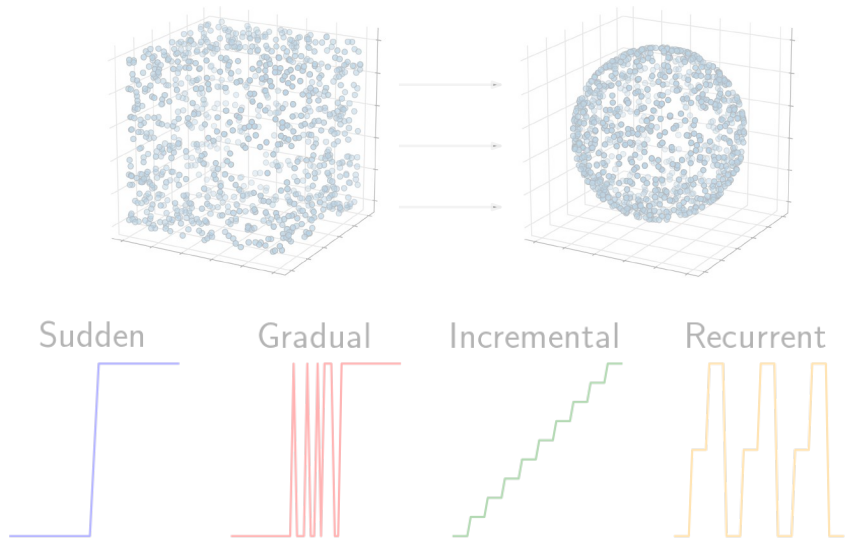
- The space becomes sparse
- Number of subspaces ↗
- *Robust*/*efficient* algorithms



Data Streams

"*concept drift*" [BGE15]

- Data may change over time
- *Adaptive* techniques



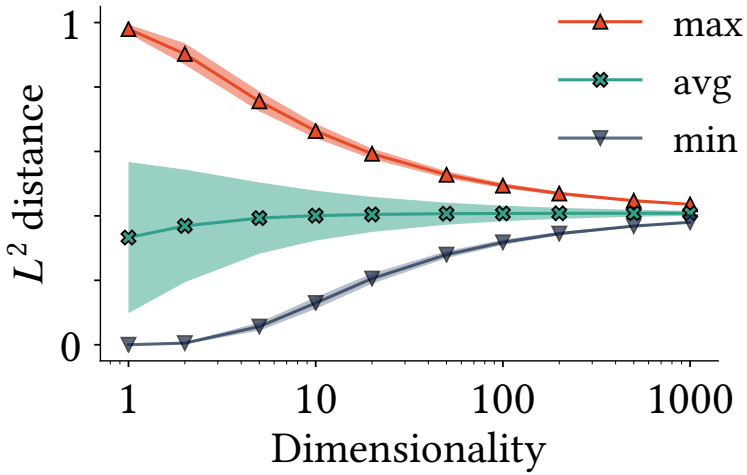
Challenges

We must deal with *High-Dimensional Data Streams (HD-DS)*.
$\rightarrow$ They come with two sets of challenges:

High-Dimensionality

"*curse of dimensionality*" [Bel57]

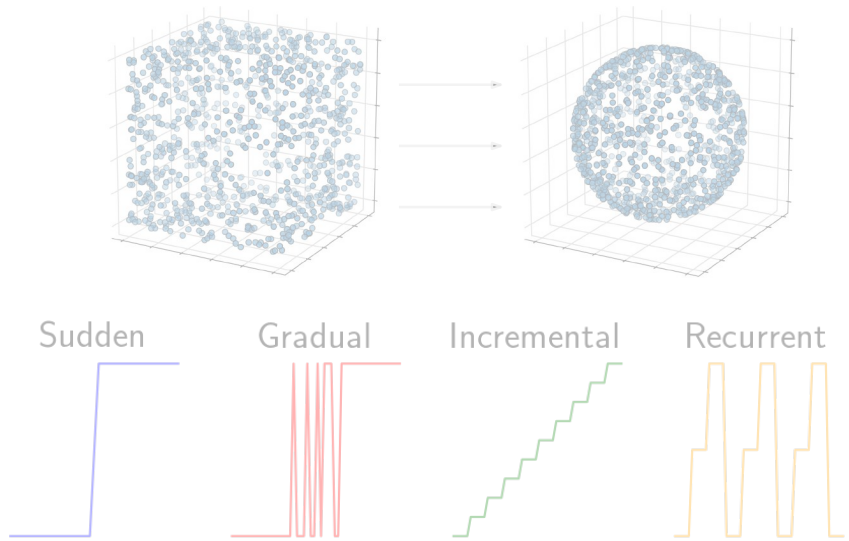
- The space becomes sparse
- Number of subspaces ↗
- *Robust*/*efficient* algorithms



Data Streams

"*concept drift*" [BGE15]

- Data may change over time
- *Adaptive* techniques



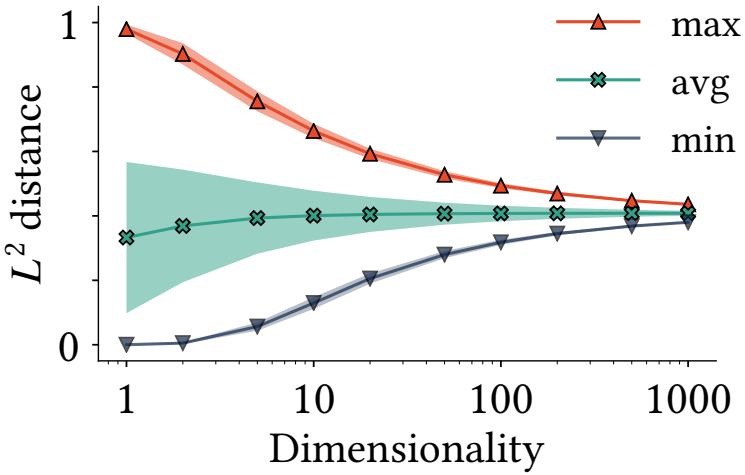
Challenges

We must deal with *High-Dimensional Data Streams (HD-DS)*.
$\rightarrow$ They come with two sets of challenges:

High-Dimensionality

"*curse of dimensionality*" [Bel57]

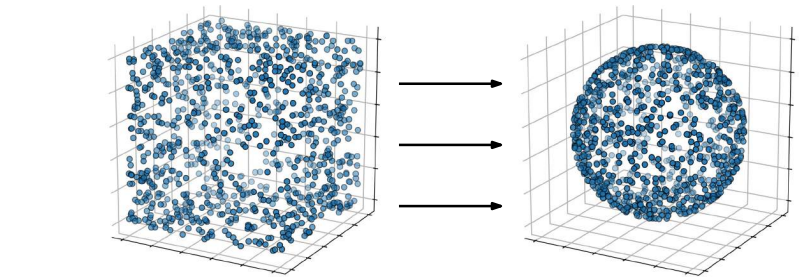
- The space becomes sparse
- Number of subspaces $\nearrow$
- *Robust*/*efficient* algorithms

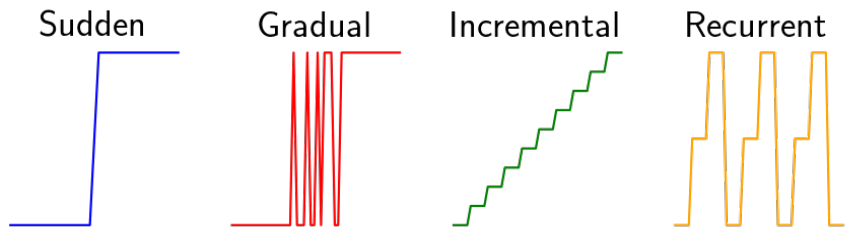


Data Streams

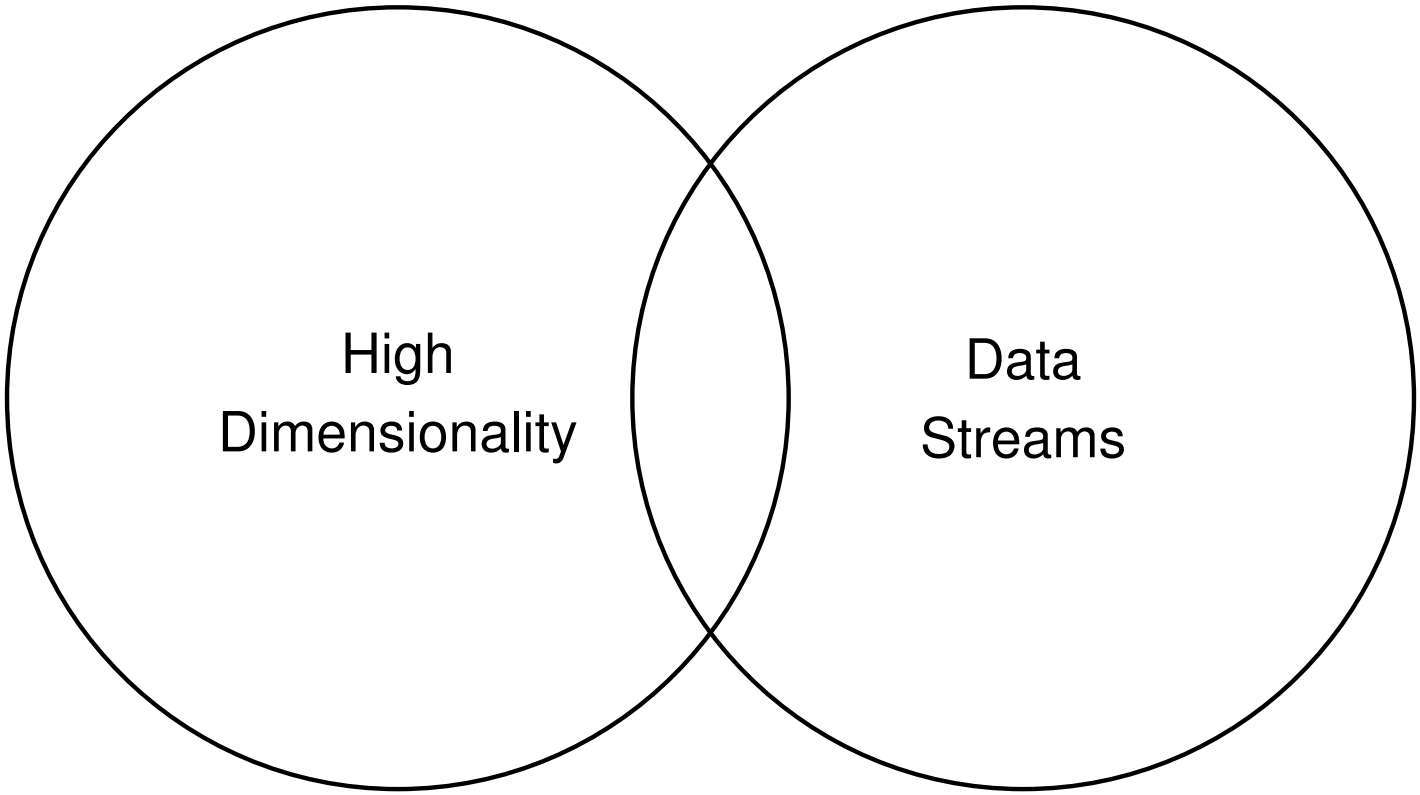
"*concept drift*" [BGE15]

- Data may change over time
- *Adaptive* techniques



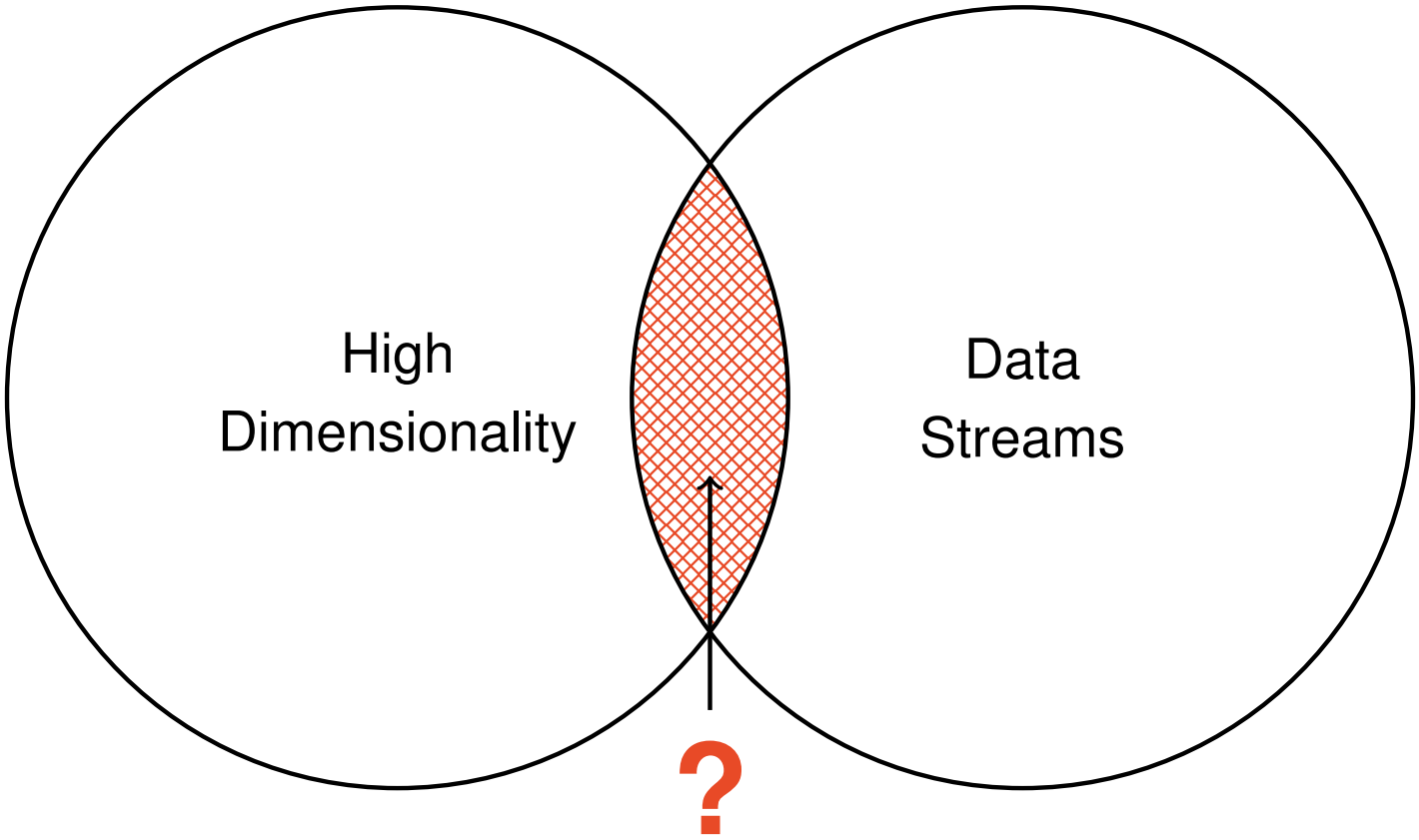


Focus of other works



This is the central interrogation of this dissertation

Focus of other works



This is the central interrogation of this dissertation

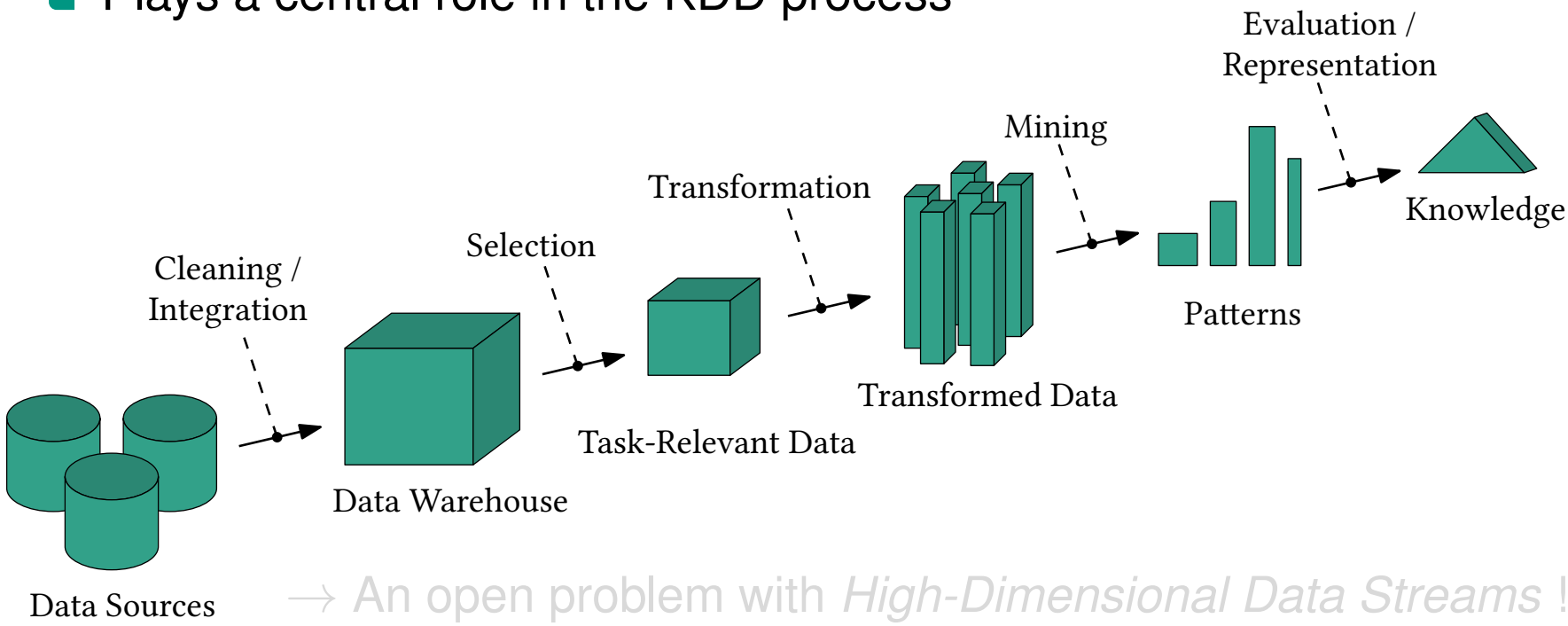
The Role of Dependency Estimation

- Dependency/Correlation:
  - Describes the strength of relationship between attributes
  - In practice: Can only be estimated from empirical observations
  - Examples: Pearson, Spearman, Mutual Information (MI), ...
- Plays a central role in the KDD process

→ An open problem with *High-Dimensional Data Streams* !

The Role of Dependency Estimation

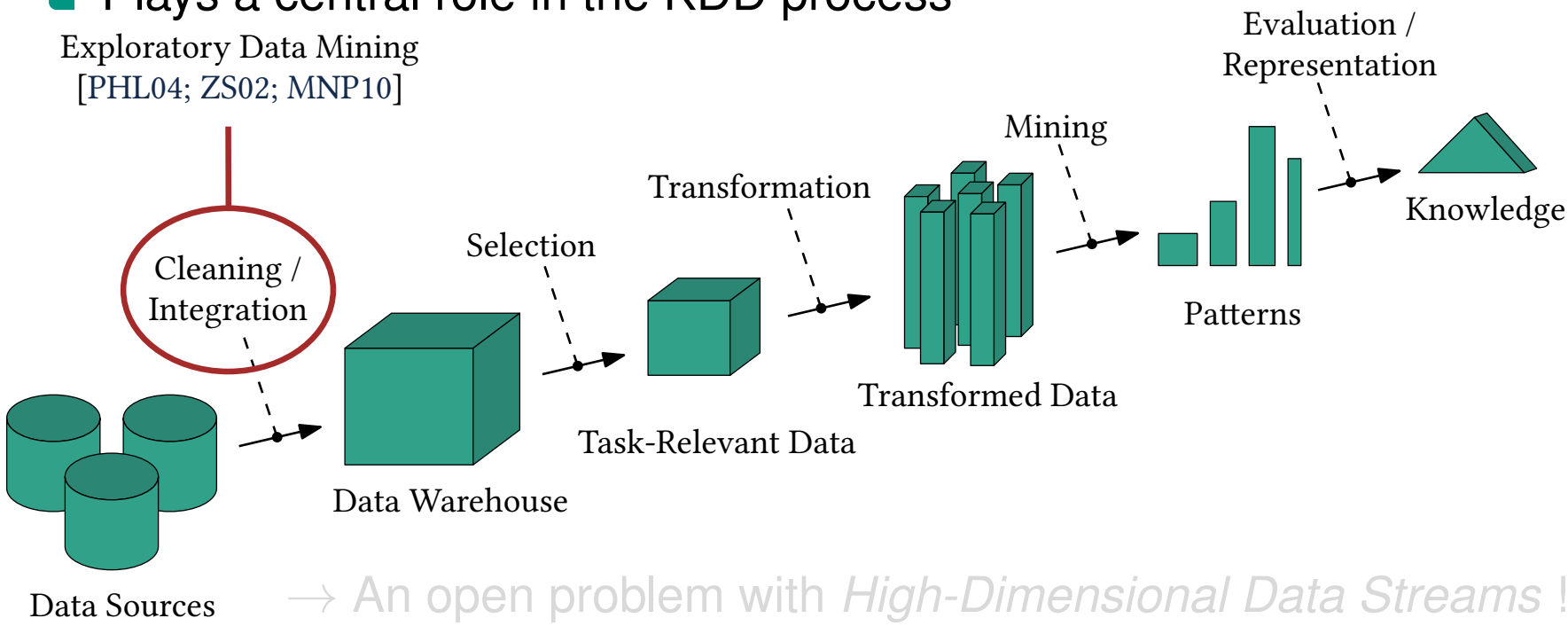
- Dependency/Correlation:
  - Describes the strength of relationship between attributes
  - In practice: Can only be estimated from empirical observations
  - Examples: Pearson, Spearman, Mutual Information (MI), ...
- Plays a central role in the KDD process



→ An open problem with *High-Dimensional Data Streams* !

The Role of Dependency Estimation

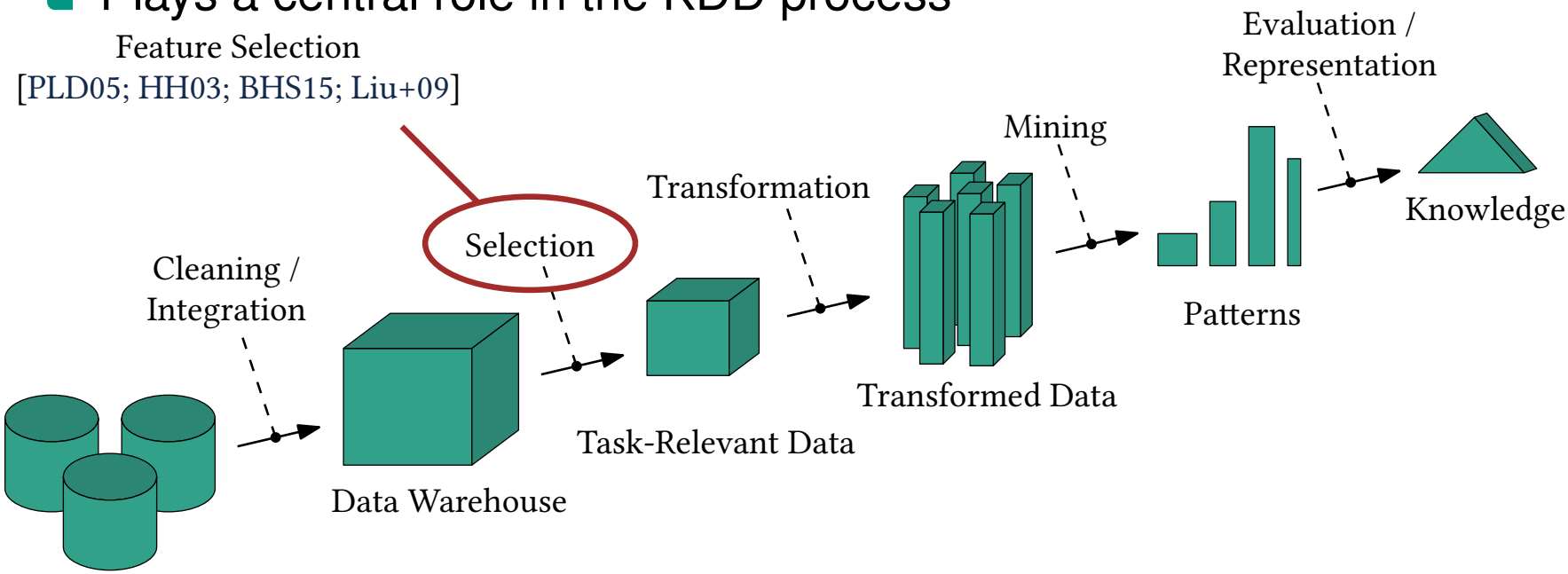
- Dependency/Correlation:
  - Describes the strength of relationship between attributes
  - In practice: Can only be estimated from empirical observations
  - Examples: Pearson, Spearman, Mutual Information (MI), ...
- Plays a central role in the KDD process



→ An open problem with *High-Dimensional Data Streams* !

The Role of Dependency Estimation

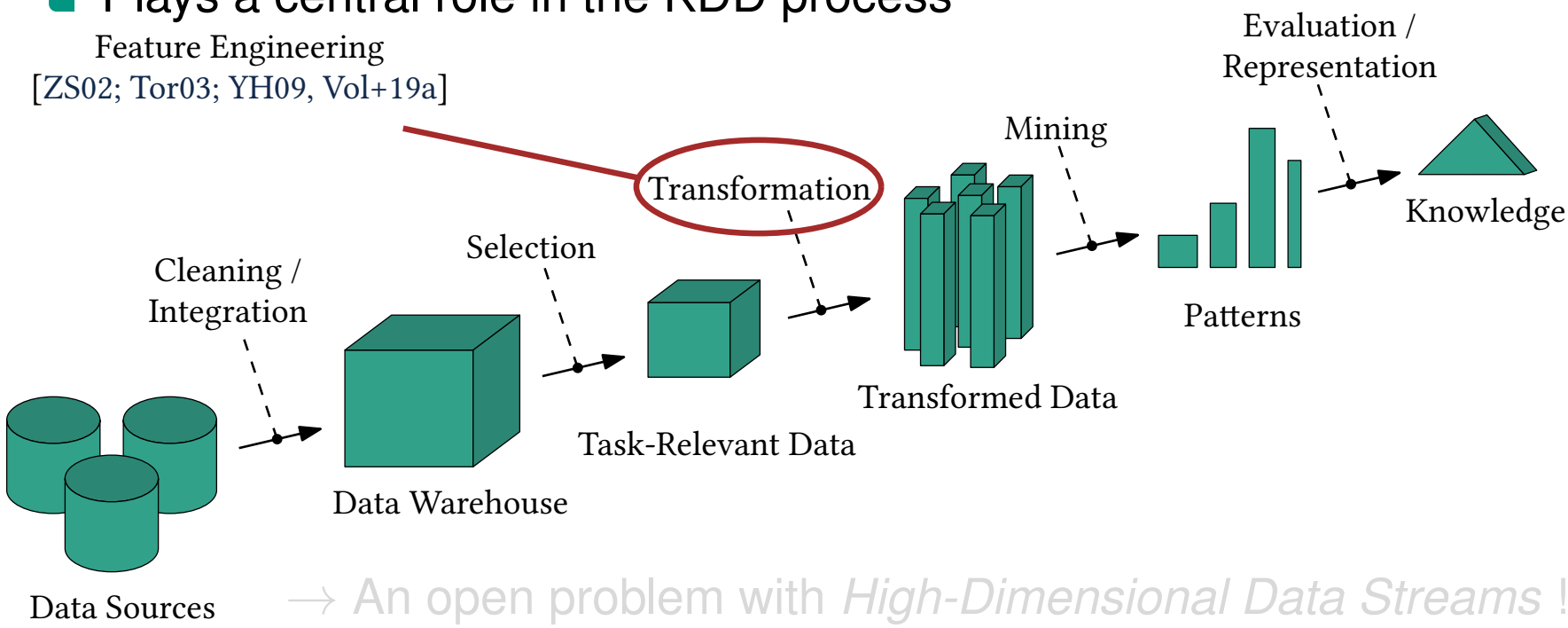
- Dependency/Correlation:
  - Describes the strength of relationship between attributes
  - In practice: Can only be estimated from empirical observations
  - Examples: Pearson, Spearman, Mutual Information (MI), ...
- Plays a central role in the KDD process



→ An open problem with *High-Dimensional Data Streams* !

The Role of Dependency Estimation

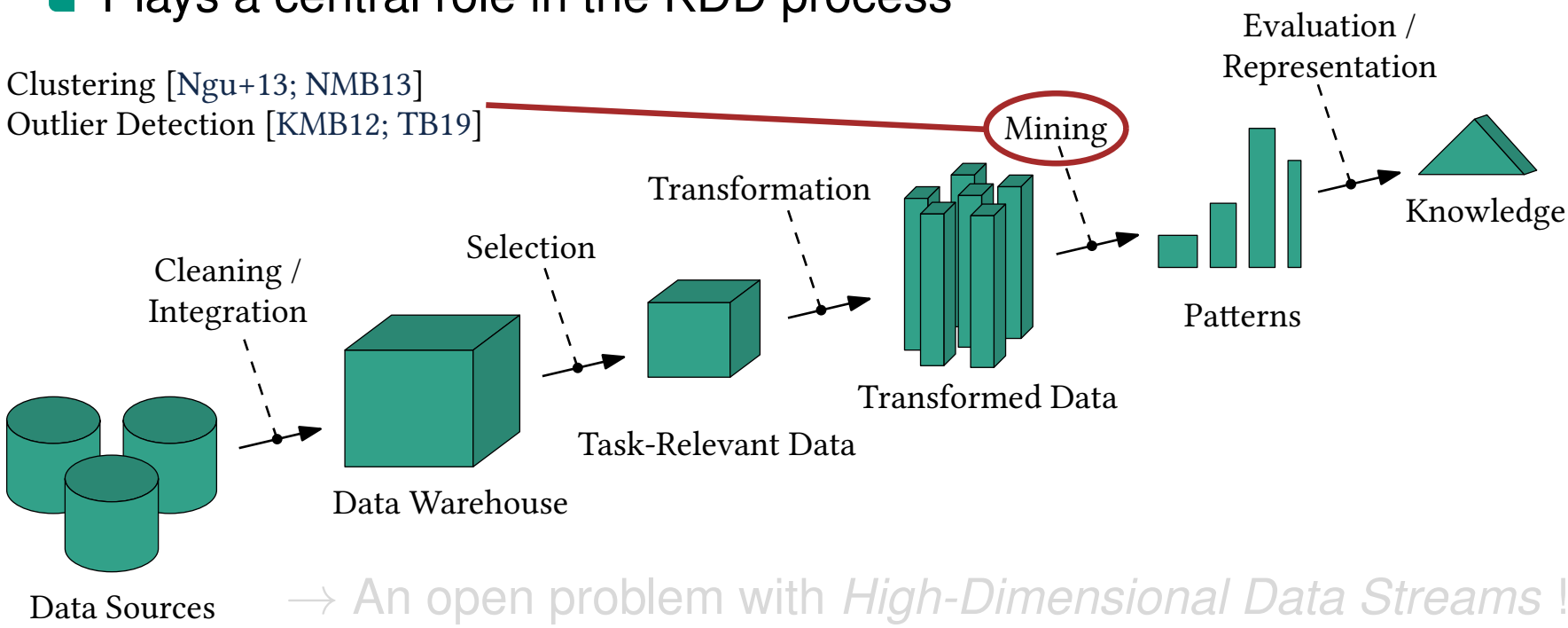
- Dependency/Correlation:
  - Describes the strength of relationship between attributes
  - In practice: Can only be estimated from empirical observations
  - Examples: Pearson, Spearman, Mutual Information (MI), ...
- Plays a central role in the KDD process



→ An open problem with *High-Dimensional Data Streams* !

The Role of Dependency Estimation

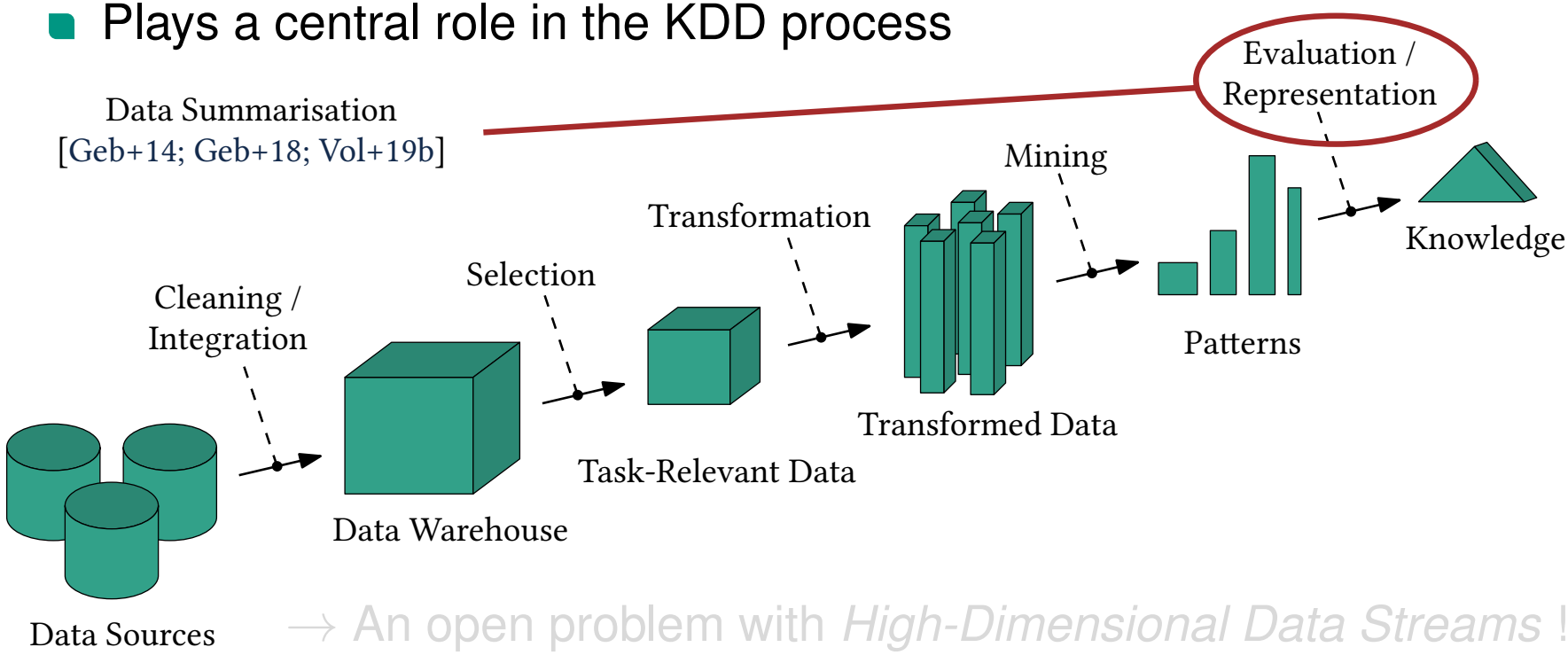
- Dependency/Correlation:
  - Describes the strength of relationship between attributes
  - In practice: Can only be estimated from empirical observations
  - Examples: Pearson, Spearman, Mutual Information (MI), ...
- Plays a central role in the KDD process



→ An open problem with *High-Dimensional Data Streams* !

The Role of Dependency Estimation

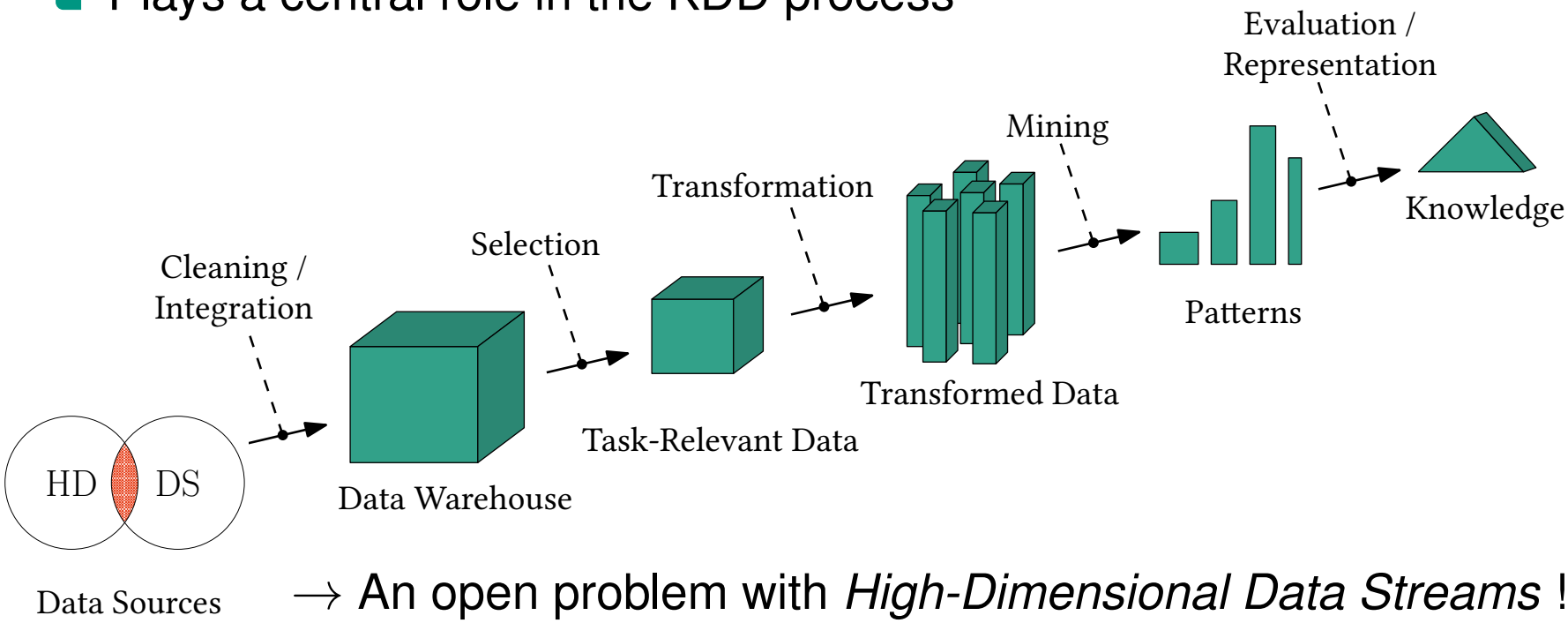
- Dependency/Correlation:
  - Describes the strength of relationship between attributes
  - In practice: Can only be estimated from empirical observations
  - Examples: Pearson, Spearman, Mutual Information (MI), ...
- Plays a central role in the KDD process



→ An open problem with *High-Dimensional Data Streams* !

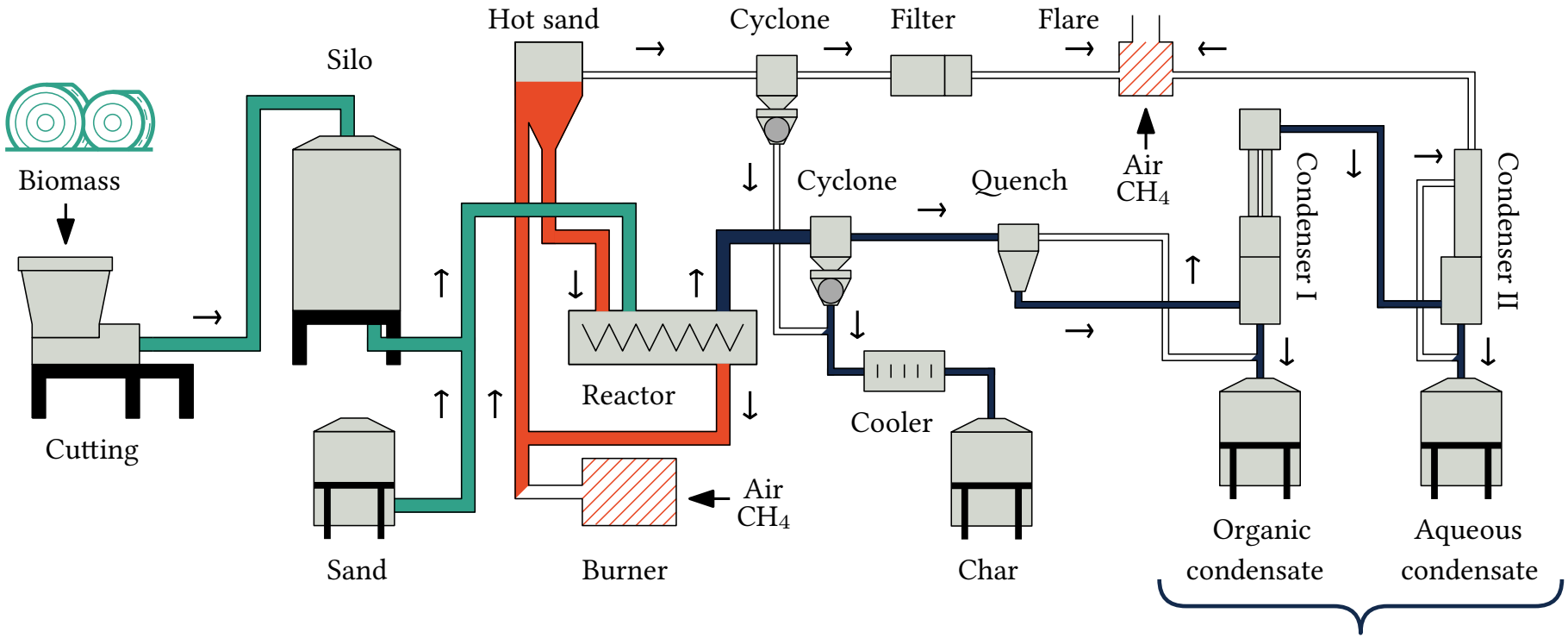
The Role of Dependency Estimation

- Dependency/Correlation:
  - Describes the strength of relationship between attributes
  - In practice: Can only be estimated from empirical observations
  - Examples: Pearson, Spearman, Mutual Information (MI), ...
- Plays a central role in the KDD process



→ An open problem with *High-Dimensional Data Streams* !

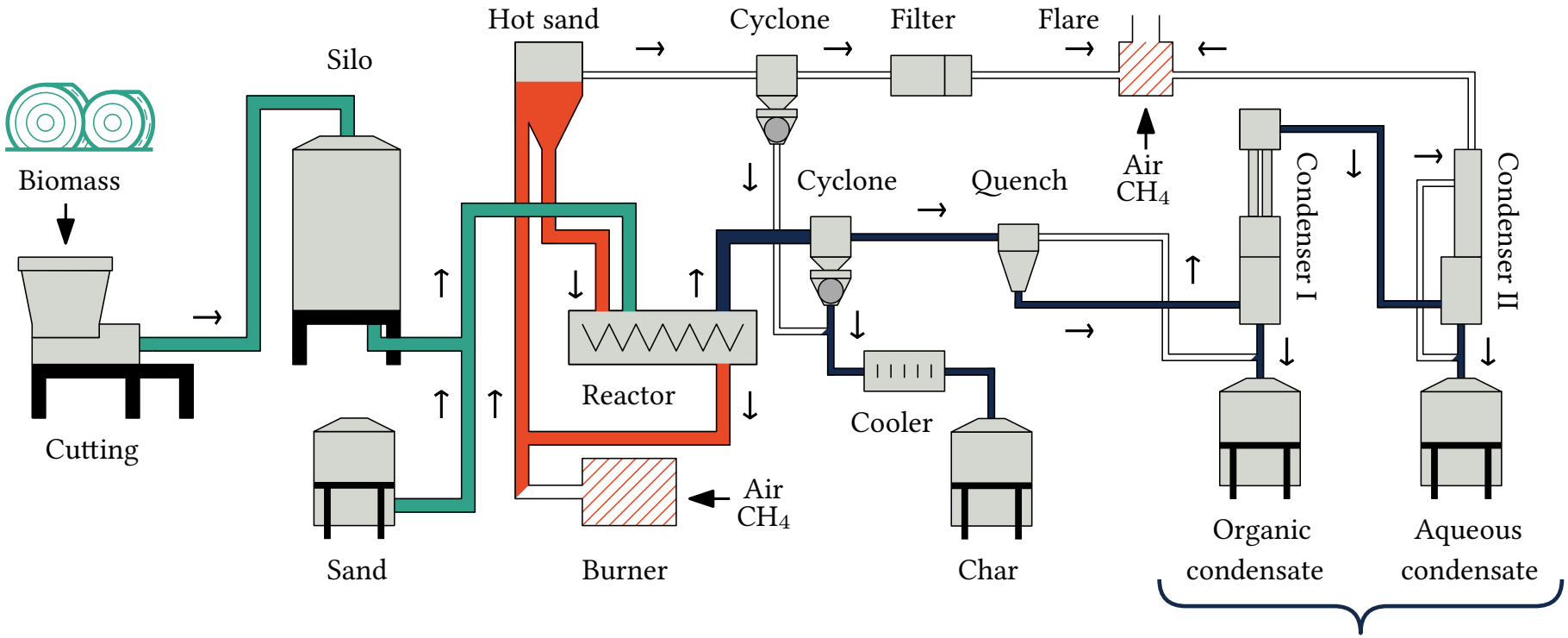
Bioliq®: The pyrolysis as a HD-DS



Simplified schematics of Bioliq's Pyrolysis

- $>$ 400 sensors, continuously measured $\rightarrow$ HD-DS
- Data Mining may help to analyse the pyrolysis process

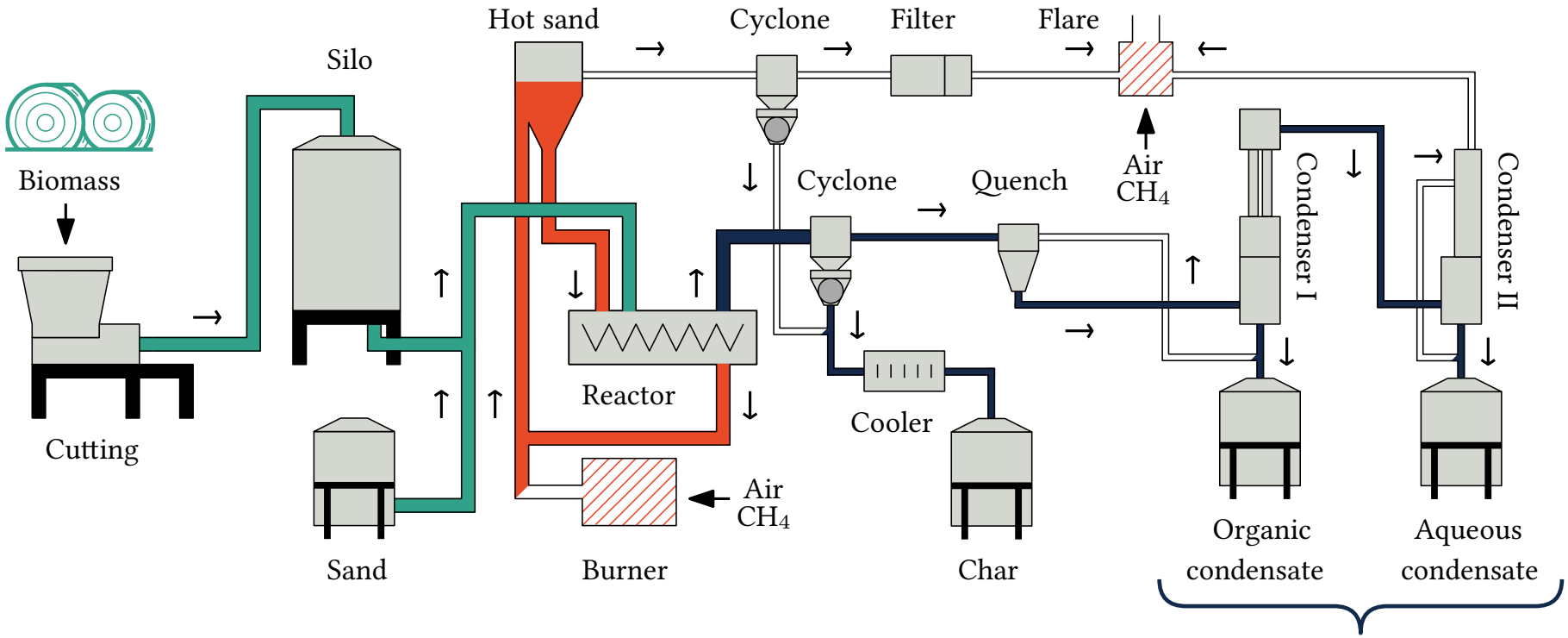
Bioliq®: The pyrolysis as a HD-DS



Simplified schematics of Bioliq's Pyrolysis

- $>$ 400 sensors, continuously measured $\rightarrow$ HD-DS
- Data Mining may help to analyse the pyrolysis process

Bioliq®: The pyrolysis as a HD-DS

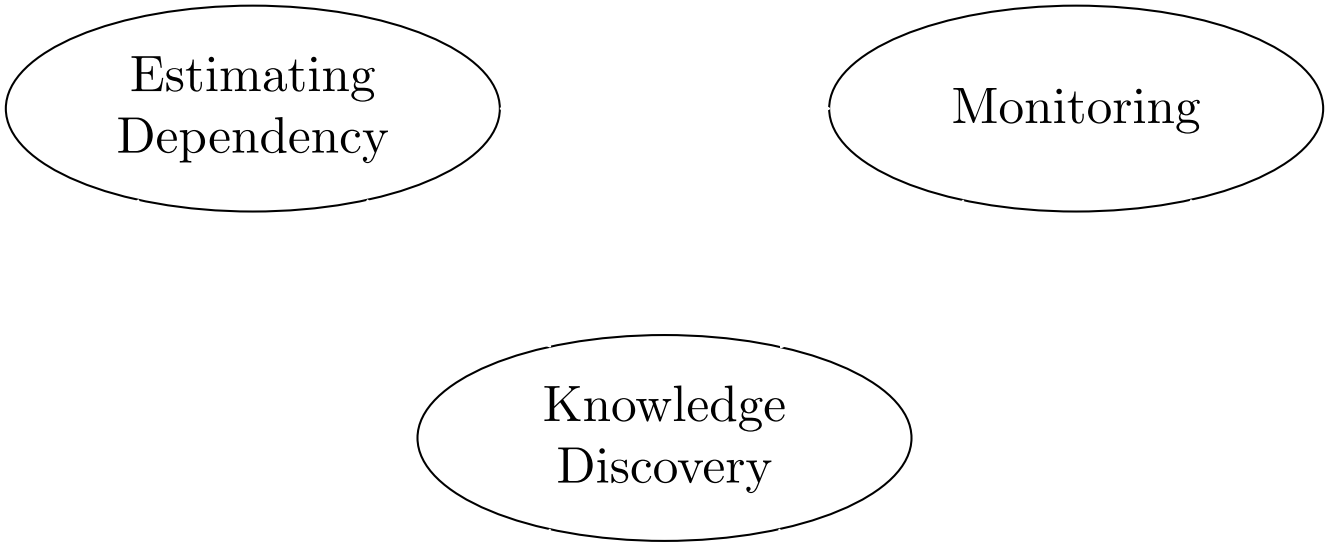


Simplified schematics of Bioliq's Pyrolysis

- $>$ 400 sensors, continuously measured $\rightarrow$ HD-DS
- Data Mining may help to analyse the pyrolysis process

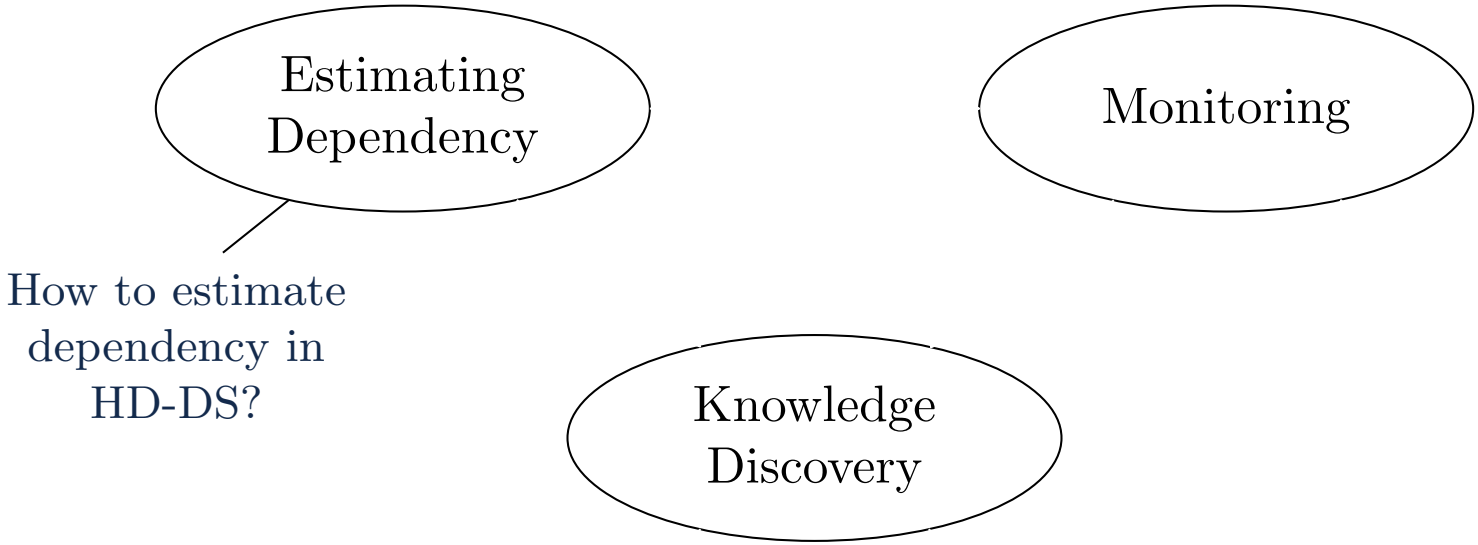
Our Contributions

are organised around 3 research questions:



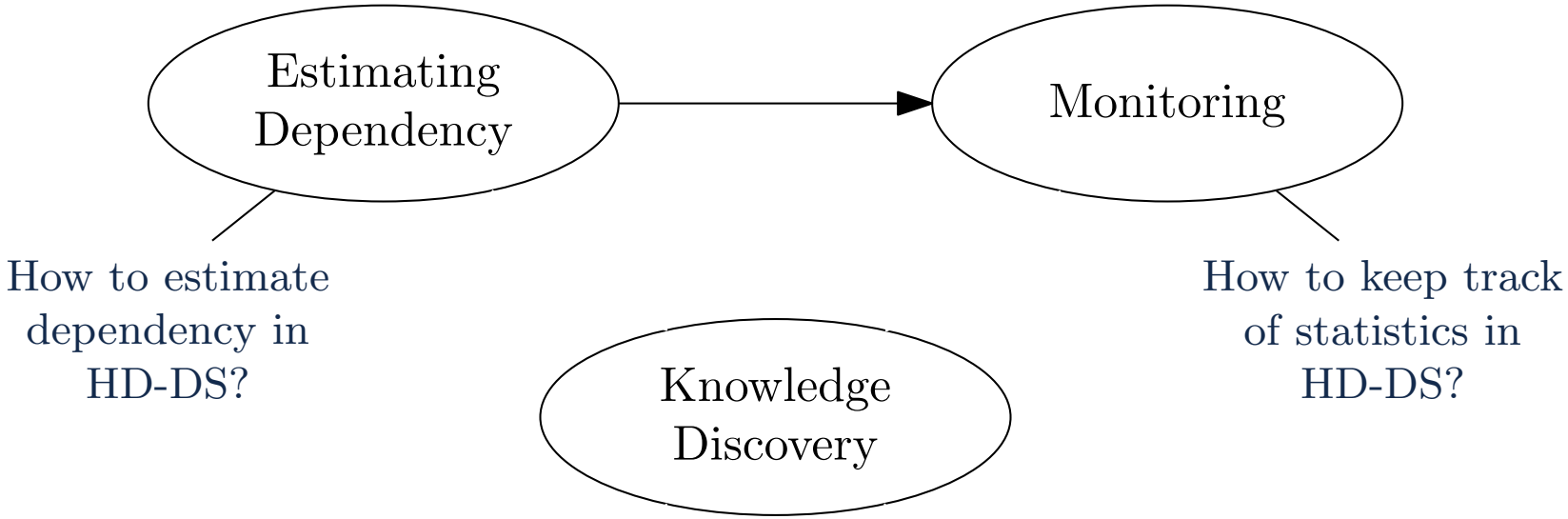
Our Contributions

are organised around 3 research questions:



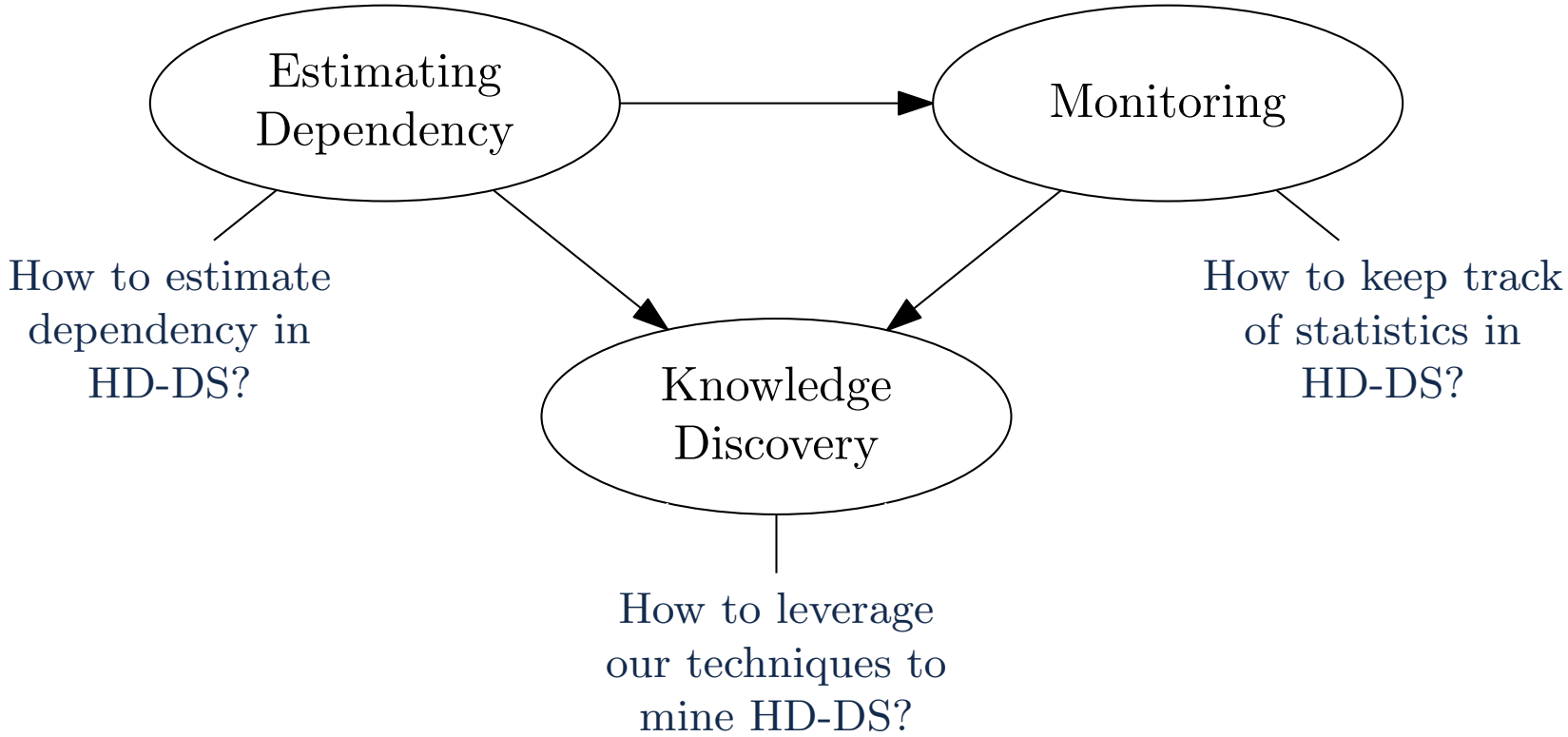
Our Contributions

are organised around 3 research questions:



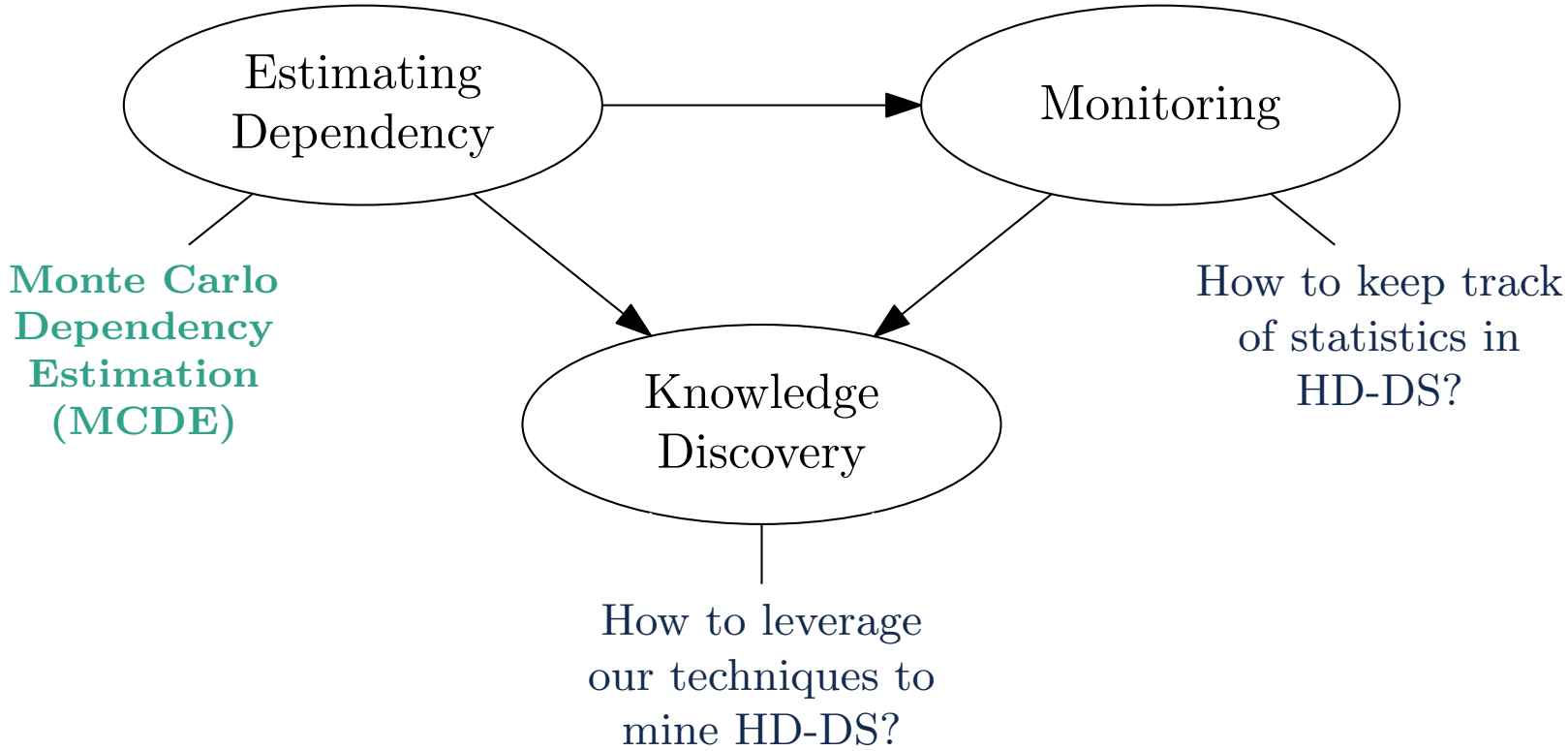
Our Contributions

are organised around 3 research questions:



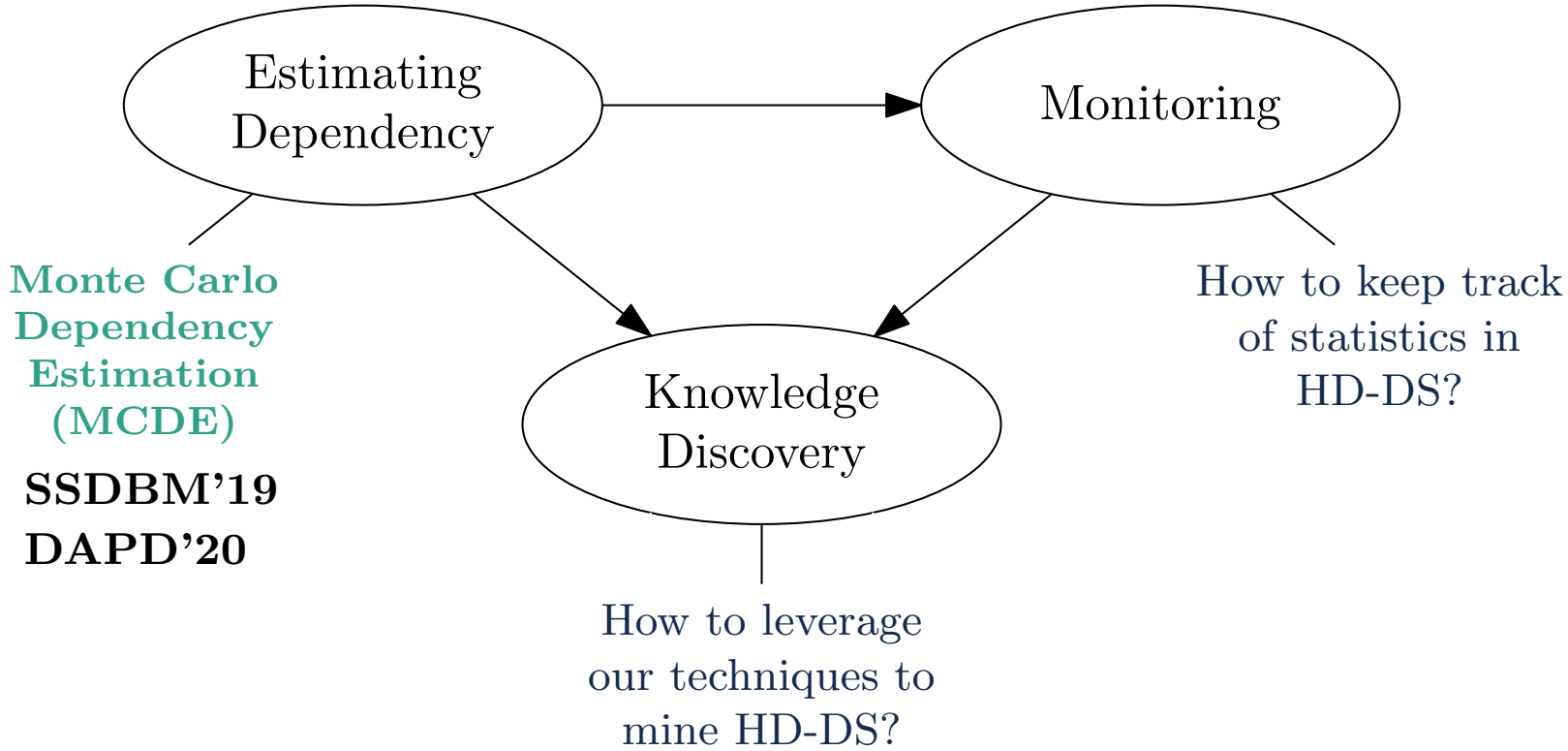
Our Contributions

are organised around 3 research questions:



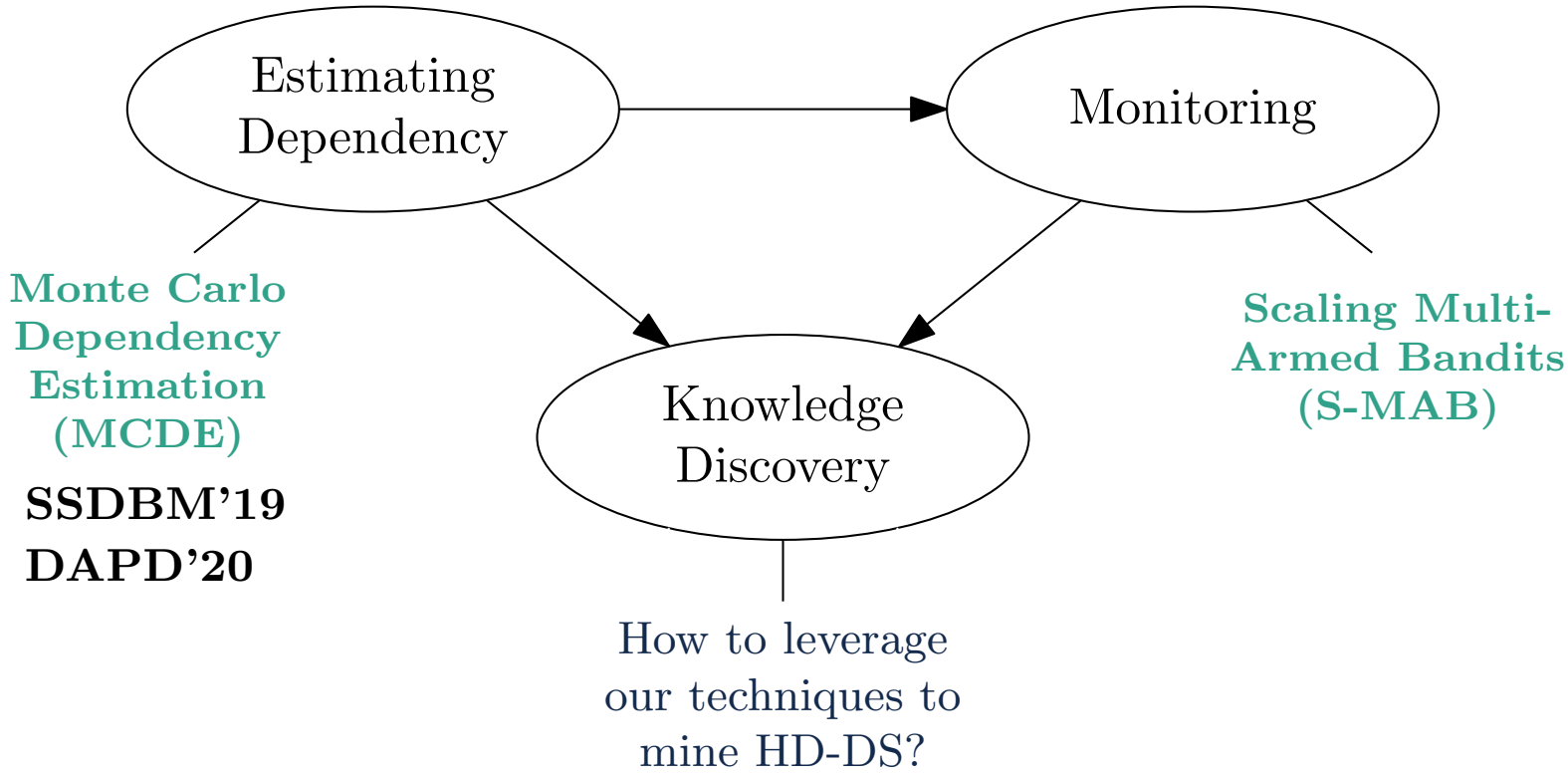
Our Contributions

are organised around 3 research questions:



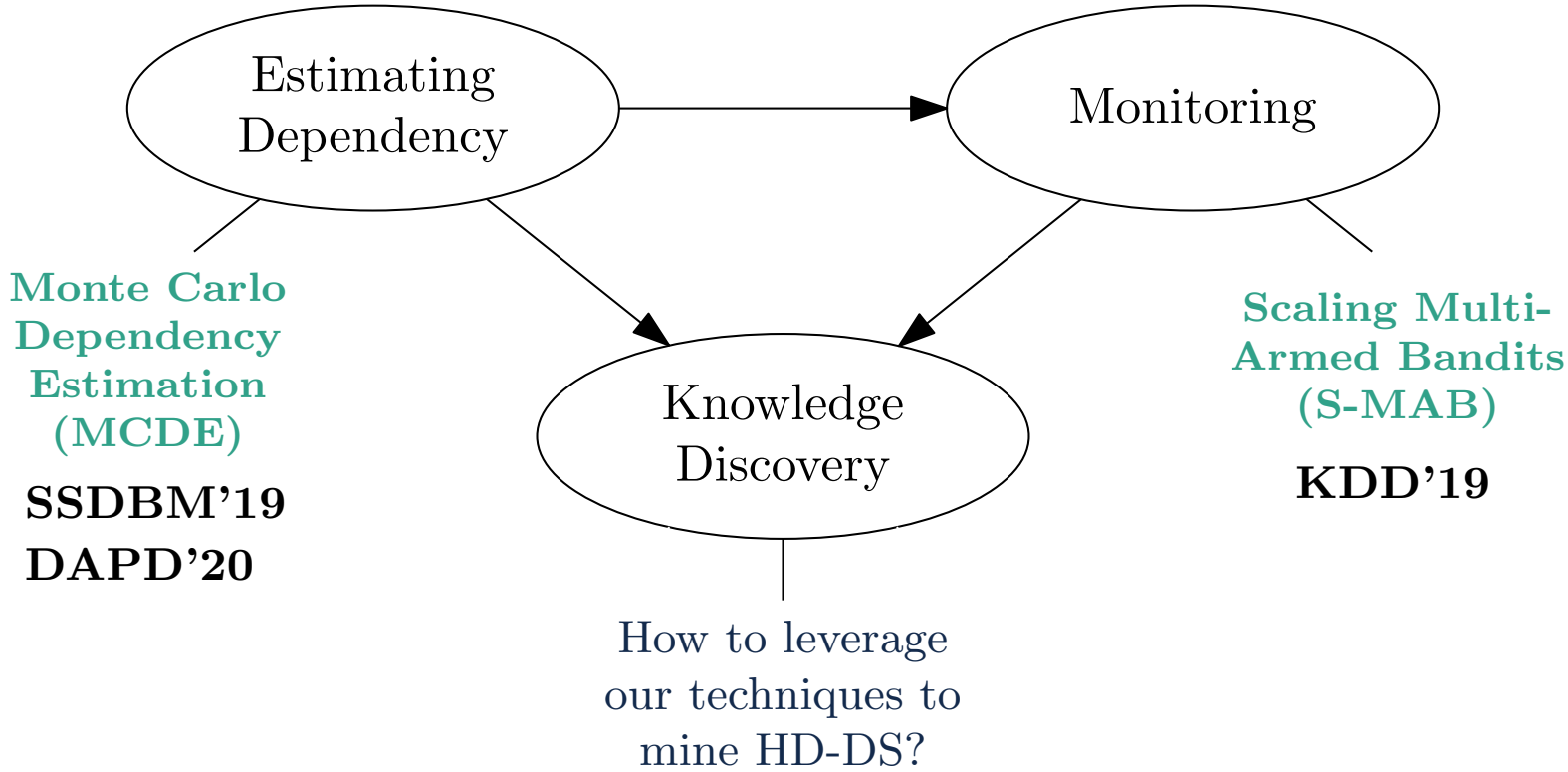
Our Contributions

are organised around 3 research questions:



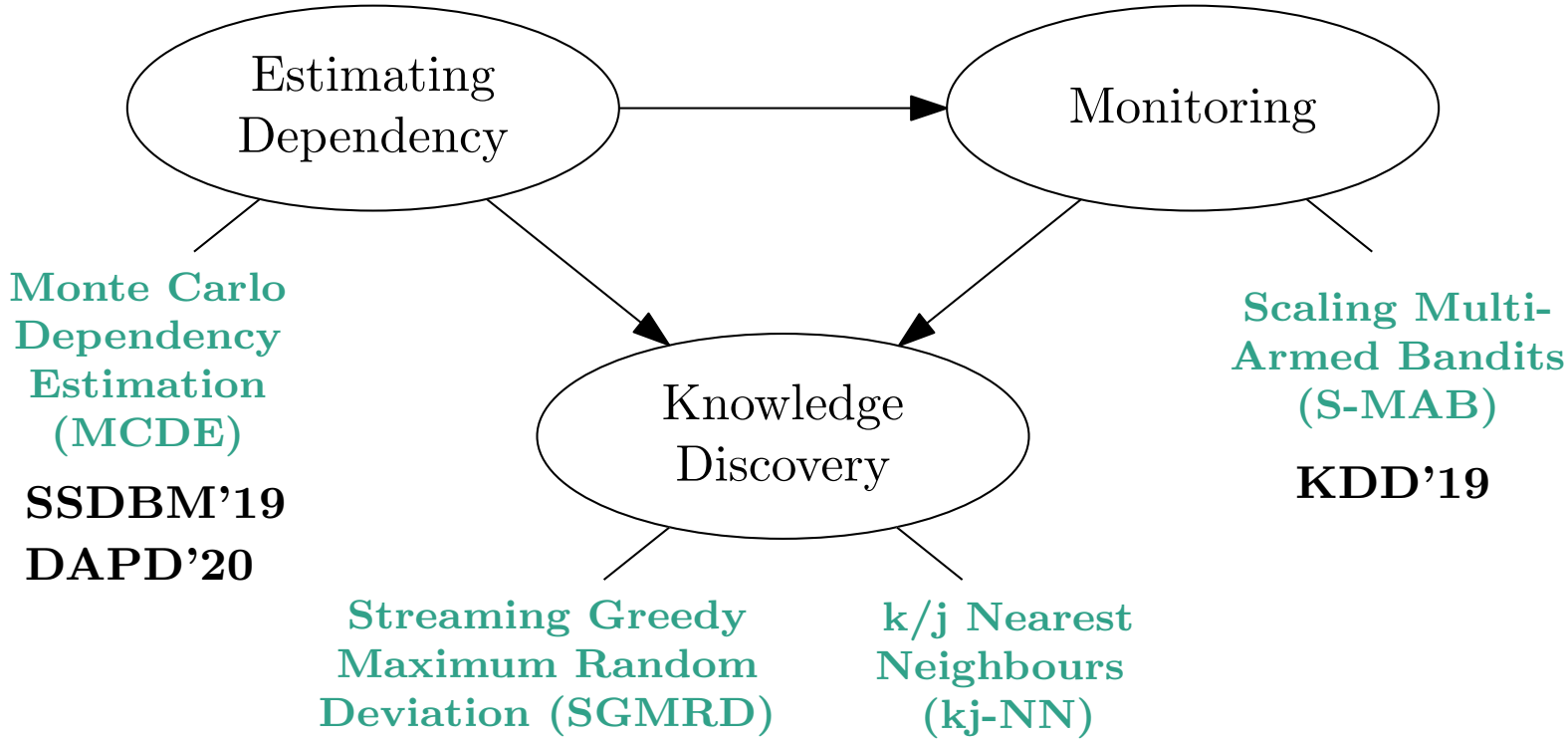
Our Contributions

are organised around 3 research questions:



Our Contributions

are organised around 3 research questions:



Monte Carlo Dependency Estimation

For any given subspace $S = \{X_1, \ldots, X_d\}$:

- We see each dimension $X_i \in S$ as a random variable:

$$\overbrace{p(S) = \prod_{X_i \in S} p_{X_i}(S)}^{independence} \quad \Rightarrow \quad \forall X_i \in S, \overbrace{p_{X_i}(S)}^{marginal} = \underbrace{p(S|\overline{X_i})}_{conditional}$$

- Then $\neg B \Rightarrow \neg A$ (non-independence)
  - Our idea: quantify $\neg B$ as $discrepancy(p_{X_i}(S), p(S|\overline{X_i})), \forall X_i \in S$
  - We propose a Monte Carlo method to approximate the *discrepancy*

Monte Carlo Dependency Estimation

For any given subspace $S = \{X_1, \ldots, X_d\}$:

- We see each dimension $X_i \in S$ as a random variable:

$$\overbrace{p(S) = \prod_{X_i \in S} p_{X_i}(S)}^{independence} \quad \Rightarrow \quad \forall X_i \in S, \overbrace{p_{X_i}(S)}^{marginal} = \underbrace{p(S|\overline{X_i})}_{conditional}$$

- Then $\neg B \Rightarrow \neg A$ (non-independence)
  - Our idea: quantify $\neg B$ as $discrepancy(p_{X_i}(S), p(S|\overline{X_i})), \forall X_i \in S$
  - We propose a Monte Carlo method to approximate the *discrepancy*

Monte Carlo Dependency Estimation

For any given subspace $S = \{X_1, \ldots, X_d\}$:

- We see each dimension $X_i \in S$ as a random variable:

$$\underbrace{\overbrace{p(S) = \prod_{X_i \in S} p_{X_i}(S)}^{independence}}_{A} \Rightarrow \underbrace{\forall X_i \in S, \overbrace{p_{X_i}(S)}^{marginal} = \underbrace{p(S|\overline{X_i})}_{conditional}}_{B}$$

- Then $\neg B \Rightarrow \neg A$ (non-independence)
  - Our idea: quantify $\neg B$ as $discrepancy(p_{X_i}(S), p(S|\overline{X_i})), \forall X_i \in S$
  - We propose a Monte Carlo method to approximate the *discrepancy*

Monte Carlo Dependency Estimation

For any given subspace $S = \{X_1, \ldots, X_d\}$:

- We see each dimension $X_i \in S$ as a random variable:

$$\underbrace{\overbrace{p(S) = \prod_{X_i \in S} p_{X_i}(S)}^{independence}}_{A} \quad \Rightarrow \quad \underbrace{\forall X_i \in S, \overbrace{p_{X_i}(S)}^{marginal} = \underbrace{p(S|\overline{X_i})}_{conditional}}_{B}$$

- Then $\neg B \Rightarrow \neg A$ (non-independence)
  - Our idea: quantify $\neg B$ as $discrepancy(p_{X_i}(S), p(S|\overline{X_i})), \forall X_i \in S$
  - We propose a Monte Carlo method to approximate the *discrepancy*

Monte Carlo Dependency Estimation

For any given subspace $S = \{X_1, \ldots, X_d\}$:

- We see each dimension $X_i \in S$ as a random variable:

$$\underbrace{\overbrace{p(S) = \prod_{X_i \in S} p_{X_i}(S)}^{independence}}_{A} \Rightarrow \underbrace{\forall X_i \in S, \overbrace{p_{X_i}(S)}^{marginal} = \underbrace{p(S|\overline{X_i})}_{conditional}}_{B}$$

- Then $\neg B \Rightarrow \neg A$ (non-independence)
  - Our idea: quantify $\neg B$ as *discrepancy*$(p_{X_i}(S), p(S|\overline{X_i})), \forall X_i \in S$
  - We propose a Monte Carlo method to approximate the *discrepancy*

Monte Carlo Dependency Estimation

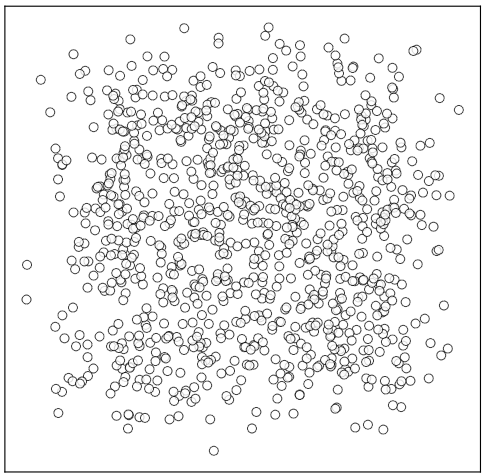
For any given subspace $S = \{X_1, \ldots, X_d\}$:

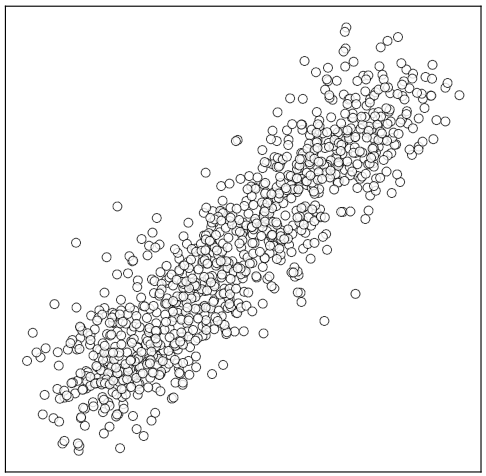
- We see each dimension $X_i \in S$ as a random variable:

$$\underbrace{\overbrace{p(S) = \prod_{X_i \in S} p_{X_i}(S)}^{independence}}_{A} \Rightarrow \underbrace{\forall X_i \in S, \overbrace{p_{X_i}(S)}^{marginal} = \underbrace{p(S|\overline{X_i})}_{conditional}}_{B}$$

- Then $\neg B \Rightarrow \neg A$ (non-independence)
  - Our idea: quantify $\neg B$ as *discrepancy*$(p_{X_i}(S), p(S|\overline{X_i})), \forall X_i \in S$
  - We propose a Monte Carlo method to approximate the *discrepancy*

Monte Carlo Dependency Estimation





- Take a random slice
- Test $\mathcal{T}(\hat{p}_{X_i}(S), \hat{p}(S|\overline{X_i}))$
- Repeat M times
- Average:

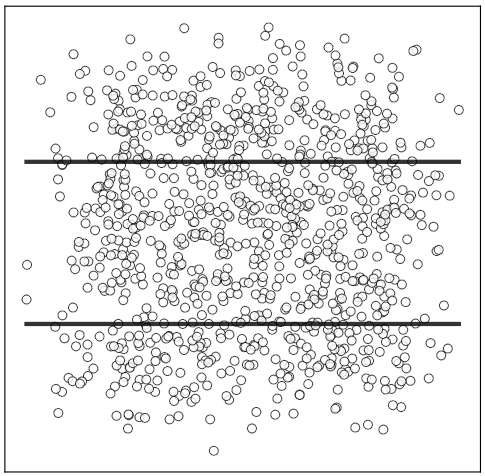
$$\mathcal{C} = \frac{1}{M}\sum_{m=1}^{M}(1 - p\text{-}value)$$

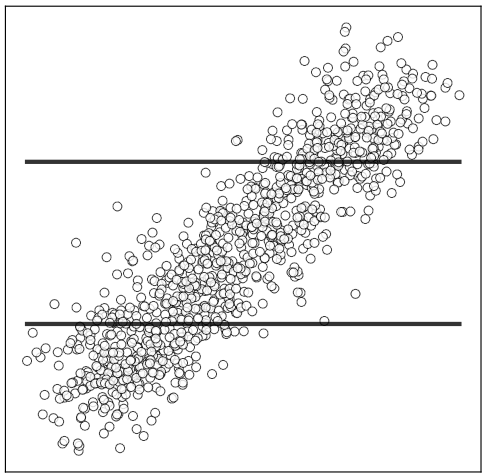
Properties:

- Independence $\Rightarrow \mathbb{E}[\mathcal{C}] = 0.5$, otherwise greater
- $\mathcal{C} \in [0, 1]$

Theorem (derived from [Hoe63]): $\Pr\left(|\mathcal{C} - \mathbb{E}[\mathcal{C}]| \geq \varepsilon\right) \leq 2e^{-2M\varepsilon^2}$

Monte Carlo Dependency Estimation





- Take a random slice
- Test $\mathcal{T}(\hat{p}_{X_i}(S), \hat{p}(S|\overline{X_i}))$
- Repeat M times
- Average:

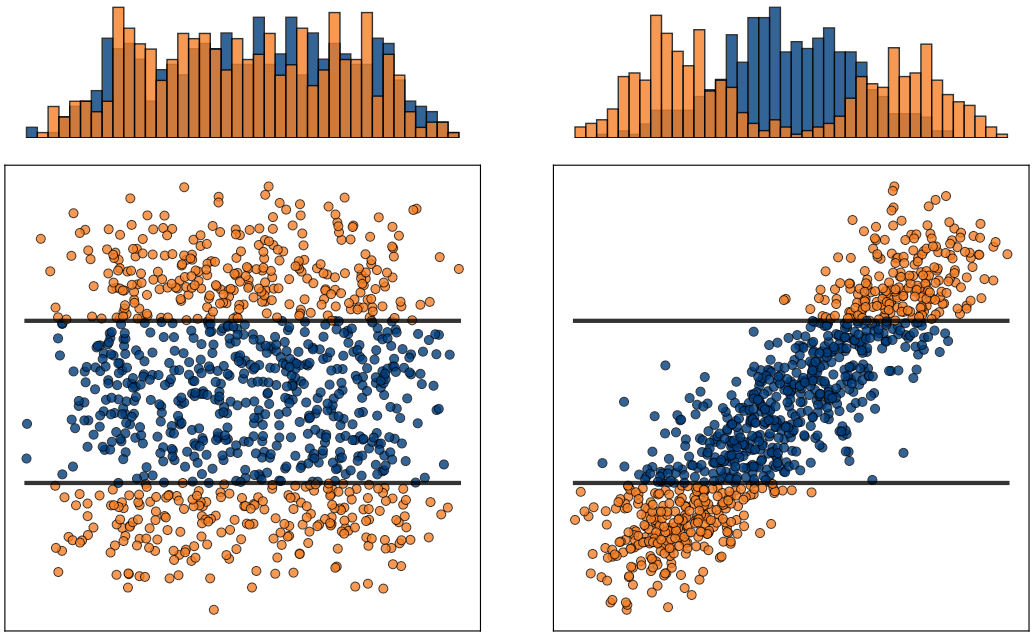
$$\mathcal{C} = \frac{1}{M}\sum_{m=1}^{M}(1 - p\text{-}value)$$

Properties:

- Independence $\Rightarrow \mathbb{E}[\mathcal{C}] = 0.5$, otherwise greater
- $\mathcal{C} \in [0, 1]$

Theorem (derived from [Hoe63]): $\Pr\left(|\mathcal{C} - \mathbb{E}[\mathcal{C}]| \geq \varepsilon\right) \leq 2e^{-2M\varepsilon^2}$

Monte Carlo Dependency Estimation



- Take a random slice
- Test $\mathcal{T}(\hat{p}_{X_i}(S), \hat{p}(S|\overline{X_i}))$
- Repeat M times
- Average:

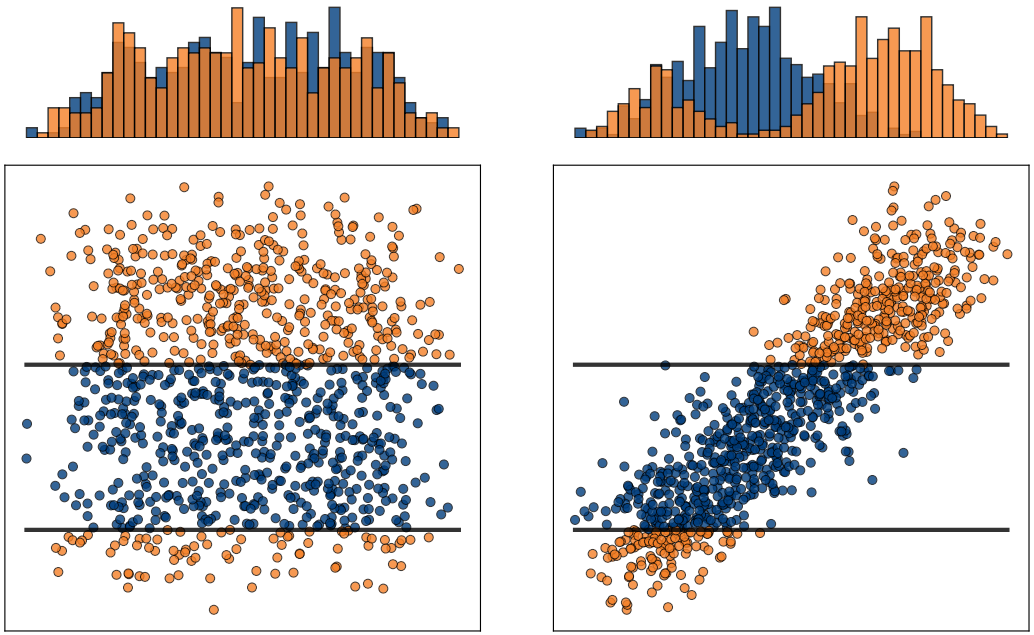
$$\mathcal{C} = \frac{1}{M}\sum_{m=1}^{M}(1 - p\text{-}value)$$

Properties:

- Independence $\Rightarrow \mathbb{E}[\mathcal{C}] = 0.5$, otherwise greater
- $\mathcal{C} \in [0, 1]$

Theorem (derived from [Hoe63]): $\Pr\left(|\mathcal{C} - \mathbb{E}[\mathcal{C}]| \geq \varepsilon\right) \leq 2e^{-2M\varepsilon^2}$

Monte Carlo Dependency Estimation



- Take a random slice
- Test $\mathcal{T}(\hat{p}_{X_i}(S), \hat{p}(S|\overline{X_i}))$
- Repeat M times
- Average:

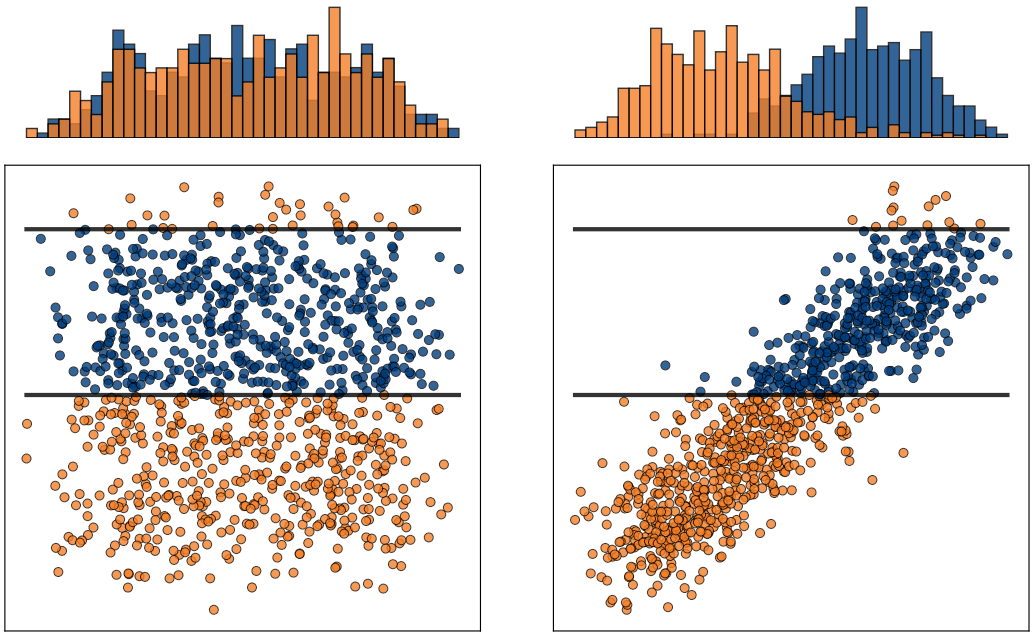
$$\mathcal{C} = \frac{1}{M}\sum_{m=1}^{M}(1 - p\text{-}value)$$

Properties:

- Independence $\Rightarrow \mathbb{E}[\mathcal{C}] = 0.5$, otherwise greater
- $\mathcal{C} \in [0, 1]$

Theorem (derived from [Hoe63]): $\Pr\left(|\mathcal{C} - \mathbb{E}[\mathcal{C}]| \geq \varepsilon\right) \leq 2e^{-2M\varepsilon^2}$

Monte Carlo Dependency Estimation



- Take a random slice
- Test $\mathcal{T}(\hat{p}_{X_i}(S), \hat{p}(S|\overline{X_i}))$
- Repeat M times
- Average:

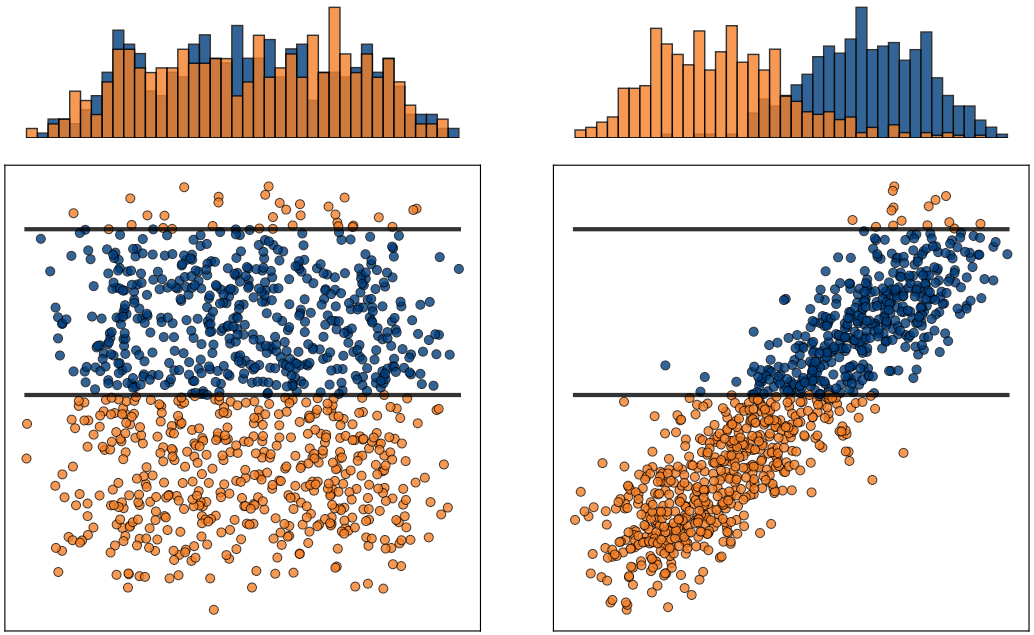
$$\mathcal{C} = \frac{1}{M}\sum_{m=1}^{M}(1 - p\text{-}value)$$

Properties:

- Independence $\Rightarrow \mathbb{E}[\mathcal{C}] = 0.5$, otherwise greater
- $\mathcal{C} \in [0, 1]$

Theorem (derived from [Hoe63]): $\Pr\left(|\mathcal{C} - \mathbb{E}[\mathcal{C}]| \geq \varepsilon\right) \leq 2e^{-2M\varepsilon^2}$

Monte Carlo Dependency Estimation



- Take a random slice
- Test $\mathcal{T}(\hat{p}_{X_i}(S), \hat{p}(S|\overline{X_i}))$
- Repeat M times
- Average:

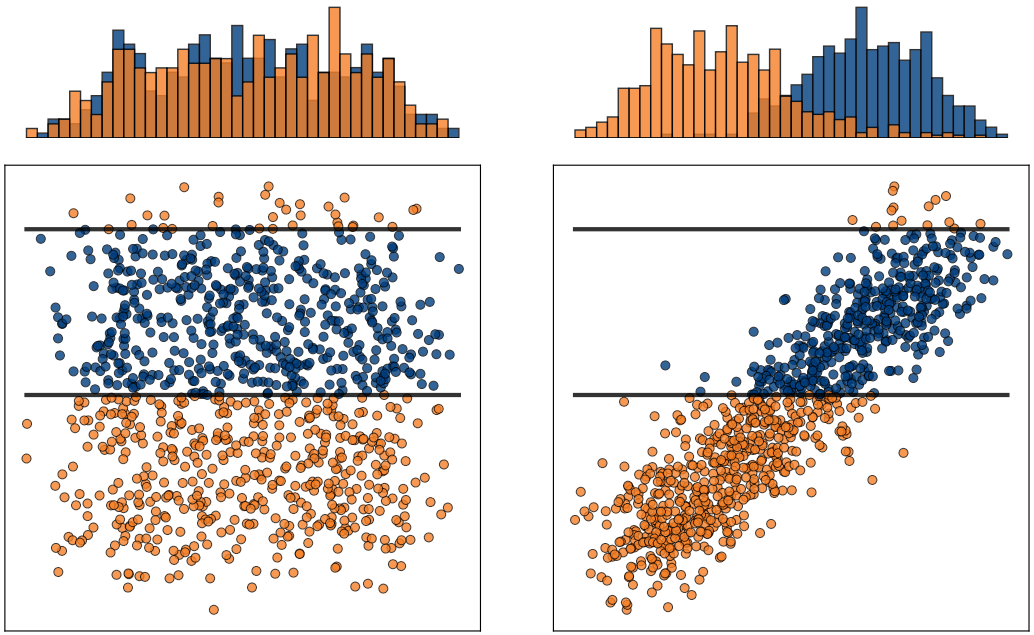
$$\mathcal{C} = \frac{1}{M}\sum_{m=1}^{M}(1 - p\text{-}value)$$

Properties:

- Independence $\Rightarrow \mathbb{E}[\mathcal{C}] = 0.5$, otherwise greater
- $\mathcal{C} \in [0, 1]$

Theorem (derived from [Hoe63]): $\Pr\left(|\mathcal{C} - \mathbb{E}[\mathcal{C}]| \geq \varepsilon\right) \leq 2e^{-2M\varepsilon^2}$

Monte Carlo Dependency Estimation



- Take a random slice
- Test $\mathcal{T}(\hat{p}_{X_i}(S), \hat{p}(S|\overline{X_i}))$
- Repeat M times
- Average:

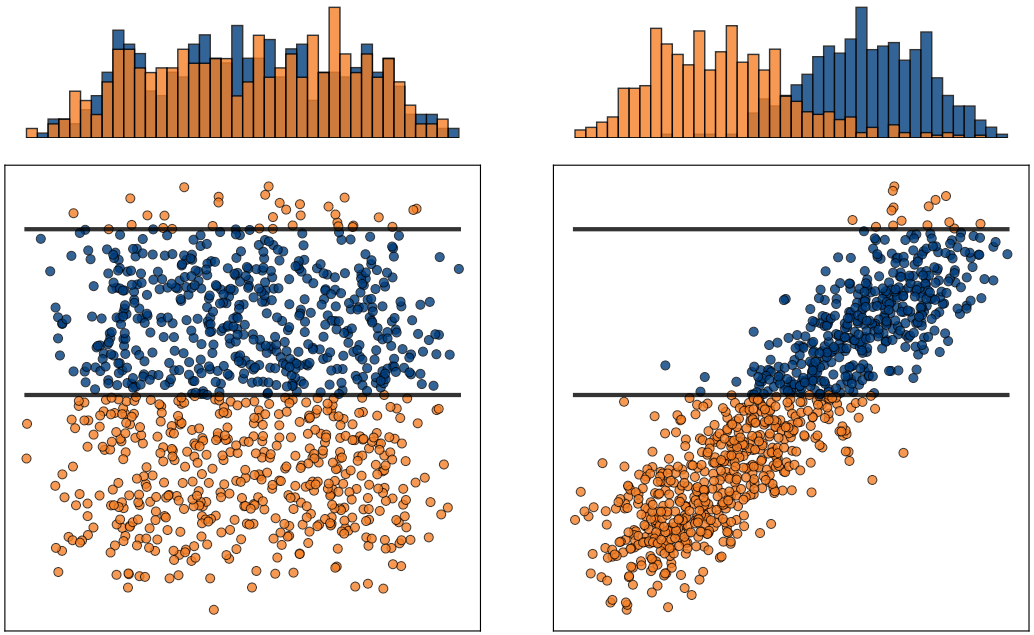
$$\mathcal{C} = \frac{1}{M}\sum_{m=1}^{M}(1 - p\text{-}value)$$

Properties:

- Independence $\Rightarrow \mathbb{E}[\mathcal{C}] = 0.5$, otherwise greater
- $\mathcal{C} \in [0, 1]$

Theorem (derived from [Hoe63]): $\Pr\left(|\mathcal{C} - \mathbb{E}[\mathcal{C}]| \geq \varepsilon\right) \leq 2e^{-2M\varepsilon^2}$

Monte Carlo Dependency Estimation



- Take a random slice
- Test $\mathcal{T}(\hat{p}_{X_i}(S), \hat{p}(S|\overline{X_i}))$
- Repeat M times
- Average:

$$\mathcal{C} = \frac{1}{M}\sum_{m=1}^{M}(1 - p\text{-}value)$$

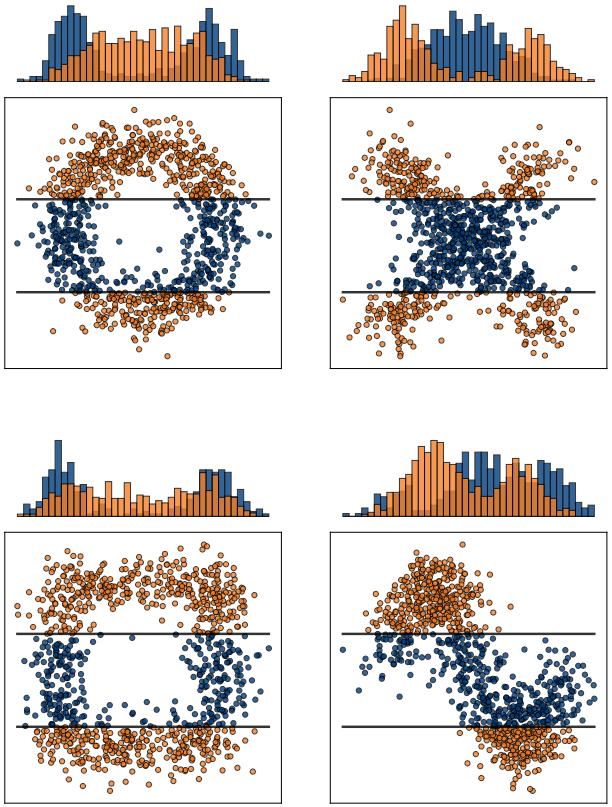
Properties:

- Independence $\Rightarrow \mathbb{E}[\mathcal{C}] = 0.5$, otherwise greater
- $\mathcal{C} \in [0, 1]$

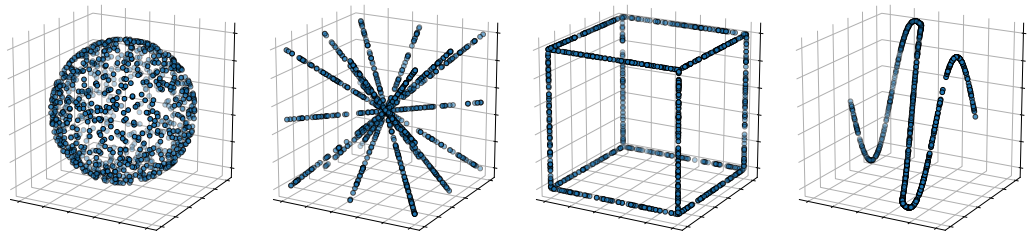
Theorem (derived from [Hoe63]): $\Pr\left(|\mathcal{C} - \mathbb{E}[\mathcal{C}]| \geq \varepsilon\right) \leq 2e^{-2M\varepsilon^2}$

MCDE: A General Estimation Framework

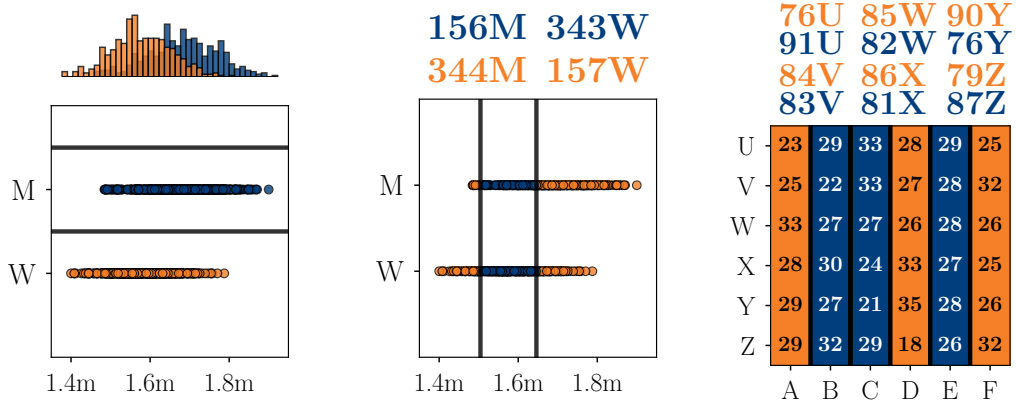
- Non-linear dependencies



- Multivariate data



- Heterogeneous data



Evaluating MCDE

We systematically assess MCDE against 12 requirements.

	Efficient	Single-Scan	Adaptation	Anytime	Heterogeneity	Multivariate	General	Intuitive	Non-parametric	Interpretable	Sensitive	Robust
MS*	++	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✓
TC**	-	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗
II***	--	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗
CMI†	+	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗
MAC‡	--	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓
UDS‡‡	-	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓
HiCS$	+	✗	✗	✓	✗	✓	✓	✗	✓	✗	✗	✗
MCDE	++	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

* [SS07], ** [Wat60], *** [McG54], † [NMV+13], ‡ [NMV+14], ‡‡ [NMV16], $ [KMB12]

- MCDE outperforms the competitors w.r.t. those requirements

Evaluating MCDE

We systematically assess MCDE against 12 requirements.

		Specific to Stream Algorithms										
		Domingos & Hulten 2003										
		Efficient	Single-Scan	Adaptation	Anytime	Heterogeneity	Multivariate	General	Intuitive	Non-parametric	Interpretable	Sensitive	Robust
MS*	++	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✓
TC**	-	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗
II***	--	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗
CMI†	+	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗
MAC‡	--	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓
UDS‡‡	-	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓
HiCS$	+	✗	✗	✓	✗	✓	✓	✗	✓	✗	✗	✗
MCDE	++	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

* [SS07], ** [Wat60], *** [McG54], † [NMV+13], ‡ [NMV+14], ‡‡ [NMV16], $ [KMB12]

- MCDE outperforms the competitors w.r.t. those requirements

Evaluating MCDE

We systematically assess MCDE against 12 requirements.

		Specific to Stream Algorithms					Specific to Dependency Estimation						
		Domingos & Hulten 2003											
		Efficient	Single-Scan	Adaptation	Anytime	Heterogeneity	Multivariate	General	Intuitive	Non-parametric	Interpretable	Sensitive	Robust
MS*	++	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✓
TC**	-	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗
II***	--	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗
CMI†	+	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗
MAC‡	--	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓
UDS‡‡	-	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓
HiCS$	+	✗	✗	✓	✗	✓	✓	✗	✓	✗	✗	✗
MCDE	++	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

* [SS07], ** [Wat60], *** [McG54], † [NMV+13], ‡ [NMV+14], ‡‡ [NMV16], $ [KMB12]

- MCDE outperforms the competitors w.r.t. those requirements

Deploying MCDE @ Bioliq

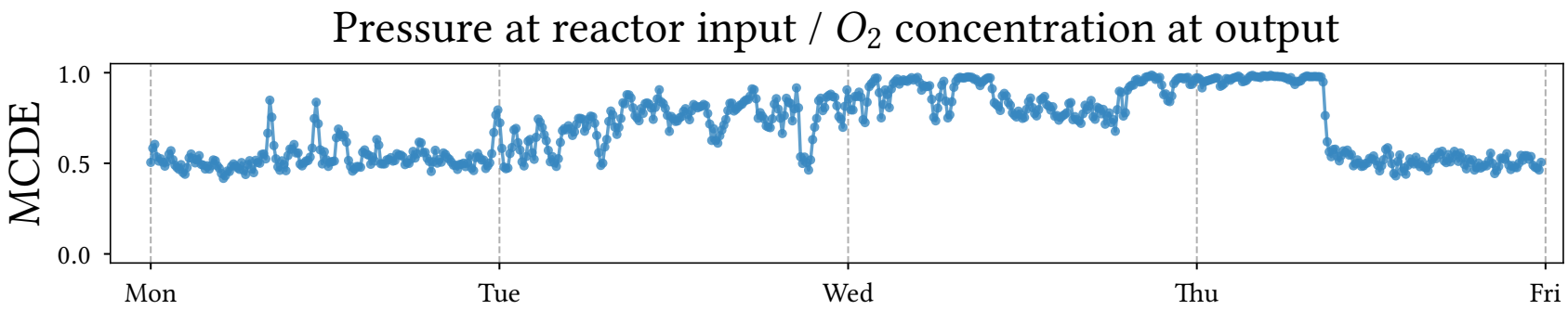
- We estimated MCDE during 4 days at Bioliq between 20 sensors
  - We presented 20 patterns to the plant operators
  - They found that 6 of them were very interesting (here is one of them)

Deploying MCDE @ Bioliq

- We estimated MCDE during 4 days at Bioliq between 20 sensors
  - We presented 20 patterns to the plant operators
  - They found that 6 of them were very interesting (here is one of them)

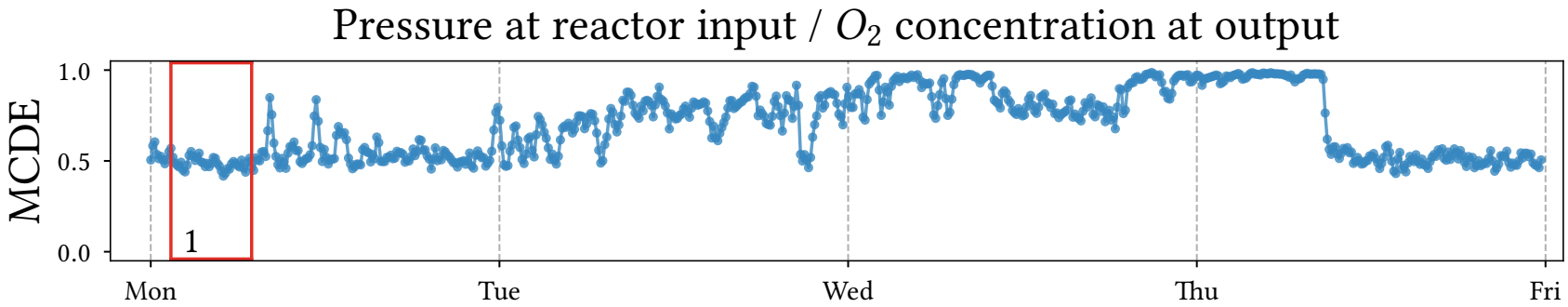
Deploying MCDE @ Bioliq

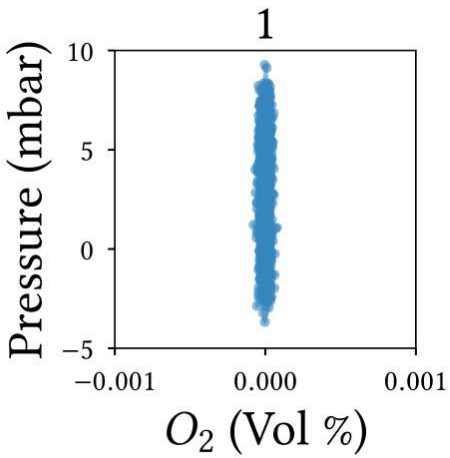
- We estimated MCDE during 4 days at Bioliq between 20 sensors
  - We presented 20 patterns to the plant operators
  - They found that 6 of them were very interesting (here is one of them)



Deploying MCDE @ Bioliq

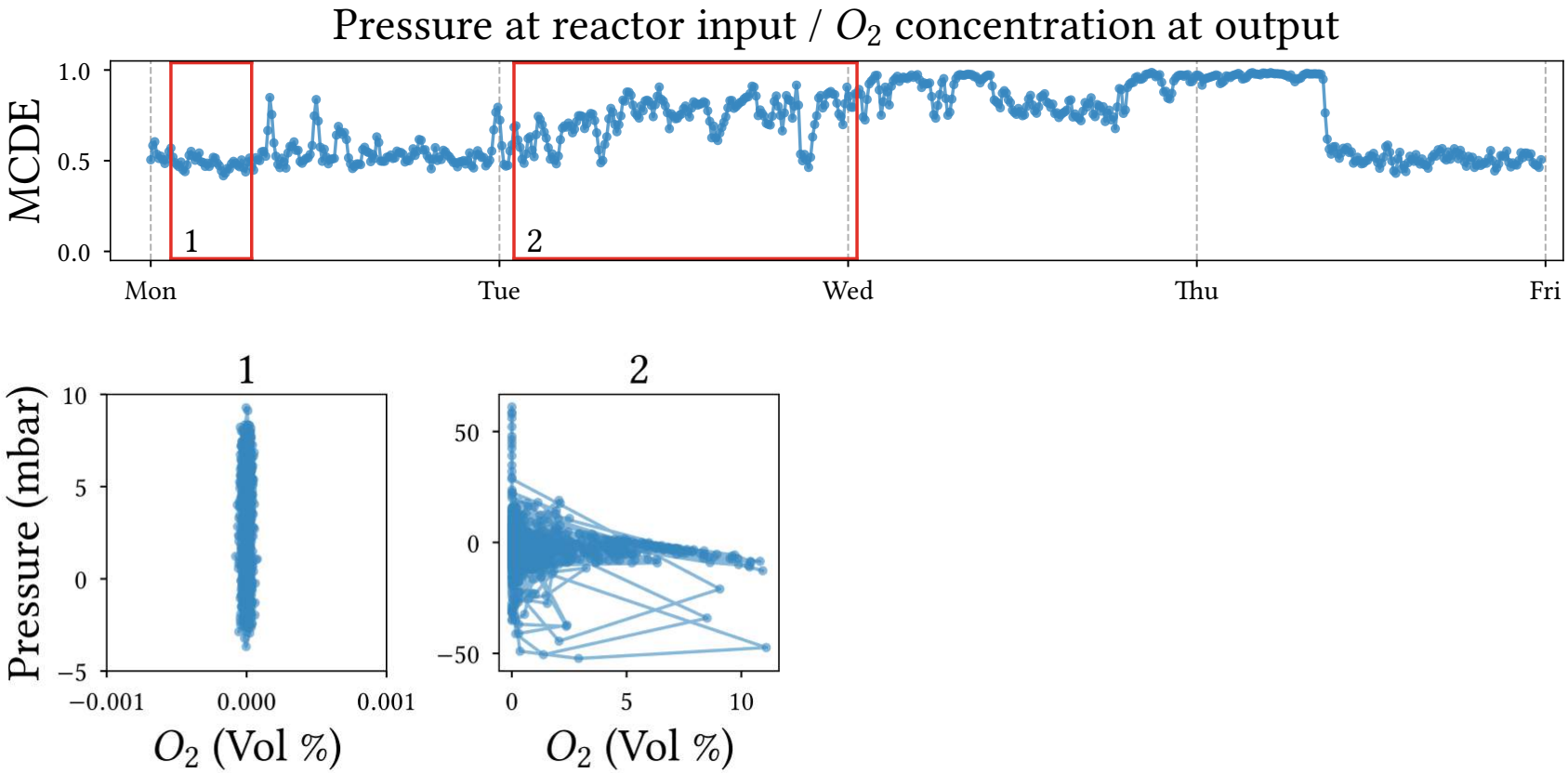
- We estimated MCDE during 4 days at Bioliq between 20 sensors
  - We presented 20 patterns to the plant operators
  - They found that 6 of them were very interesting (here is one of them)





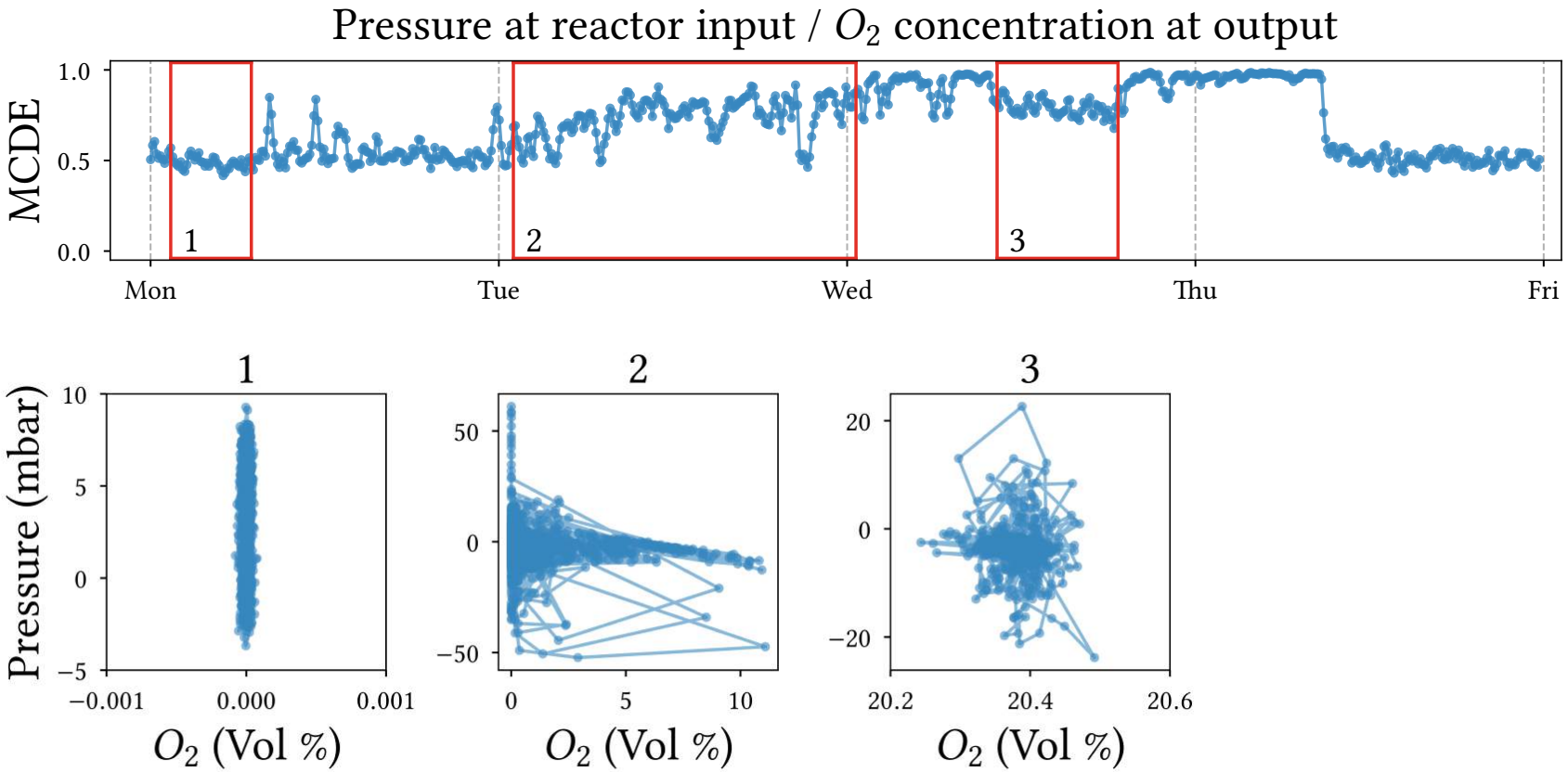
Deploying MCDE @ Bioliq

- We estimated MCDE during 4 days at Bioliq between 20 sensors
  - We presented 20 patterns to the plant operators
  - They found that 6 of them were very interesting (here is one of them)



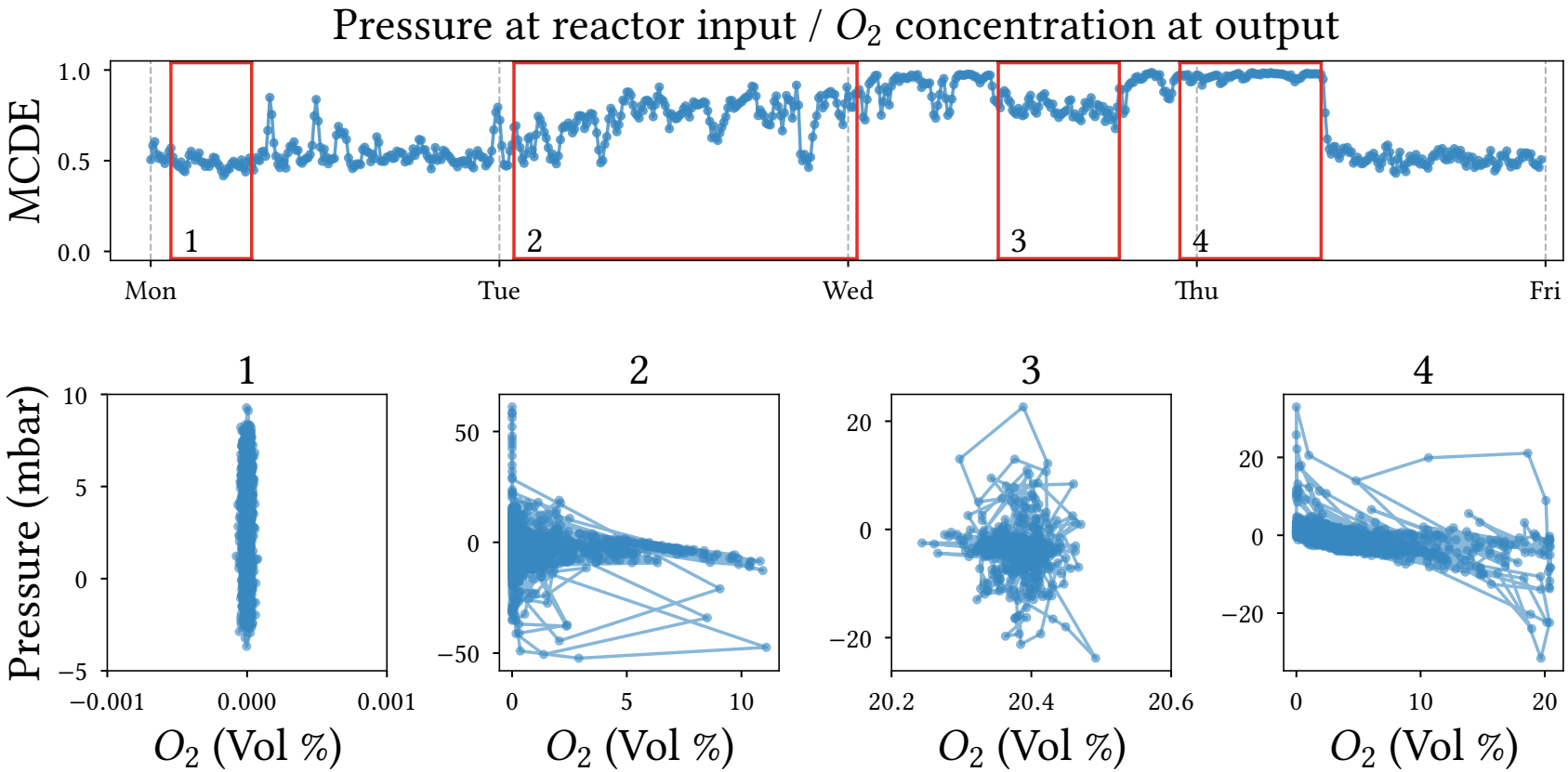
Deploying MCDE @ Bioliq

- We estimated MCDE during 4 days at Bioliq between 20 sensors
  - We presented 20 patterns to the plant operators
  - They found that 6 of them were very interesting (here is one of them)



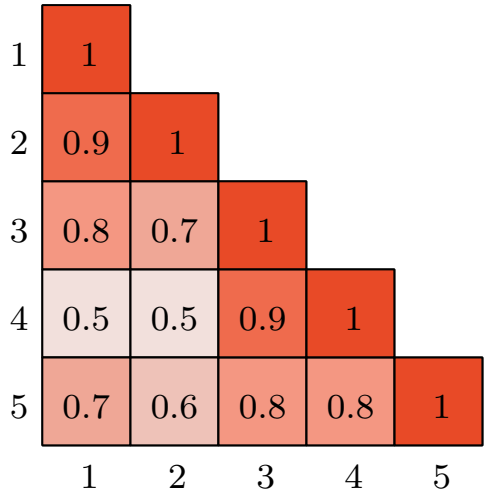
Deploying MCDE @ Bioliq

- We estimated MCDE during 4 days at Bioliq between 20 sensors
  - We presented 20 patterns to the plant operators
  - They found that 6 of them were very interesting (here is one of them)



From Estimating to Monitoring

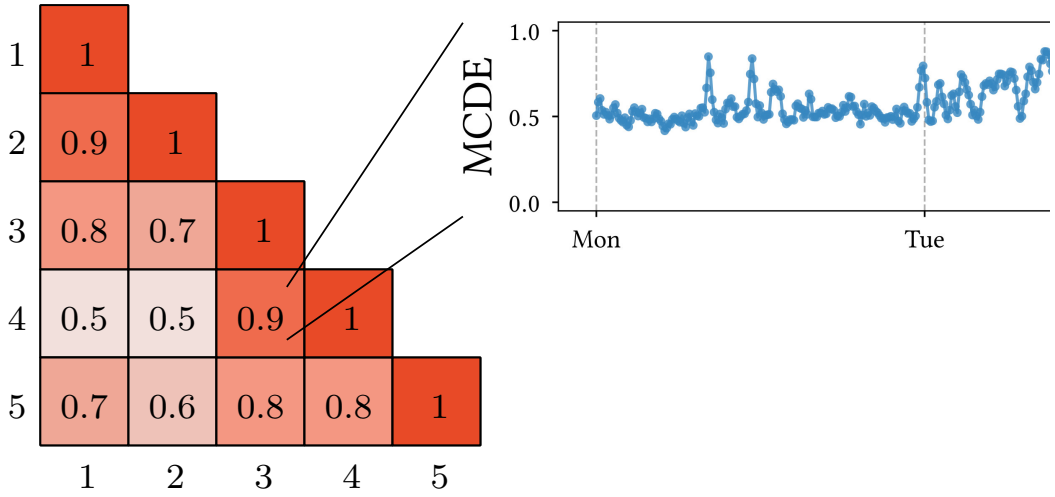
Now, we are able to estimate dependency in HD-DS.



- However, there are too many pairs/subspaces
- Only a few pairs/subspaces are actually interesting
  - Others have low correlation, or never change
- We can reduce the cost of monitoring if we find them
  - Which? How many?

From Estimating to Monitoring

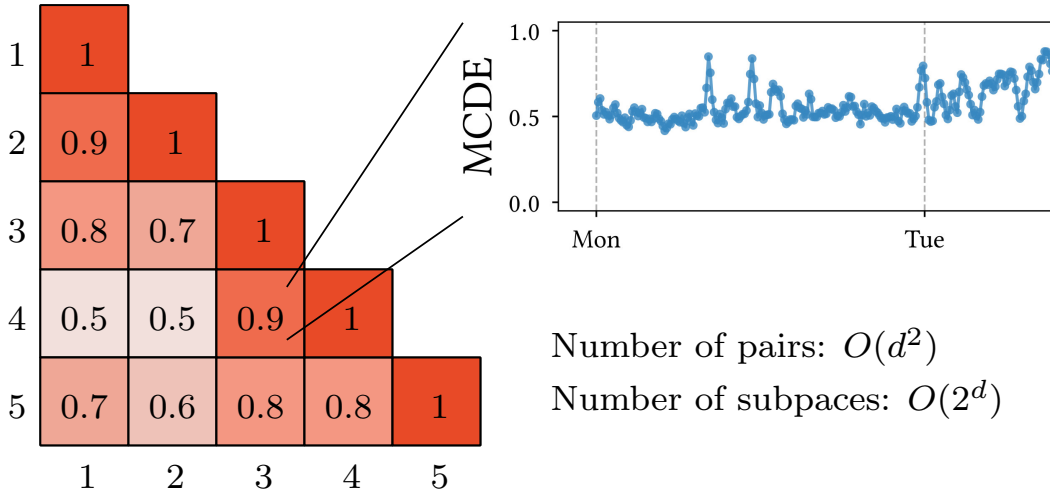
Now, we are able to estimate dependency in HD-DS.



- However, there are too many pairs/subspaces
- Only a few pairs/subspaces are actually interesting
  - Others have low correlation, or never change
- We can reduce the cost of monitoring if we find them
  - Which? How many?

From Estimating to Monitoring

Now, we are able to estimate dependency in HD-DS.



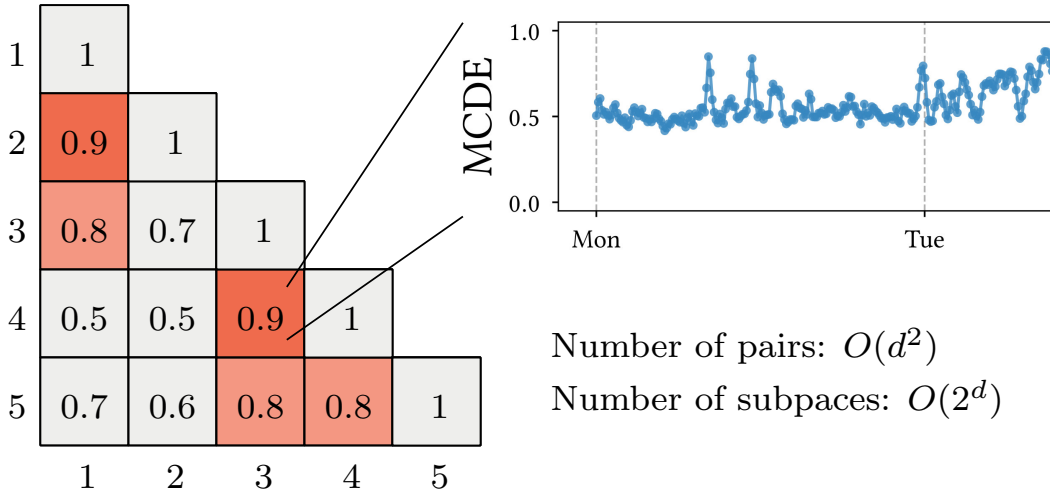
Number of pairs: $O(d^2)$

Number of subpaces: $O(2^d)$

- However, there are too many pairs/subspaces
- Only a few pairs/subspaces are actually interesting
  - Others have low correlation, or never change
- We can reduce the cost of monitoring if we find them
  - Which? How many?

From Estimating to Monitoring

Now, we are able to estimate dependency in HD-DS.



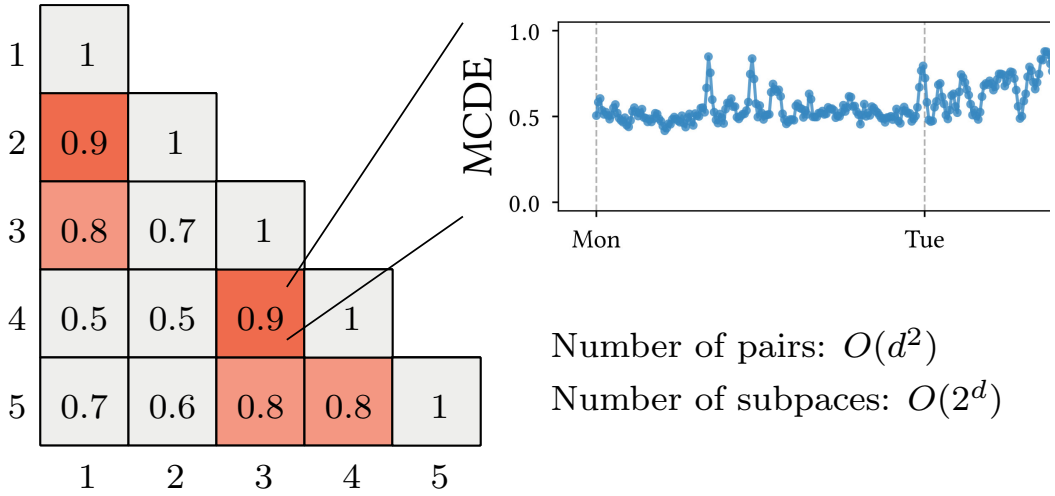
Number of pairs: $O(d^2)$

Number of subpaces: $O(2^d)$

- However, there are too many pairs/subspaces
- Only a few pairs/subspaces are actually interesting
  - Others have low correlation, or never change
- We can reduce the cost of monitoring if we find them
  - Which? How many?

From Estimating to Monitoring

Now, we are able to estimate dependency in HD-DS.

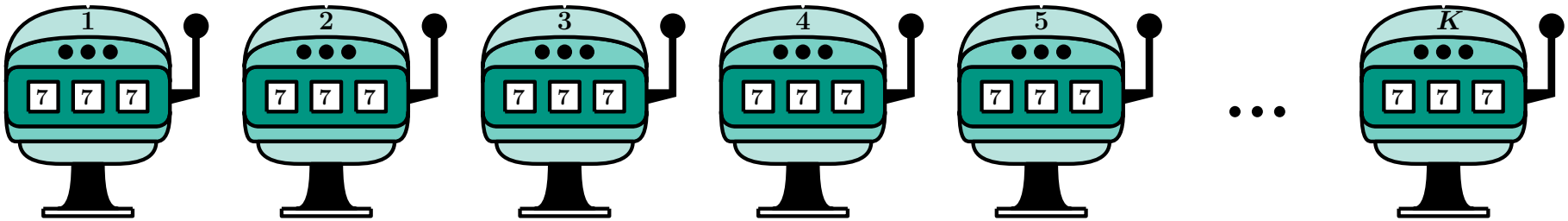


Number of pairs: $O(d^2)$

Number of subpaces: $O(2^d)$

- However, there are too many pairs/subspaces
- Only a few pairs/subspaces are actually interesting
  - Others have low correlation, or never change
- We can reduce the cost of monitoring if we find them
  - Which? How many?

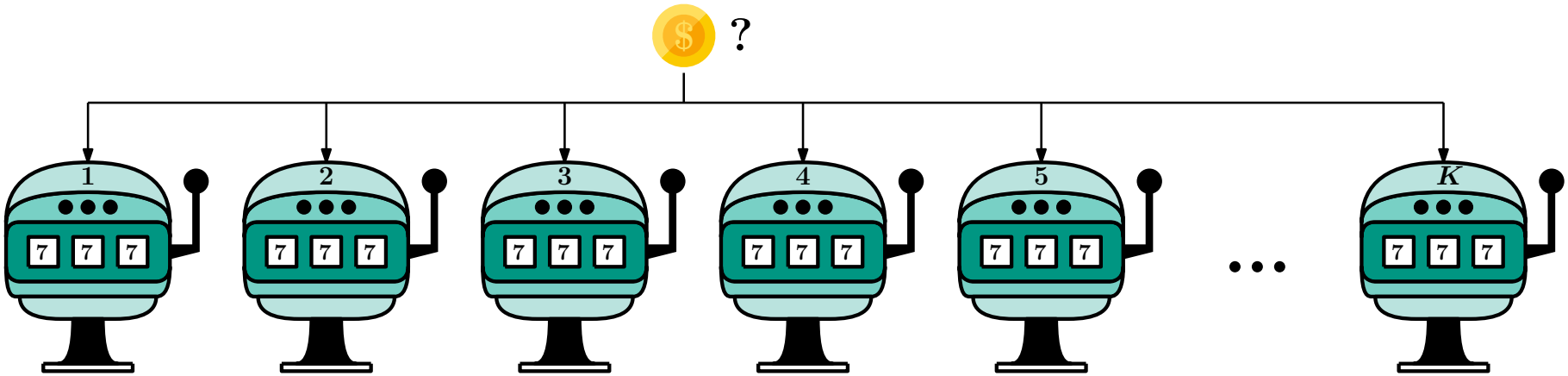
Monitoring as Multi-Armed Bandit (MAB)



To maximise our gain, we need to find:

- Which arms are the best?
- How many arms to play?
- When to adapt?

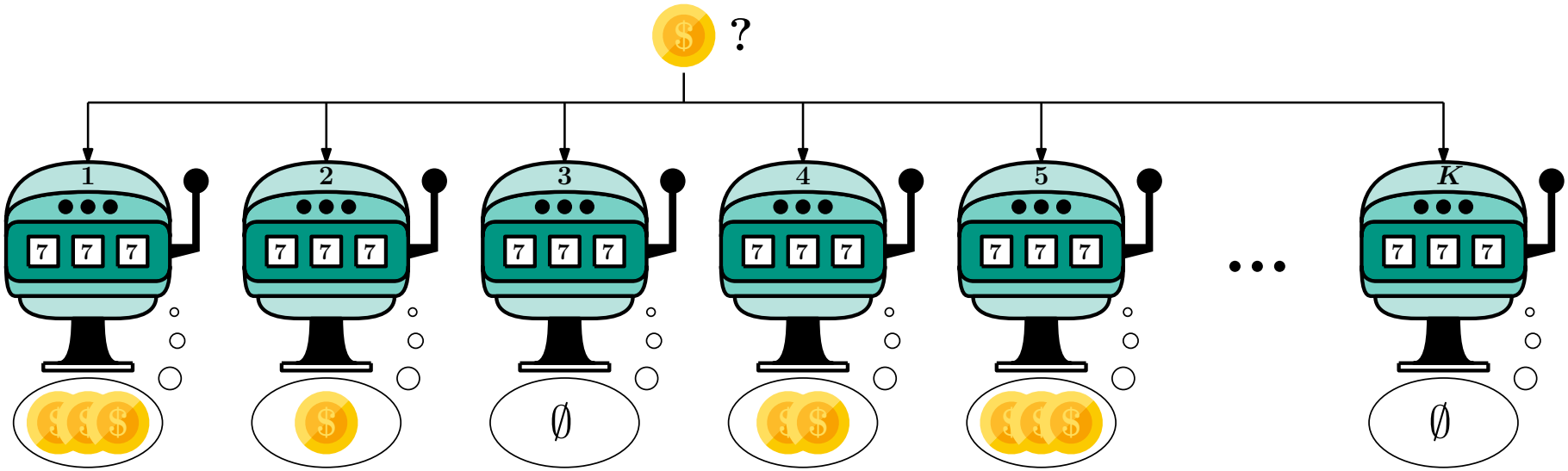
Monitoring as Multi-Armed Bandit (MAB)



To maximise our gain, we need to find:

- Which arms are the best?
- How many arms to play?
- When to adapt?

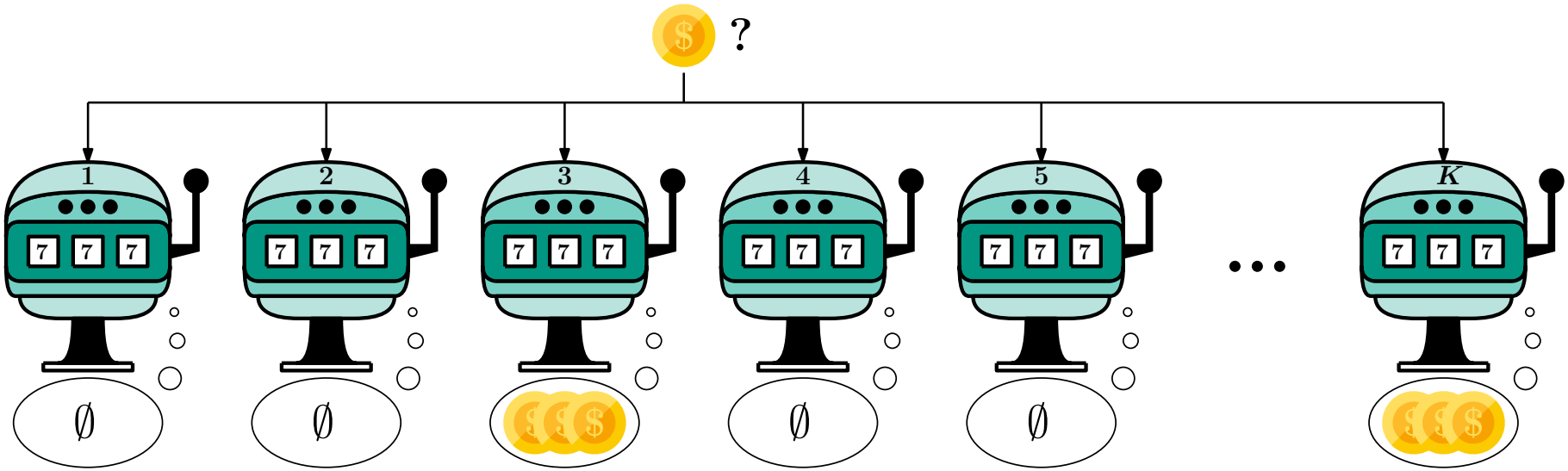
Monitoring as Multi-Armed Bandit (MAB)



To maximise our gain, we need to find:

- Which arms are the best?
- How many arms to play?
- When to adapt?

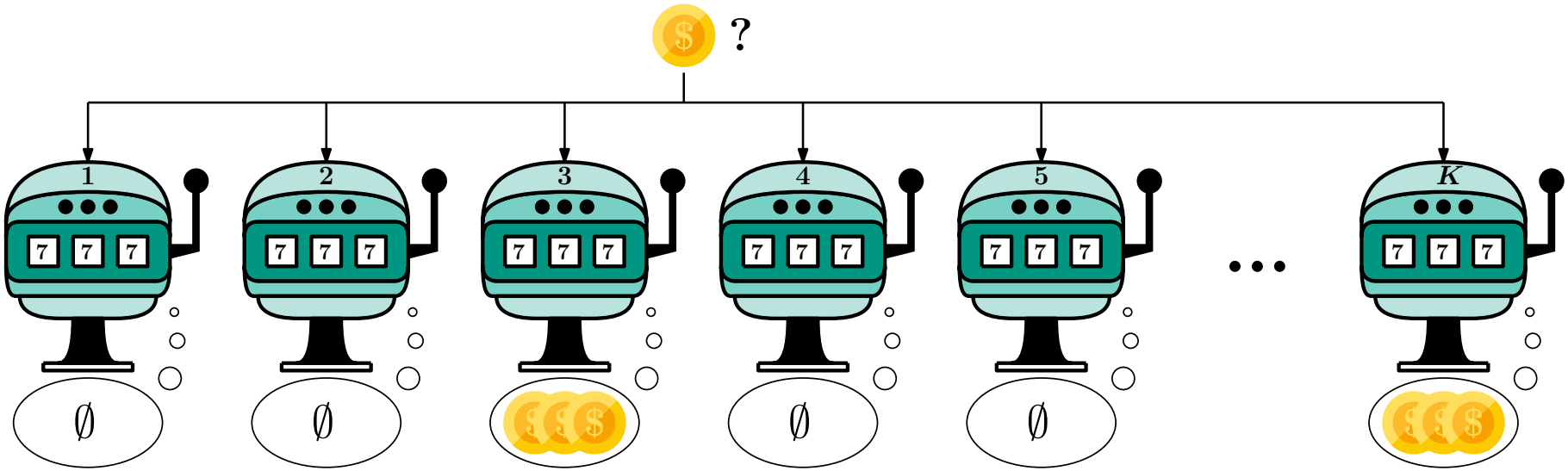
Monitoring as Multi-Armed Bandit (MAB)



To maximise our gain, we need to find:

- Which arms are the best?
- How many arms to play?
- When to adapt?

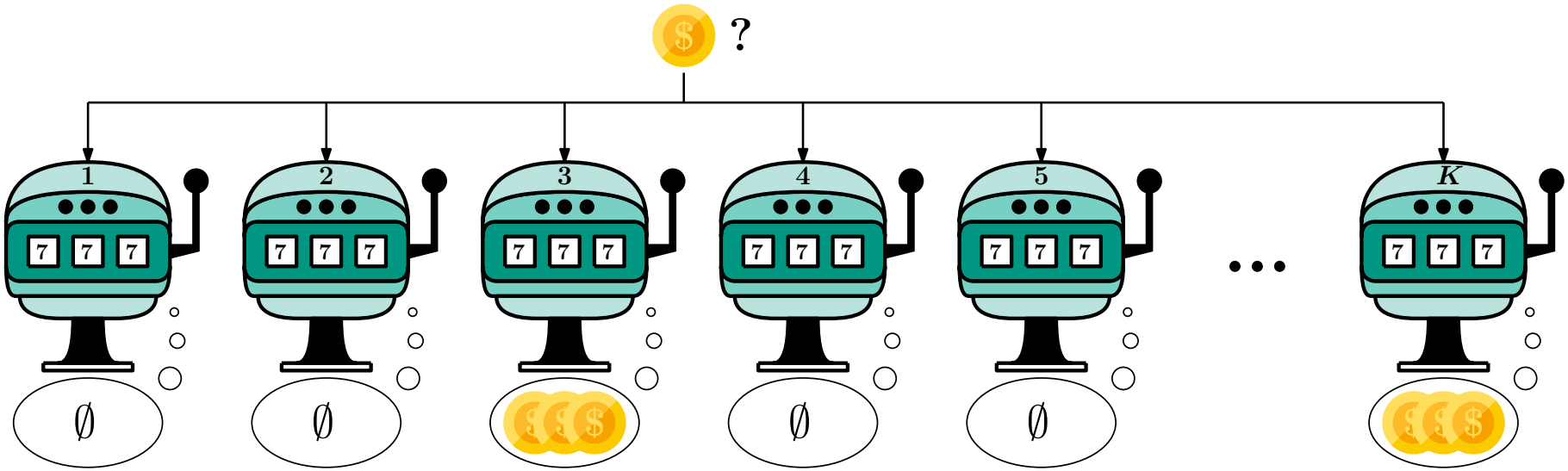
Monitoring as Multi-Armed Bandit (MAB)



To maximise our gain, we need to find:

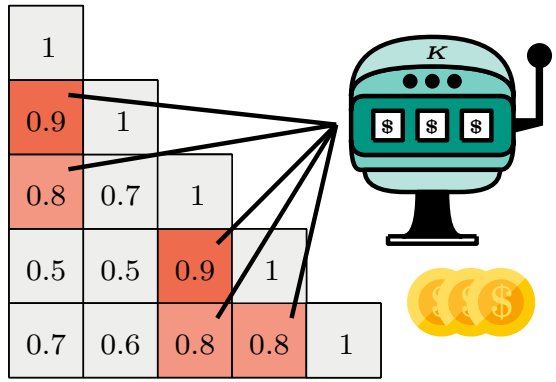
- Which arms are the best?
- How many arms to play?
- When to adapt?

Monitoring as Multi-Armed Bandit (MAB)



To maximise our gain, we need to find:

- Which arms are the best?
- How many arms to play?
- When to adapt?



Bandit Models

Problem: Existing models do not capture our setting

The "Classical" MAB:

- Only one play per round. [Tho33, KKM12]
- The environment is static (no change).

MAB with Multiple Plays (MP-MAB):

- $L > 1$ plays per round [UNK10, KHN15].
- Assumes a static setting; L is a constant.

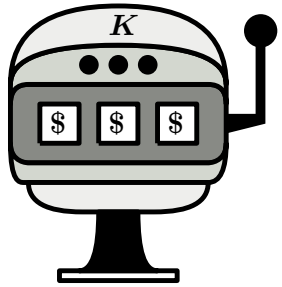
Non-static MAB:

- Integrate a forgetting mechanism [ACFS02, GM11].
- Usually: sliding window, exponential weighting.
- Single-play; Difficult parameter setting.

$\rightarrow$ **We invent the Scaling Multi-Armed Bandit (S-MAB):**

- A MP-MAB with a dynamic number of plays.
- Supports the non-static setting.

Bandit Models



Problem: Existing models do not capture our setting

The "Classical" MAB:

- Only one play per round. [Tho33, KKM12]
- The environment is static (no change).

MAB with Multiple Plays (MP-MAB):

- $L > 1$ plays per round [UNK10, KHN15].
- Assumes a static setting; L is a constant.

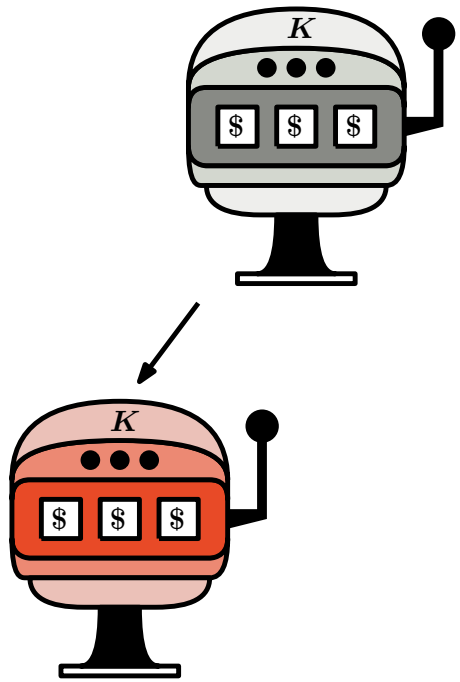
Non-static MAB:

- Integrate a forgetting mechanism [ACFS02, GM11].
- Usually: sliding window, exponential weighting.
- Single-play; Difficult parameter setting.

→ **We invent the Scaling Multi-Armed Bandit (S-MAB):**

- A MP-MAB with a dynamic number of plays.
- Supports the non-static setting.

Bandit Models



Problem: Existing models do not capture our setting

The "Classical" MAB:

- Only one play per round. [Tho33, KKM12]
- The environment is static (no change).

MAB with Multiple Plays (MP-MAB):

- $L > 1$ plays per round [UNK10, KHN15].
- Assumes a static setting; L is a constant.

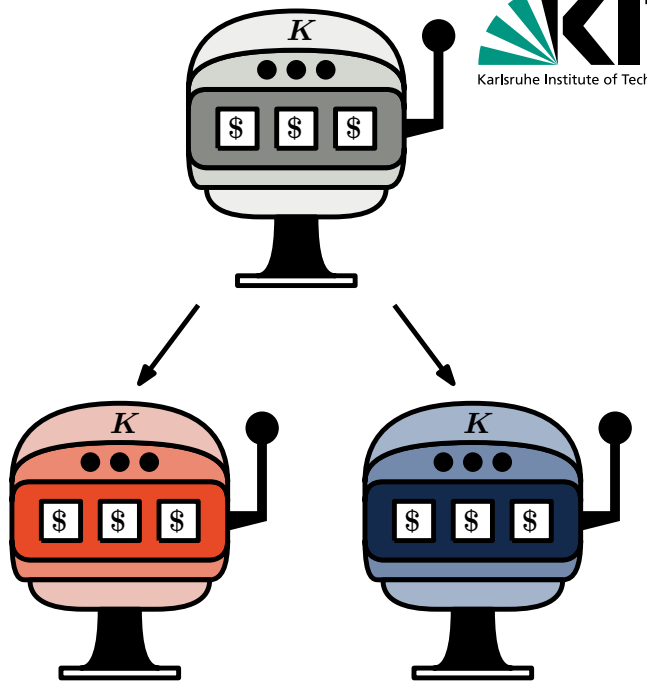
Non-static MAB:

- Integrate a forgetting mechanism [ACFS02, GM11].
- Usually: sliding window, exponential weighting.
- Single-play; Difficult parameter setting.

→ **We invent the Scaling Multi-Armed Bandit (S-MAB):**

- A MP-MAB with a dynamic number of plays.
- Supports the non-static setting.

Bandit Models



Problem: Existing models do not capture our setting

The "Classical" MAB:

- Only one play per round. [Tho33, KKM12]
- The environment is static (no change).

MAB with Multiple Plays (MP-MAB):

- $L > 1$ plays per round [UNK10, KHN15].
- Assumes a static setting; L is a constant.

Non-static MAB:

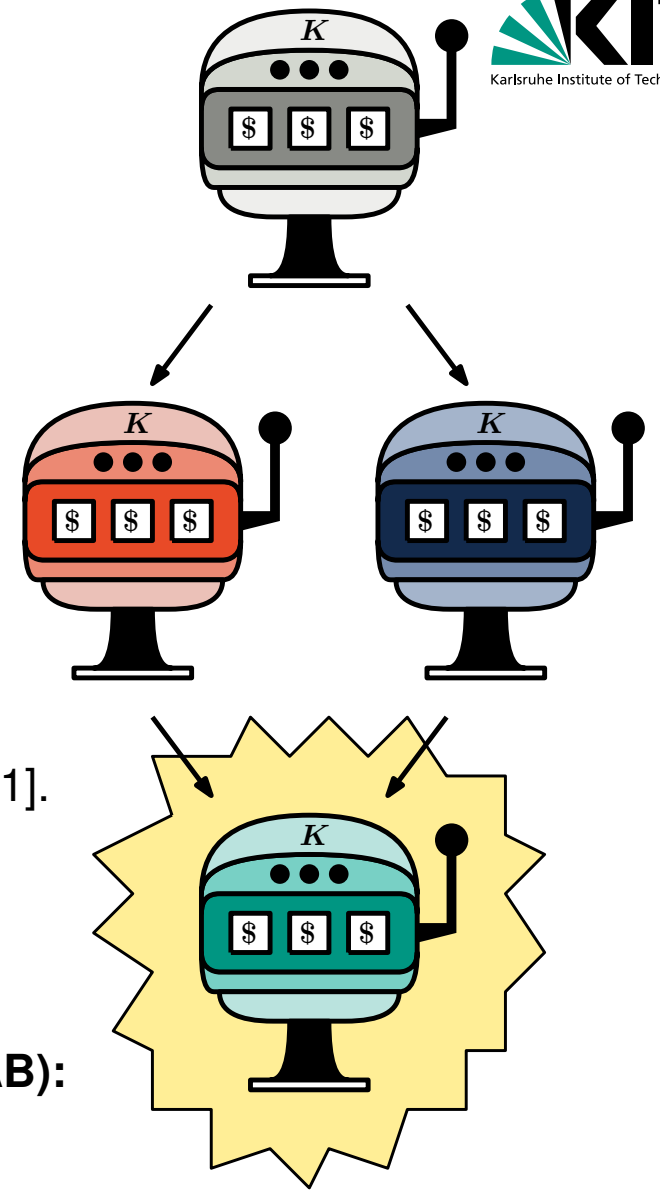
- Integrate a forgetting mechanism [ACFS02, GM11].
- Usually: sliding window, exponential weighting.
- Single-play; Difficult parameter setting.

→ **We invent the Scaling Multi-Armed Bandit (S-MAB):**

- A MP-MAB with a dynamic number of plays.
- Supports the non-static setting.

Bandit Models

Problem: Existing models do not capture our setting



The "Classical" MAB:

- Only one play per round. [Tho33, KKM12]
- The environment is static (no change).

MAB with Multiple Plays (MP-MAB):

- $L > 1$ plays per round [UNK10, KHN15].
- Assumes a static setting; L is a constant.

Non-static MAB:

- Integrate a forgetting mechanism [ACFS02, GM11].
- Usually: sliding window, exponential weighting.
- Single-play; Difficult parameter setting.

→ **We invent the Scaling Multi-Armed Bandit (S-MAB):**

- A MP-MAB with a dynamic number of plays.
- Supports the non-static setting.

The Scaling Multi-Armed Bandit (S-MAB)

Idea: S-MAB solves the following constrained optimisation problem:

$$\underbrace{\max_{I_t \subset [K]}}_{\text{Set of arms}} \overbrace{\sum_{i \in I_t} S_i(t)}^{\text{Sum of rewards}} \quad s.t. \quad \underbrace{\eta_t = \frac{\sum_{i \in I_t} \mu_i}{|I_t|}}_{\text{Average expected reward}} > \overbrace{\eta^*}^{\text{Efficiency}} \tag{1}$$

If the player always chooses the best arms, then the problem is equivalent to finding an optimal number of plays L^*:

$$L^* = \max_{1 \leq L \leq K} L \quad s.t. \quad \underbrace{\frac{\sum_{i=1}^{L} \mu_i}{L}}_{\substack{\text{Average expected reward} \\ \text{from the top-L arms}}} > \eta^* \tag{2}$$

The Scaling Multi-Armed Bandit (S-MAB)

Idea: S-MAB solves the following constrained optimisation problem:

$$\underbrace{\max_{I_t \subset [K]}}_{\text{Set of arms}} \overbrace{\sum_{i \in I_t} S_i(t)}^{\text{Sum of rewards}} \quad s.t. \quad \underbrace{\eta_t = \frac{\sum_{i \in I_t} \mu_i}{|I_t|}}_{\text{Average expected reward}} > \overbrace{\eta^*}^{\text{Efficiency}} \tag{1}$$

If the player always chooses the best arms, then the problem is equivalent to finding an optimal number of plays L^*:

$$L^* = \max_{1 \leq L \leq K} L \quad s.t. \quad \underbrace{\frac{\sum_{i=1}^{L} \mu_i}{L}}_{\substack{\text{Average expected reward} \\ \text{from the top-L arms}}} > \eta^* \tag{2}$$

General Scaling Multi-Armed Bandit

Two components for success: $\overbrace{\text{finding the top-arms}}^{\text{Existing bandits}} + \overbrace{\text{finding } L^*}^{\text{New !}}$

At each round $t = 1, \dots, T$:

- 1. The player chooses I_t with $|I_t| = L_t$, and observes a reward vector X_t
- 2. They update their estimation $\hat{\mu}_i$ for $i \in I_t$
- 3. **They choose L_{t+1} ($\rightarrow$ Scaling)**

There exists many approaches for steps $1, 2$:

- Multiple-Play Thompson Sampling (MP-TS) [Tho33, KKM12, KHN15]
- UCB-type bandits [ACF02, CHL+16, GC11]

For step 3 $\rightarrow$ We introduce a "scaling policy" (see next slide)

General Scaling Multi-Armed Bandit

Two components for success: $\overbrace{\text{finding the top-arms}}^{\text{Existing bandits}} + \overbrace{\text{finding } L^*}^{\text{New !}}$

At each round $t = 1, \ldots, T$:

- 1. The player chooses I_t with $|I_t| = L_t$, and observes a reward vector X_t
- 2. They update their estimation $\hat{\mu}_i$ for $i \in I_t$
- 3. **They choose L_{t+1} ($\rightarrow$ Scaling)**

There exists many approaches for steps 1, 2:

- Multiple-Play Thompson Sampling (MP-TS) [Tho33, KKM12, KHN15]
- UCB-type bandits [ACF02, CHL+16, GC11]

For step 3 $\rightarrow$ We introduce a "scaling policy" (see next slide)

General Scaling Multi-Armed Bandit

Two components for success: $\overbrace{\text{finding the top-arms}}^{\text{Existing bandits}} + \overbrace{\text{finding } L^*}^{\text{New !}}$

At each round $t = 1, \dots, T$:

- 1. The player chooses I_t with $|I_t| = L_t$, and observes a reward vector X_t
- 2. They update their estimation $\hat{\mu}_i$ for $i \in I_t$
- 3. **They choose L_{t+1} ($\rightarrow$ Scaling)**

There exists many approaches for steps $1, 2$:

- Multiple-Play Thompson Sampling (MP-TS) [Tho33, KKM12, KHN15]
- UCB-type bandits [ACF02, CHL+16, GC11]

For step 3 $\rightarrow$ We introduce a "scaling policy" (see next slide)

General Scaling Multi-Armed Bandit

Two components for success: $\overbrace{\text{finding the top-arms}}^{\text{Existing bandits}} + \overbrace{\text{finding } L^*}^{\text{New !}}$

At each round $t = 1, \dots, T$:

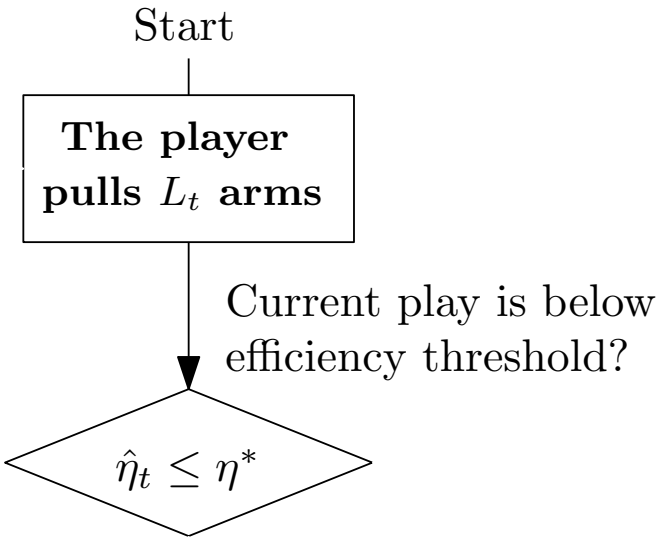
- 1. The player chooses I_t with $|I_t| = L_t$, and observes a reward vector X_t
- 2. They update their estimation $\hat{\mu}_i$ for $i \in I_t$
- 3. **They choose L_{t+1} ($\rightarrow$ Scaling)**

There exists many approaches for steps $1, 2$:

- Multiple-Play Thompson Sampling (MP-TS) [Tho33, KKM12, KHN15]
- UCB-type bandits [ACF02, CHL+16, GC11]

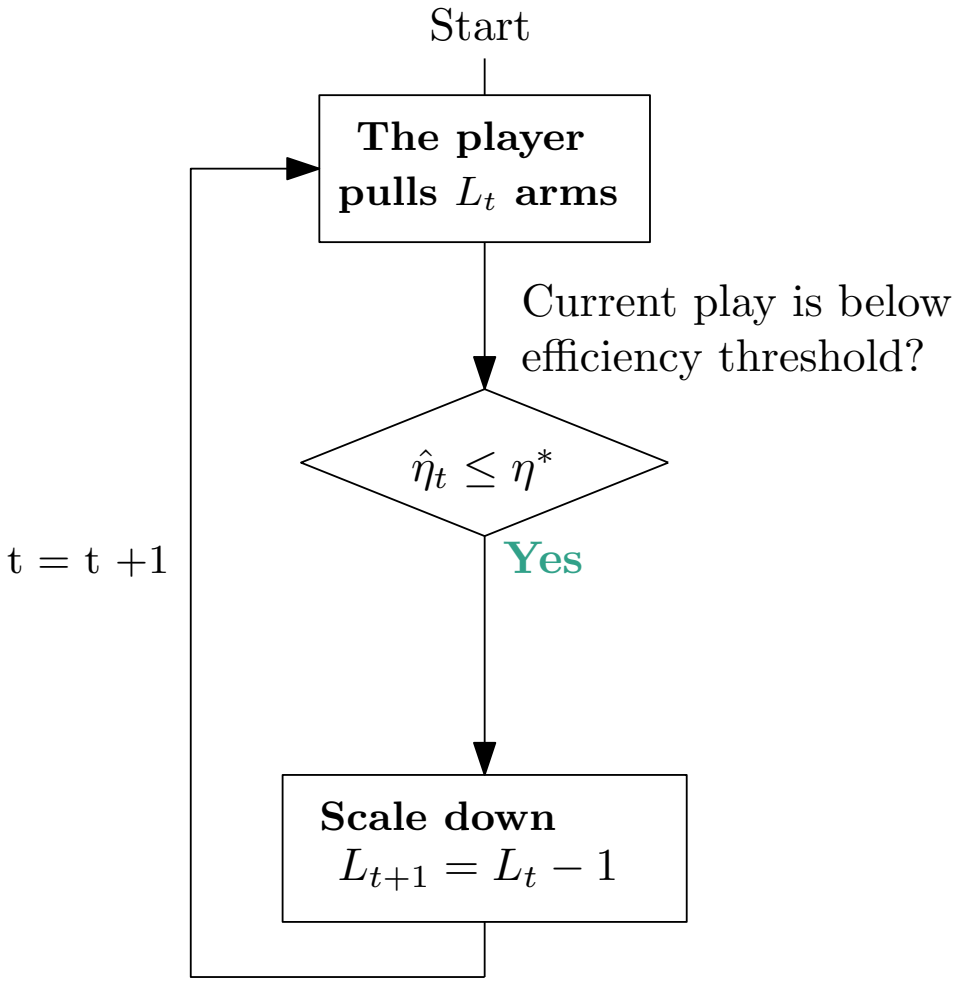
For step 3 $\rightarrow$ We introduce a "scaling policy" (see next slide)

Scaling Policy: Kullback-Leibler Scaling



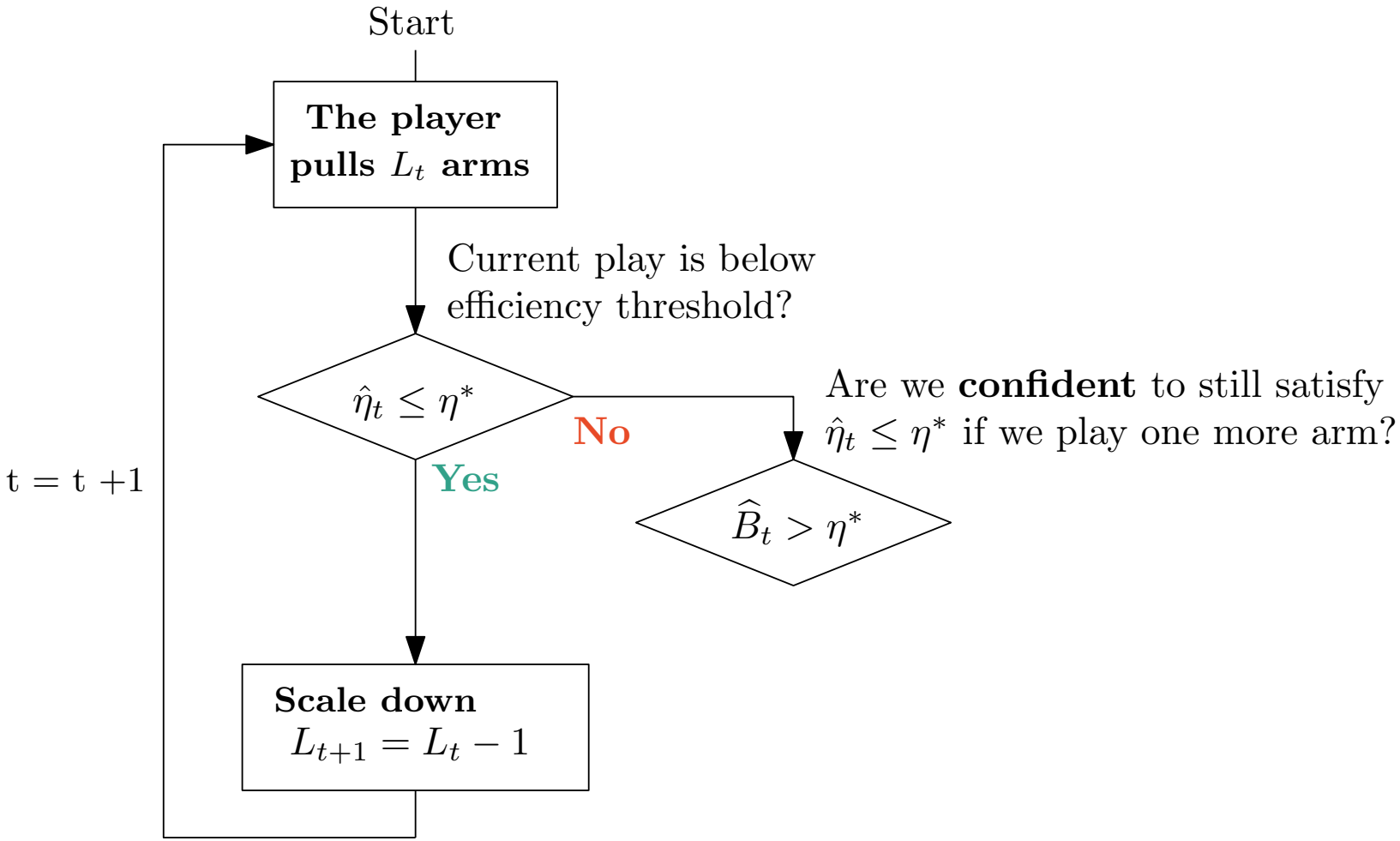
$\widehat{B}_t$ is an upper bound based on Kullback-Leibler divergence [GC11]

Scaling Policy: Kullback-Leibler Scaling



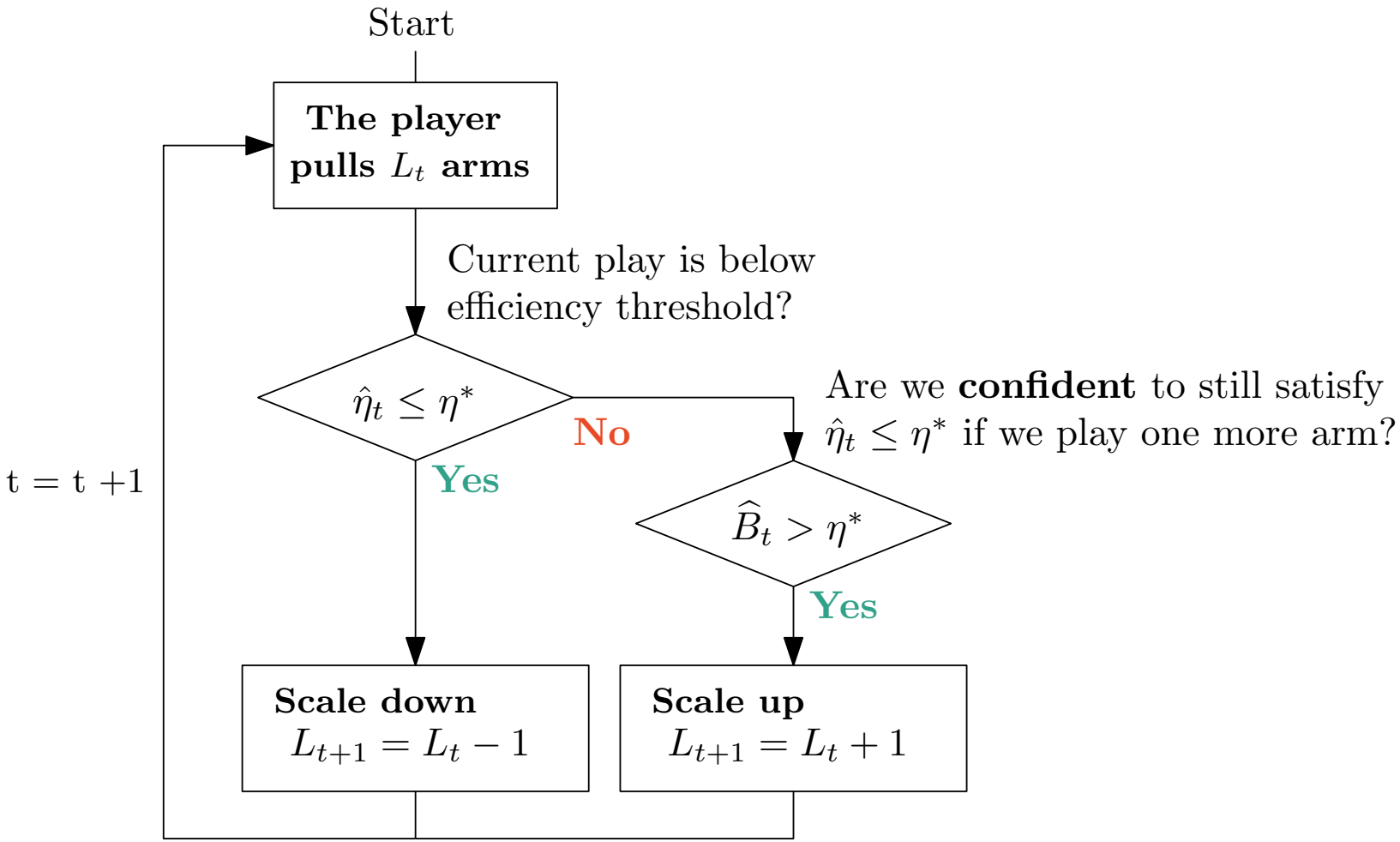
$\widehat{B}_t$ is an upper bound based on Kullback-Leibler divergence [GC11]

Scaling Policy: Kullback-Leibler Scaling



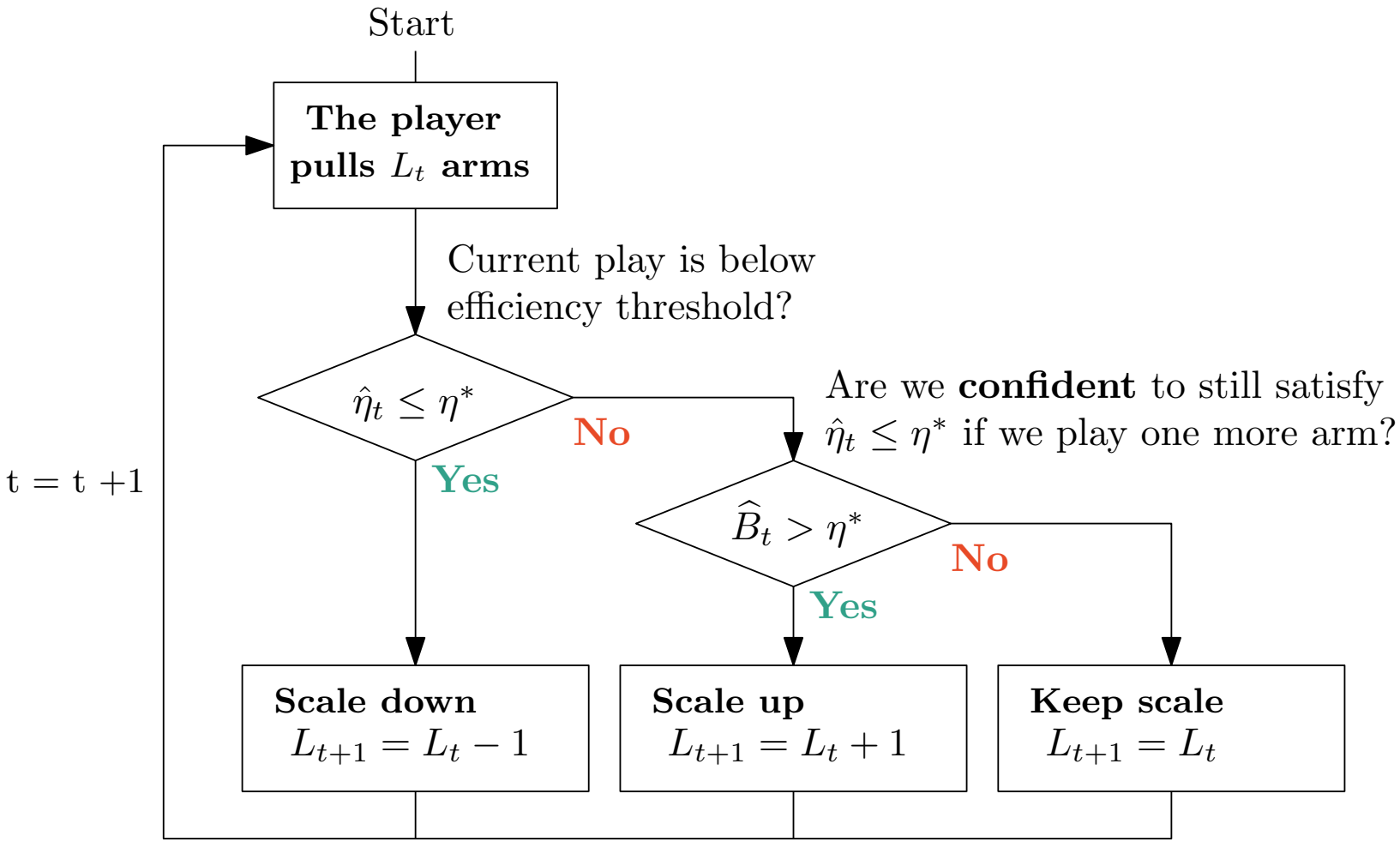
$\widehat{B}_t$ is an upper bound based on Kullback-Leibler divergence [GC11]

Scaling Policy: Kullback-Leibler Scaling



$\widehat{B}_t$ is an upper bound based on Kullback-Leibler divergence [GC11]

Scaling Policy: Kullback-Leibler Scaling



$\widehat{B}_t$ is an upper bound based on Kullback-Leibler divergence [GC11]

Evaluating the S-MAB

We want to minimise two notions of regret:

$$\underbrace{\mathrm{Reg}(T) = \sum_{t=1}^{T} \left[\overbrace{\sum_{i \in I_t^*} \mu_i}^{\text{Best arms}} - \overbrace{\sum_{i \in I_t} \mu_i}^{\text{Played arms}} \right]}_{\text{“standard” regret}} \qquad \underbrace{\mathrm{PReg}(T) = \sum_{t=1}^{T} |L^* - L_t|}_{\text{“pull” regret}}$$

$\mathrm{Reg}(T)$ is small $\rightarrow$ Top-L arms identification (step 1, 2)
$\mathrm{PReg}(T)$ is small $\rightarrow$ Scaling converges to L^* (step 3)
$\rightarrow$ We prove that $\mathrm{Reg}(T)$ and $\mathrm{PReg}(T)$ are in $O(\log T)$ with our scaling policy

For the non-static setting:

- We combine S-MAB with Adaptative Window (ADWIN) [BG07]
- Leads to excellent empirical performance

Evaluating the S-MAB

We want to minimise two notions of regret:

$$\underbrace{\mathrm{Reg}(T) = \sum_{t=1}^{T} \left[\overbrace{\sum_{i \in I_t^*} \mu_i}^{\text{Best arms}} - \overbrace{\sum_{i \in I_t} \mu_i}^{\text{Played arms}} \right]}_{\text{“standard” regret}} \qquad \underbrace{\mathrm{PReg}(T) = \sum_{t=1}^{T} |L^* - L_t|}_{\text{“pull” regret}}$$

$\mathrm{Reg}(T)$ is small $\rightarrow$ Top-L arms identification (step 1, 2)
$\mathrm{PReg}(T)$ is small $\rightarrow$ Scaling converges to L^* (step 3)
$\rightarrow$ We prove that $\mathrm{Reg}(T)$ and $\mathrm{PReg}(T)$ are in $O(\log T)$ with our scaling policy

For the non-static setting:

- We combine S-MAB with Adaptative Window (ADWIN) [BG07]
- Leads to excellent empirical performance

Evaluating the S-MAB

We want to minimise two notions of regret:

$$\underbrace{\mathrm{Reg}(T) = \sum_{t=1}^{T} \left[\overbrace{\sum_{i \in I_t^*} \mu_i}^{\text{Best arms}} - \overbrace{\sum_{i \in I_t} \mu_i}^{\text{Played arms}} \right]}_{\text{“standard” regret}} \qquad \underbrace{\mathrm{PReg}(T) = \sum_{t=1}^{T} |L^* - L_t|}_{\text{“pull” regret}}$$

$\mathrm{Reg}(T)$ is small $\rightarrow$ Top-L arms identification (step 1, 2)
$\mathrm{PReg}(T)$ is small $\rightarrow$ Scaling converges to L^* (step 3)
$\rightarrow$ We prove that $\mathrm{Reg}(T)$ and $\mathrm{PReg}(T)$ are in $O(\log T)$ with our scaling policy

For the non-static setting:

- We combine S-MAB with Adaptative Window (ADWIN) [BG07]
- Leads to excellent empirical performance

Evaluating the S-MAB

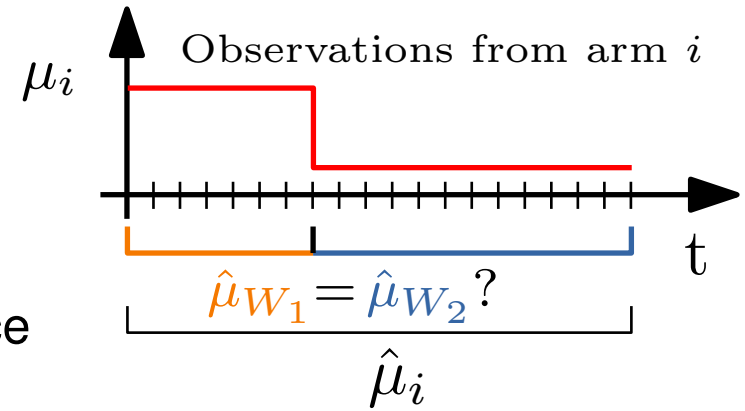
We want to minimise two notions of regret:

$$\underbrace{\mathrm{Reg}(T) = \sum_{t=1}^{T} \left[\overbrace{\sum_{i \in I_t^*} \mu_i}^{\text{Best arms}} - \overbrace{\sum_{i \in I_t} \mu_i}^{\text{Played arms}} \right]}_{\text{“standard” regret}} \qquad \underbrace{\mathrm{PReg}(T) = \sum_{t=1}^{T} |L^* - L_t|}_{\text{“pull” regret}}$$

$\mathrm{Reg}(T)$ is small $\rightarrow$ Top-L arms identification (step 1, 2)
$\mathrm{PReg}(T)$ is small $\rightarrow$ Scaling converges to L^* (step 3)
$\rightarrow$ We prove that $\mathrm{Reg}(T)$ and $\mathrm{PReg}(T)$ are in $O(\log T)$ with our scaling policy

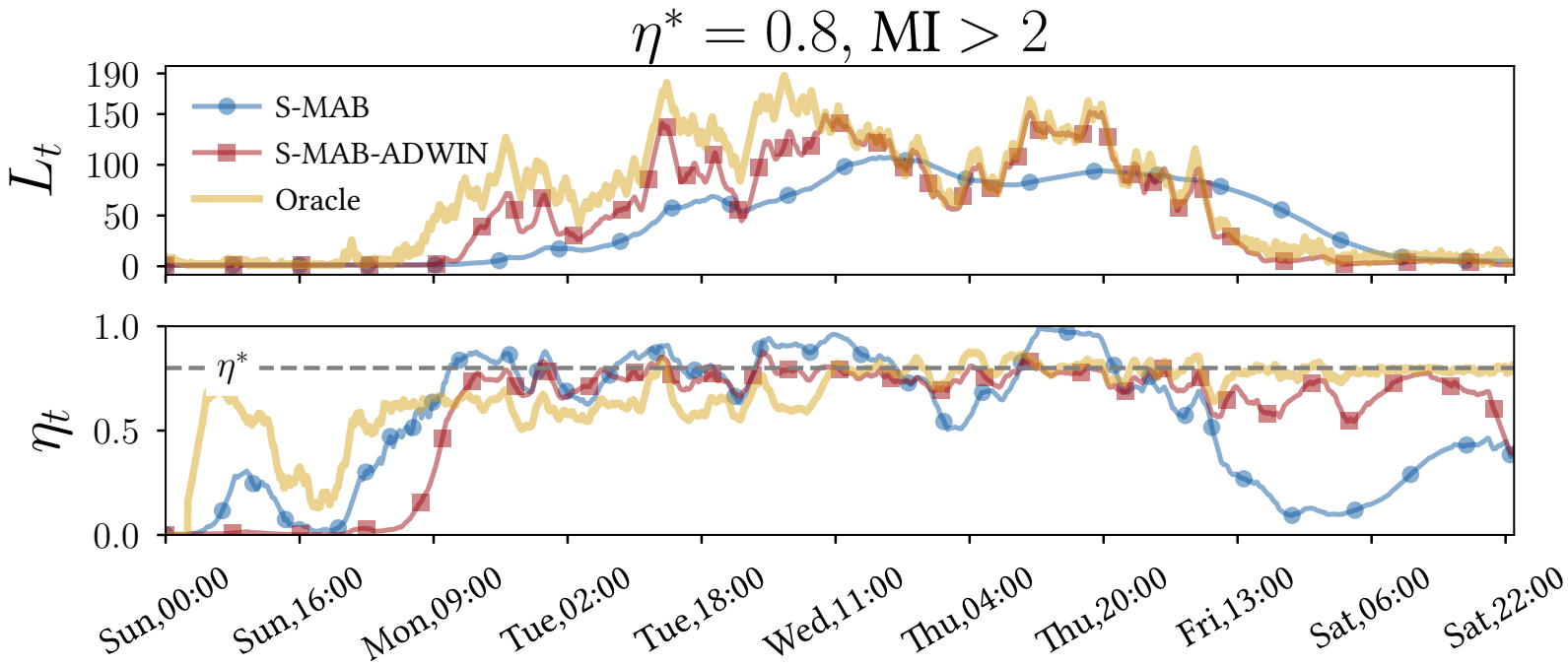
For the non-static setting:

- We combine S-MAB with Adaptative Window (ADWIN) [BG07]
- Leads to excellent empirical performance



Deploying S-MAB @ Bioliq

- Bioliq: Monitoring MI between 20 sensors, 1 week of data



- S-MAB-ADWIN can adapt to non-static streams !

Monitoring $\rightarrow$ Knowledge Discovery

Subspace Search in Data Streams

- Find at any time, a set of “high-quality” subspaces
  - High-quality subspaces reveal patterns: clusters, outliers, . . .
  - Correlation/Dependency is a good proxy for such “quality”
- They are difficult to find, because we consider HD-DS:
  - There are so many subspaces
  - Interesting subspaces may change over time

Monitoring → Knowledge Discovery

Subspace Search in Data Streams

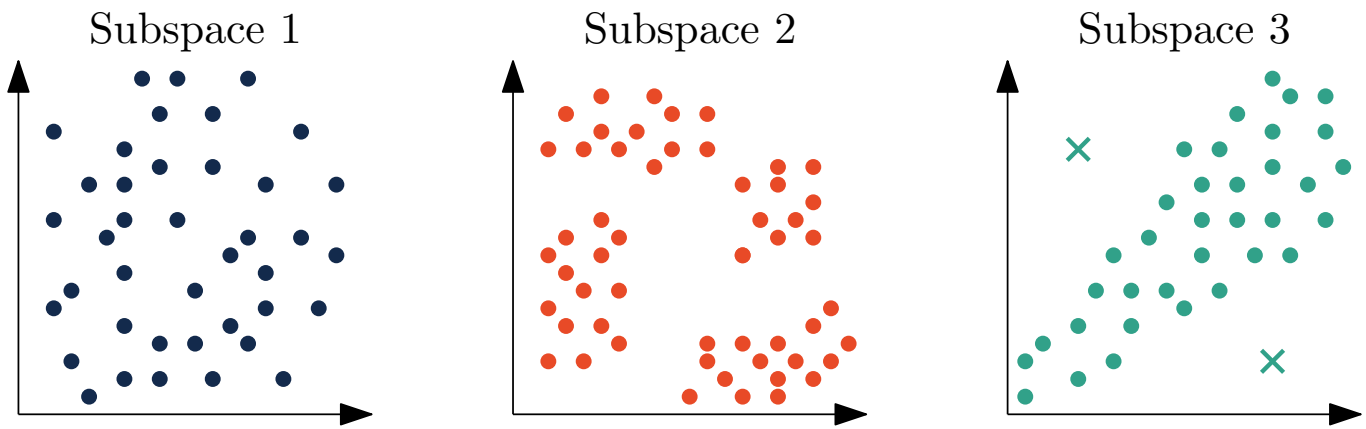
- Find at any time, a set of “high-quality” subspaces
  - High-quality subspaces reveal patterns: clusters, outliers, . . .
  - Correlation/Dependency is a good proxy for such “quality”

- They are difficult to find, because we consider HD-DS:
  - There are so many subspaces
  - Interesting subspaces may change over time

Monitoring → Knowledge Discovery

Subspace Search in Data Streams

- Find at any time, a set of “high-quality” subspaces
  - High-quality subspaces reveal patterns: clusters, outliers, . . .
  - Correlation/Dependency is a good proxy for such “quality”

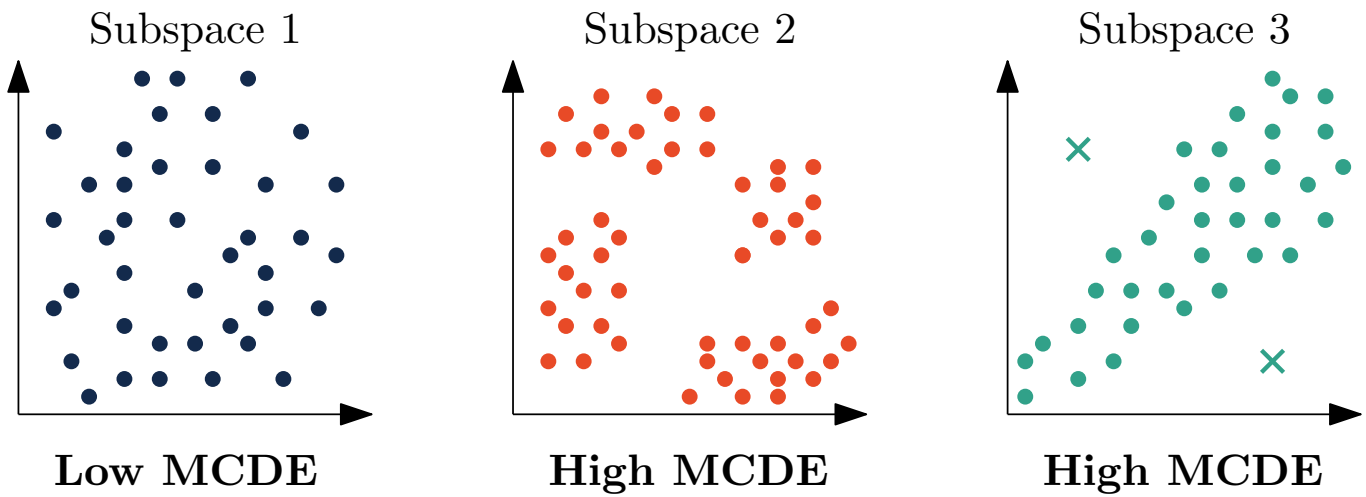


- They are difficult to find, because we consider HD-DS:
  - There are so many subspaces
  - Interesting subspaces may change over time

Monitoring → Knowledge Discovery

Subspace Search in Data Streams

- Find at any time, a set of "high-quality" subspaces
  - High-quality subspaces reveal patterns: clusters, outliers, ...
  - Correlation/Dependency is a good proxy for such "quality"

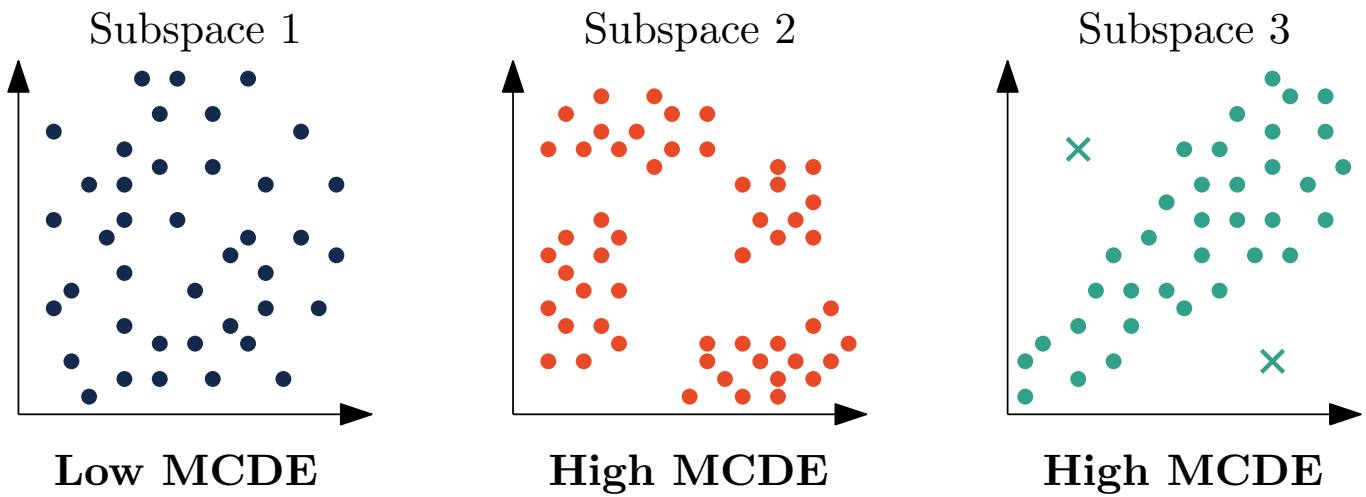


- They are difficult to find, because we consider HD-DS:
  - There are so many subspaces
  - Interesting subspaces may change over time

Monitoring → Knowledge Discovery

Subspace Search in Data Streams

- Find at any time, a set of "high-quality" subspaces
  - High-quality subspaces reveal patterns: clusters, outliers, ...
  - Correlation/Dependency is a good proxy for such "quality"



- They are difficult to find, because we consider HD-DS:
  - There are so many subspaces
  - Interesting subspaces may change over time

Subspace Search in Data Streams

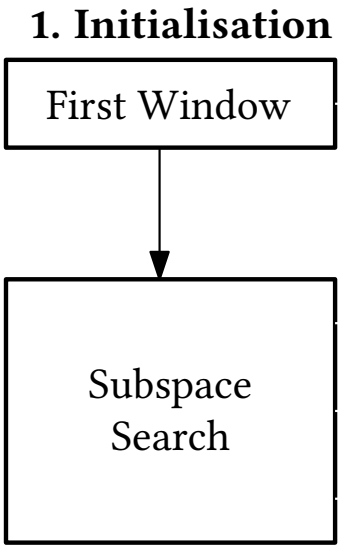
SGMRD: Streaming Greedy Maximum Random Deviation

- Extend previous static work (GMD [TB19]) to streams.
- A two-phase process:

Subspace Search in Data Streams

SGMRD: Streaming Greedy Maximum Random Deviation

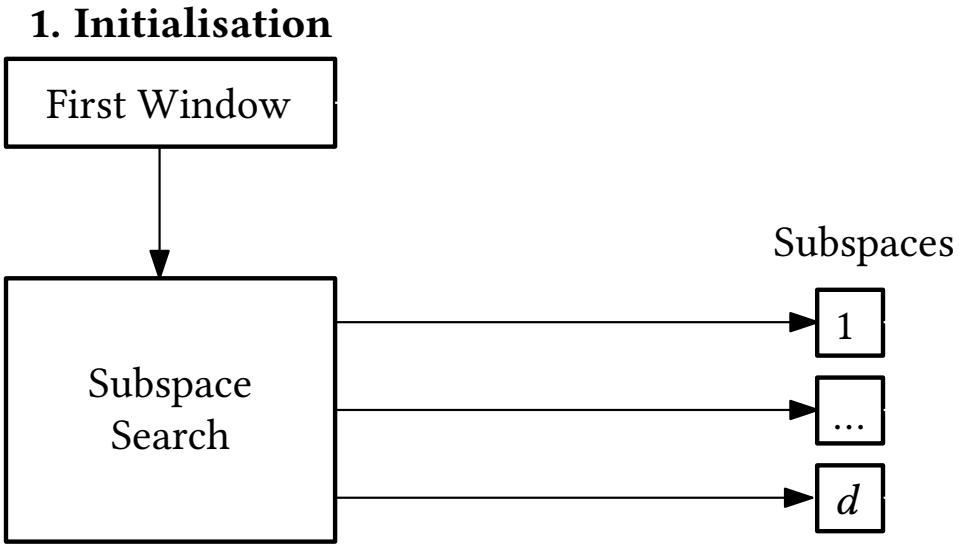
- Extend previous static work (GMD [TB19]) to streams.
- A two-phase process:



Subspace Search in Data Streams

SGMRD: Streaming Greedy Maximum Random Deviation

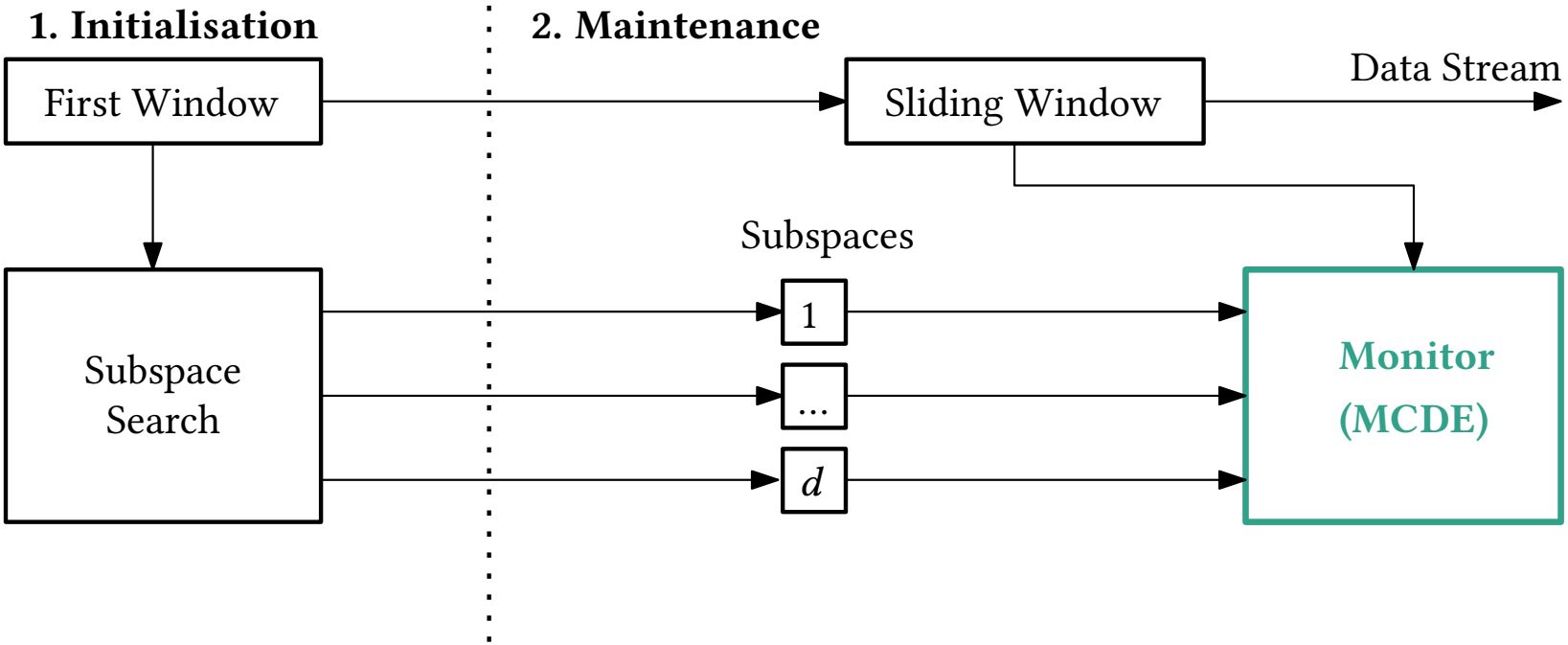
- Extend previous static work (GMD [TB19]) to streams.
- A two-phase process:



Subspace Search in Data Streams

SGMRD: Streaming Greedy Maximum Random Deviation

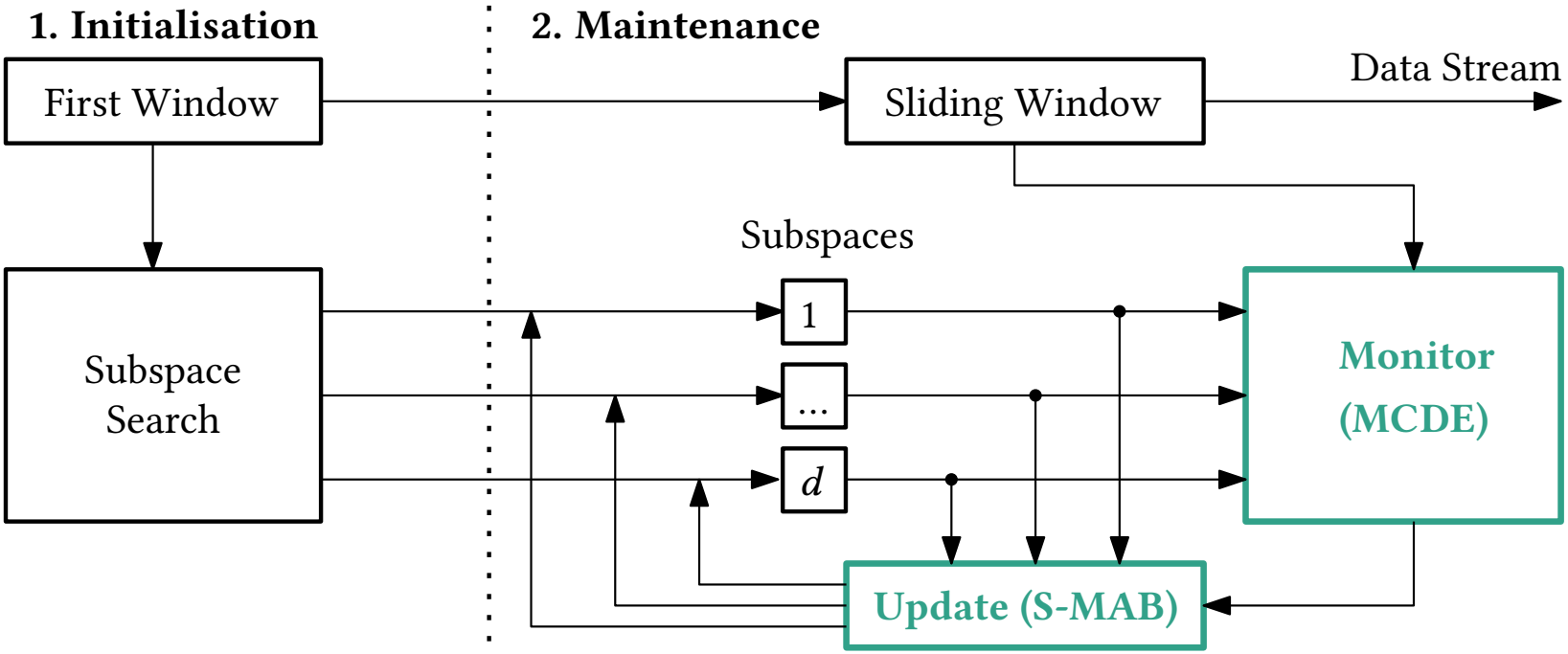
- Extend previous static work (GMD [TB19]) to streams.
- A two-phase process:



Subspace Search in Data Streams

SGMRD: Streaming Greedy Maximum Random Deviation

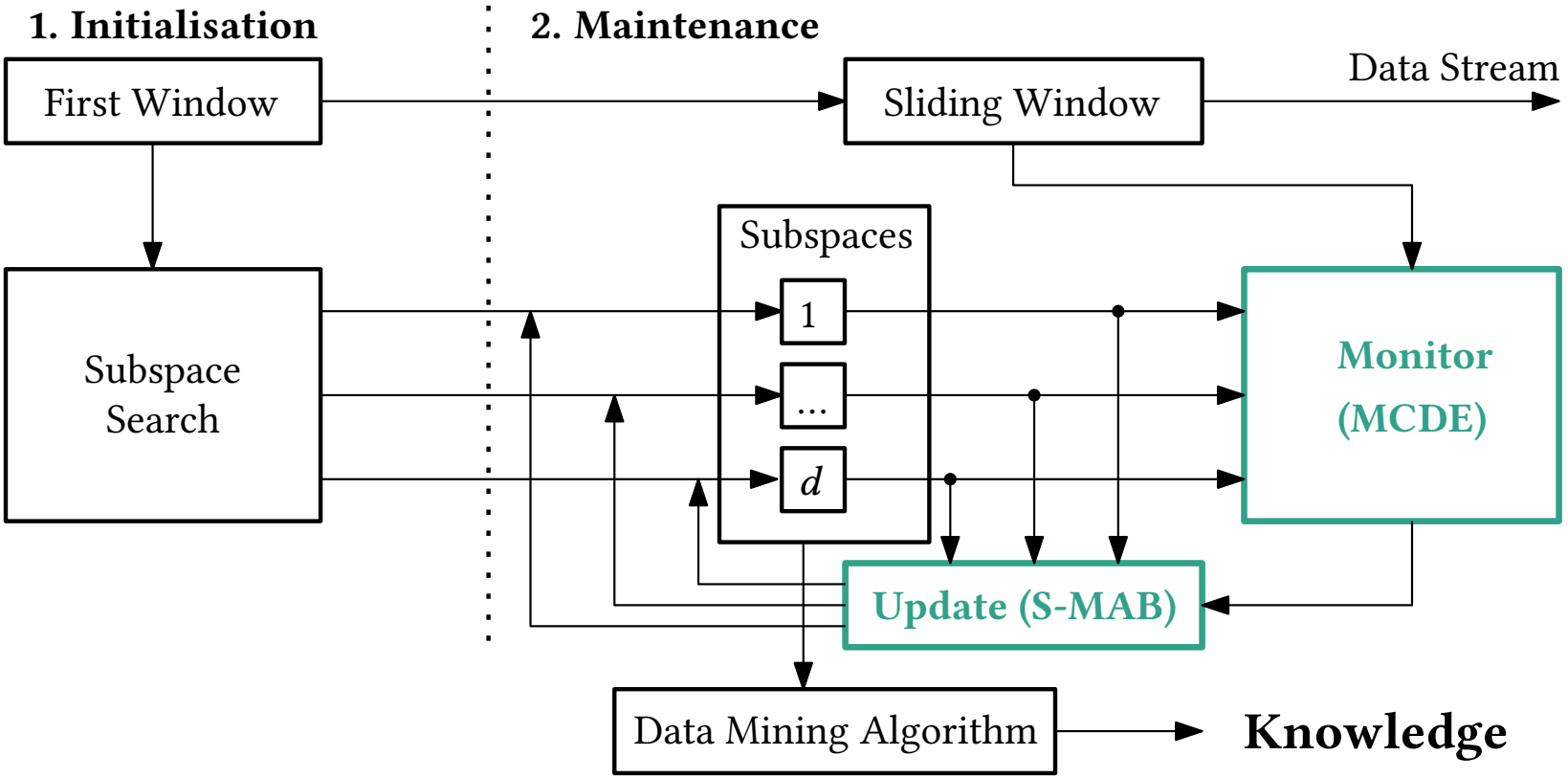
- Extend previous static work (GMD [TB19]) to streams.
- A two-phase process:



Subspace Search in Data Streams

SGMRD: Streaming Greedy Maximum Random Deviation

- Extend previous static work (GMD [TB19]) to streams.
- A two-phase process:



SGMRD improves Outlier Detection

- We use the subspaces to build an “ensemble” outlier detector.
- We find that our method outperform state-of-the-art detectors.

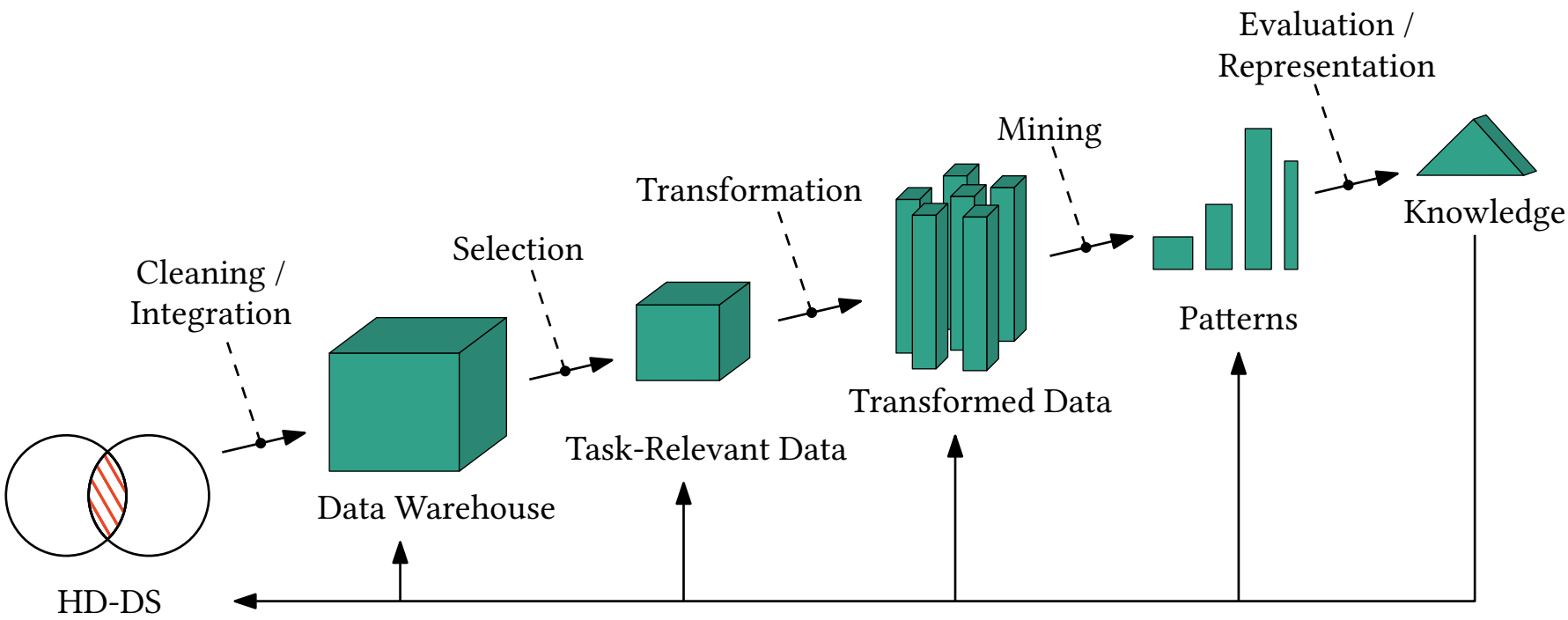
Benchmark	Approach	AUC	AvgPrc	Prc@1%	Prc@2%	Prc@5%	Rcl@1%	Rcl@2%	Rcl@5%
ACTIVITY	**SGMRD**	**97.32**	**85.39**	**94.59**	**94.83**	**94.24**	**9.44**	**18.97**	**47.10**
	LOF	93.93	61.80	74.32	64.72	64.03	7.42	12.94	32.00
	STREAMHICS	88.52	47.38	70.72	54.61	51.89	7.06	10.92	25.93
	RS-STREAM	95.95	68.23	71.62	72.58	75.00	7.15	14.52	37.48
	XSTREAM	77.71	20.41	3.60	10.14	16.31	0.36	2.02	8.13
KDDCUP99	**SGMRD**	**69.98**	**10.29**	**0.00**	**0.20**	**0.56**	**0.00**	**0.06**	**0.39**
	LOF	65.07	9.57	0.00	0.00	0.08	0.00	0.00	0.06
	STREAMHICS	57.11	7.89	0.00	0.00	0.08	0.00	0.00	0.06
	RS-STREAM	43.21	5.73	0.00	0.00	0.08	0.00	0.00	0.06
	XSTREAM	52.70	8.23	0.00	0.20	0.08	0.00	0.06	0.06
SYNTH50	**SGMRD**	**75.87**	**31.27**	**27.00**	**16.00**	**7.60**	**33.33**	**39.51**	**46.91**
	LOF	61.38	1.08	0.00	0.50	0.60	0.00	1.23	3.70
	STREAMHICS	63.90	12.00	11.00	6.00	3.40	13.58	14.81	20.99
	RS-STREAM	46.52	0.73	0.00	0.00	0.00	0.00	0.00	0.00
	XSTREAM	48.43	0.90	1.00	0.50	1.40	1.23	1.23	8.64

SGMRD improves Outlier Detection

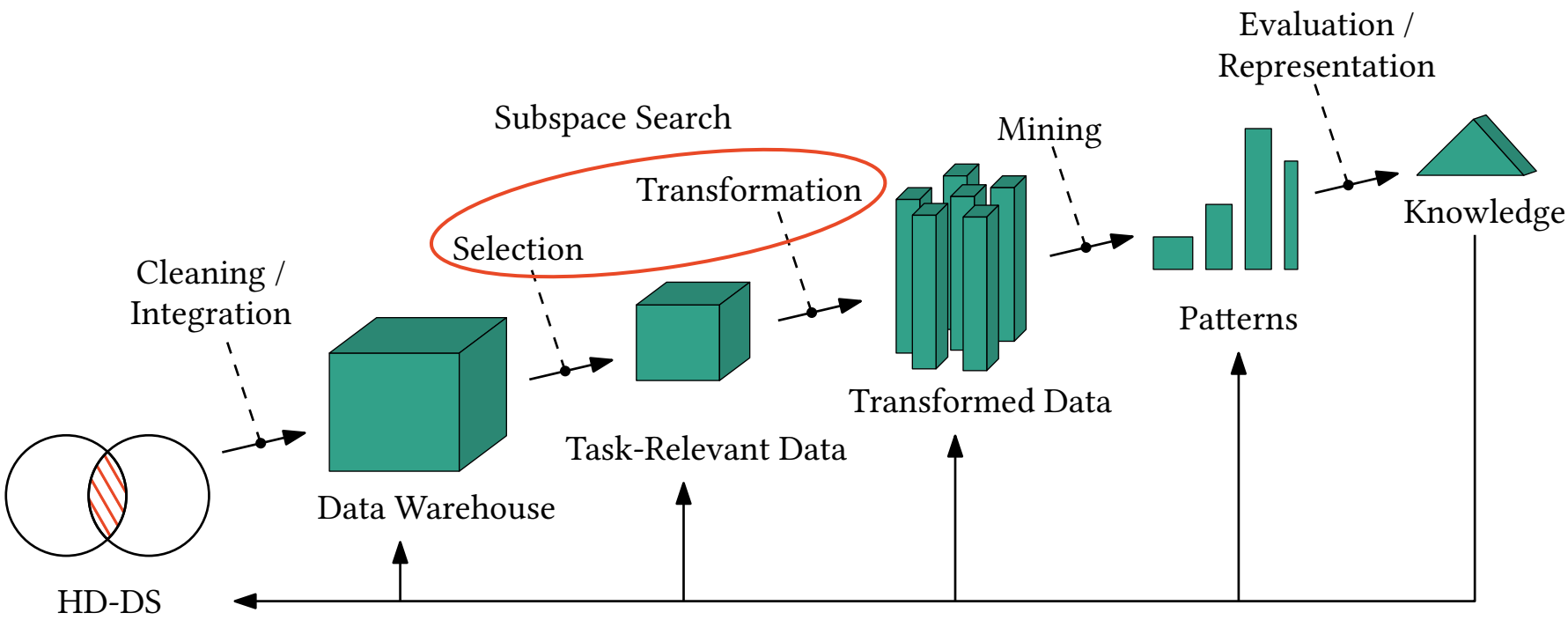
- We use the subspaces to build an “ensemble” outlier detector.
- We find that our method outperform state-of-the-art detectors.

Benchmark	Approach	AUC	AvgPrc	Prc@1%	Prc@2%	Prc@5%	Rcl@1%	Rcl@2%	Rcl@5%
ACTIVITY	**SGMRD**	**97.32**	**85.39**	**94.59**	**94.83**	**94.24**	**9.44**	**18.97**	**47.10**
	LOF	93.93	61.80	74.32	64.72	64.03	7.42	12.94	32.00
	STREAMHICS	88.52	47.38	70.72	54.61	51.89	7.06	10.92	25.93
	RS-STREAM	95.95	68.23	71.62	72.58	75.00	7.15	14.52	37.48
	XSTREAM	77.71	20.41	3.60	10.14	16.31	0.36	2.02	8.13
KDDCUP99	**SGMRD**	**69.98**	**10.29**	**0.00**	**0.20**	**0.56**	**0.00**	**0.06**	**0.39**
	LOF	65.07	9.57	0.00	0.00	0.08	0.00	0.00	0.06
	STREAMHICS	57.11	7.89	0.00	0.00	0.08	0.00	0.00	0.06
	RS-STREAM	43.21	5.73	0.00	0.00	0.08	0.00	0.00	0.06
	XSTREAM	52.70	8.23	0.00	0.20	0.08	0.00	0.06	0.06
SYNTH50	**SGMRD**	**75.87**	**31.27**	**27.00**	**16.00**	**7.60**	**33.33**	**39.51**	**46.91**
	LOF	61.38	1.08	0.00	0.50	0.60	0.00	1.23	3.70
	STREAMHICS	63.90	12.00	11.00	6.00	3.40	13.58	14.81	20.99
	RS-STREAM	46.52	0.73	0.00	0.00	0.00	0.00	0.00	0.00
	XSTREAM	48.43	0.90	1.00	0.50	1.40	1.23	1.23	8.64

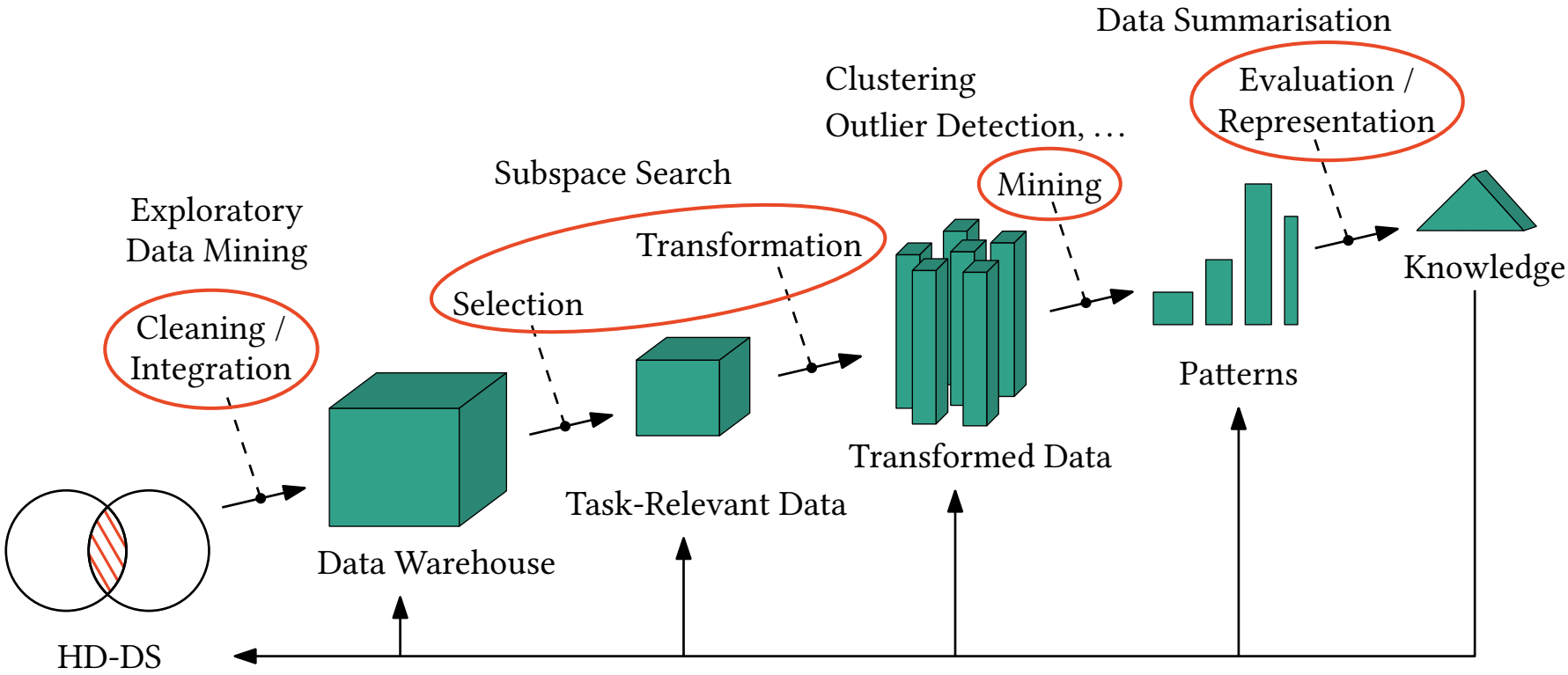
Addressing KDD in HD-DS



Addressing KDD in HD-DS



Addressing KDD in HD-DS



Conclusion

With our contributions, we can now **estimate** and **monitor dependency** in **high-dimensional data streams**, which facilitates **knowledge discovery** in this challenging, real-world setting.

- **Estimating Dependency:** MCDE [FB19] [FMKB20]
- **Monitoring:** S-MAB [FKB19]
- **Knowledge Discovery:** SGMRD [FKB20], kj-NN [FMG+20]
- Reproducibility: `https://github.com/edouardfouche`

Thanks to everyone who made this possible!

Conclusion

With our contributions, we can now **estimate** and **monitor dependency** in **high-dimensional data streams**, which facilitates **knowledge discovery** in this challenging, real-world setting.

- **Estimating Dependency:** MCDE [FB19] [FMKB20]
- **Monitoring:** S-MAB [FKB19]
- **Knowledge Discovery:** SGMRD [FKB20], kj-NN [FMG+20]
- Reproducibility: `https://github.com/edouardfouche`

Thanks to everyone who made this possible!

Conclusion

With our contributions, we can now **estimate** and **monitor dependency** in **high-dimensional data streams**, which facilitates **knowledge discovery** in this challenging, real-world setting.

- **Estimating Dependency:** MCDE [FB19] [FMKB20]
- **Monitoring:** S-MAB [FKB19]
- **Knowledge Discovery:** SGMRD [FKB20], kj-NN [FMG+20]
- Reproducibility: `https://github.com/edouardfouche`

Thanks to everyone who made this possible!

References I

[ACF02] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, 2002.

[ACFS02] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002.

[Bel57] Richard E. Bellman. *Dynamic Programming*. Princeton University Press, 1 edition, 1957.

[BG07] Albert Bifet and Ricard Gavaldà. Learning from Time-Changing Data with Adaptive Windowing. In *SDM*, pages 443–448. SIAM, 2007.

[BGE15] Jean Paul Barddal, Heitor Murilo Gomes, and Fabrício Enembreck. A survey on feature drift adaptation. In *ICTAI*, pages 1053–1060. IEEE Computer Society, 2015.

[CHL+16] Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. Combinatorial multi-armed bandit with general reward functions. In *NIPS*, pages 1651–1659, 2016.

[FB19] Edouard Fouché and Klemens Böhm. Monte carlo dependency estimation. In *SSDBM*, pages 13–24. ACM, 2019. Best Paper Award.

[FKB19] Edouard Fouché, Junpei Komiyama, and Klemens Böhm. Scaling multi-armed bandit algorithms. In *KDD*, pages 1449–1459. ACM, 2019.

[FKB20] Edouard Fouché, Florian Kalinke, and Klemens Böhm. Subspace search in data streams. In *(Currently under review)*, volume XX, pages XX–XX, 2020.

References II

[FMG+20] Edouard Fouché, Yu Meng, Fang Guo, Honglei Zhuang, Klemens Böhm, and Jiawei Han. Text outlier detection with self-supervision. In *(Currently under review)*, volume XX, pages XX–XX, 2020.

[FMKB20] Edouard Fouché, Alan Mazankiewicz, Florian Kalinke, and Klemens Böhm. A framework for dependency estimation in heterogeneous data streams. *Distributed and Parallel Databases*, 2020.

[GC11] Aurélien Garivier and Olivier Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *COLT*, volume 19 of *JMLR Proceedings*, pages 359–376. JMLR.org, 2011.

[GM11] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *ALT*, volume 6925 of *Lecture Notes in Computer Science*, pages 174–188. Springer, 2011.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.*, 58(301):13–30, 1963.

[KHN15] Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa. Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1152–1161. JMLR.org, 2015.

References III

[KKM12] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *ALT*, volume 7568 of *Lecture Notes in Computer Science*, pages 199–213. Springer, 2012.

[KMB12] Fabian Keller, Emmanuel Müller, and Klemens Böhm. Hics: High contrast subspaces for density-based outlier ranking. In *ICDE*, pages 1037–1048. IEEE Computer Society, 2012.

[McG54] William J. McGill. Multivariate information transmission. *Trans. of the IRE Professional Group on Information Theory (TIT)*, 4:93–111, 1954.

[NMV+13] Hoang Vu Nguyen, Emmanuel Müller, Jilles Vreeken, Fabian Keller, and Klemens Böhm. CMI: an information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In *SDM*, pages 198–206. SIAM, 2013.

[NMV+14] Hoang Vu Nguyen, Emmanuel Müller, Jilles Vreeken, Pavel Efros, and Klemens Böhm. Multivariate maximal correlation analysis. In *ICML*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 775–783. JMLR.org, 2014.

[NMV16] Hoang Vu Nguyen, Panagiotis Mandros, and Jilles Vreeken. Universal dependency analysis. In *SDM*, pages 792–800. SIAM, 2016.

[SS07] Friedrich Schmid and Rafael Schmidt. Multivariate extensions of spearman's rho and related statistics. *Statistics & Probability Letters*, 77(4):407 – 416, 2007.

[TB19] Holger Trittenbach and Klemens Böhm. Dimension-based subspace search for outlier detection. *Int. J. Data Sci. Anal.*, 7(2):87–101, 2019.

References IV

[Tho33] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

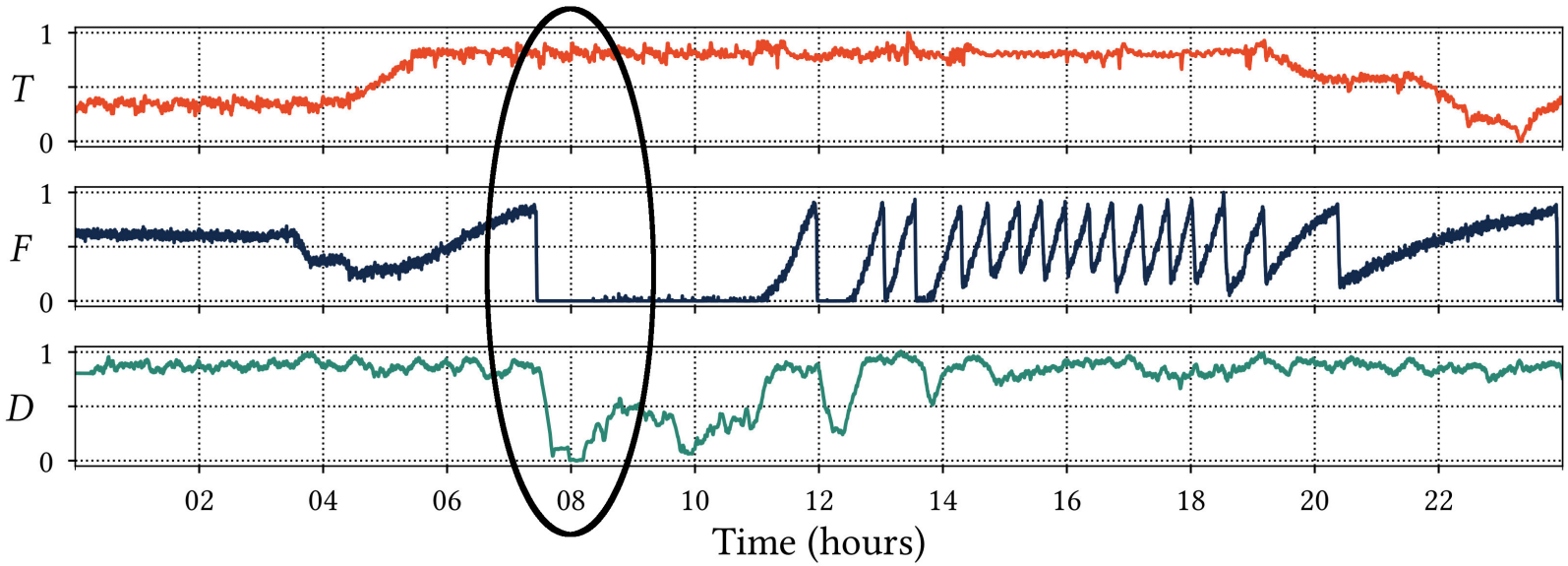
[UNK10] Taishi Uchiya, Atsuyoshi Nakamura, and Mineichi Kudo. Algorithms for adversarial bandit problems with multiple plays. In *ALT*, volume 6331 of *Lecture Notes in Computer Science*, pages 375–389. Springer, 2010.

[Wat60] Michael Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development*, 4(1):66–82, 1960.

Dependency helps to identify patterns

An example with two streams:

- T: Temperature in the reactor
- F: Filling level of the cyclone
- D: Estimation of dependency between T and F



$\rightarrow$ Monitoring Dependency shows interruption in production

Benchmarking Dependency Estimation

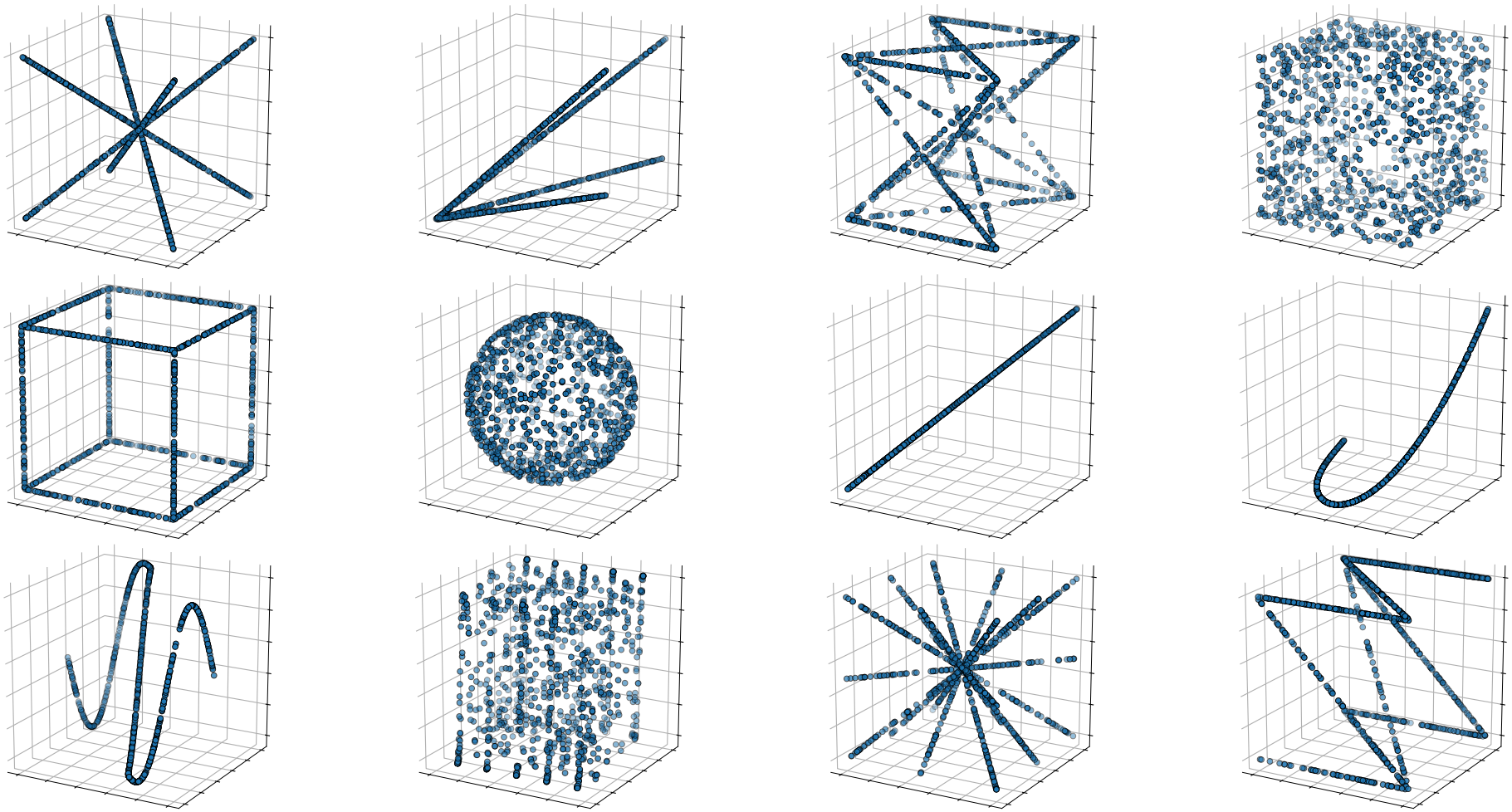
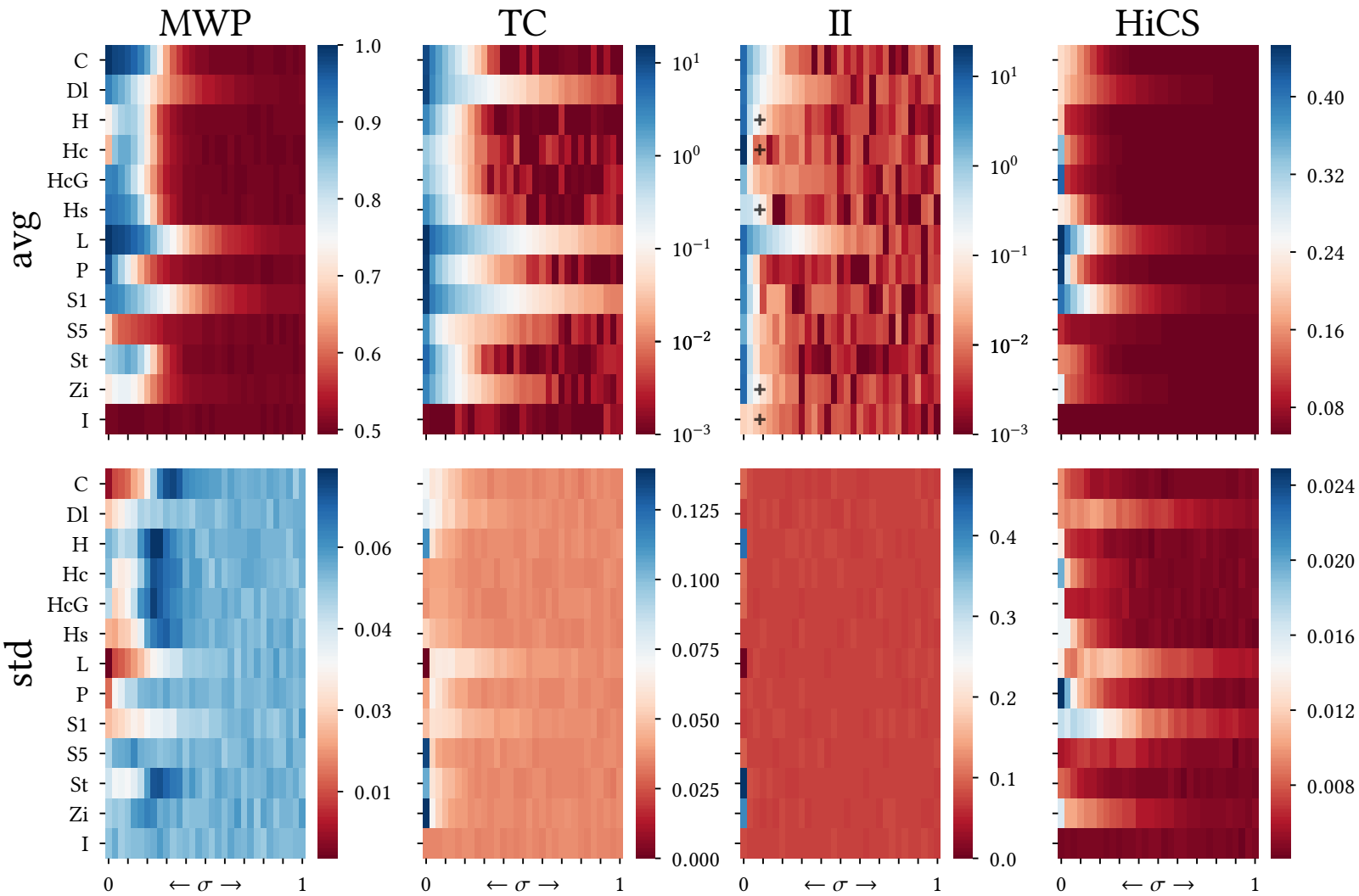
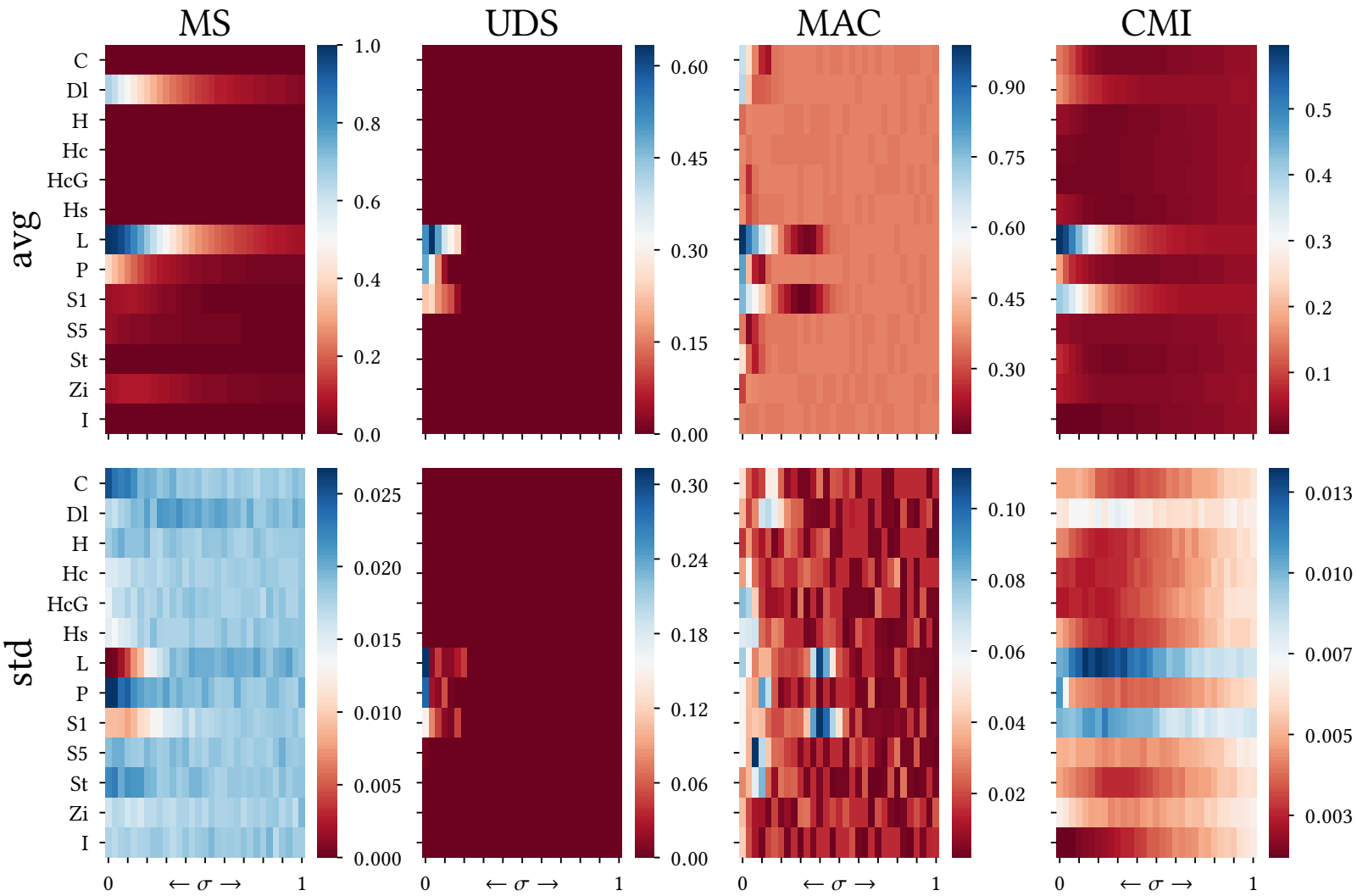


Figure 2: An assortment of 12 dependencies

Statistical Power of MCDE



Statistical Power of MCDE



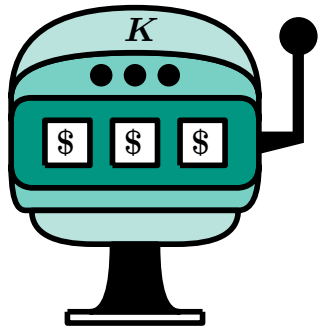
The “Classical” MAB

Let there be a set of K arms, $[K] = \{1, \ldots, K\}$.

The rewards of each arm $i \in [K]$ follow a Bernoulli distribution $\mathcal{B}_i$ with mean μ_i.

At each round $t = 1, \ldots, T$:

- The forecaster chooses **one** arm $i \in [K]$
- Then, they observe a reward $X_t \sim \mathcal{B}_i$
- They update their estimation $\hat{\mu}_i$ of μ_i



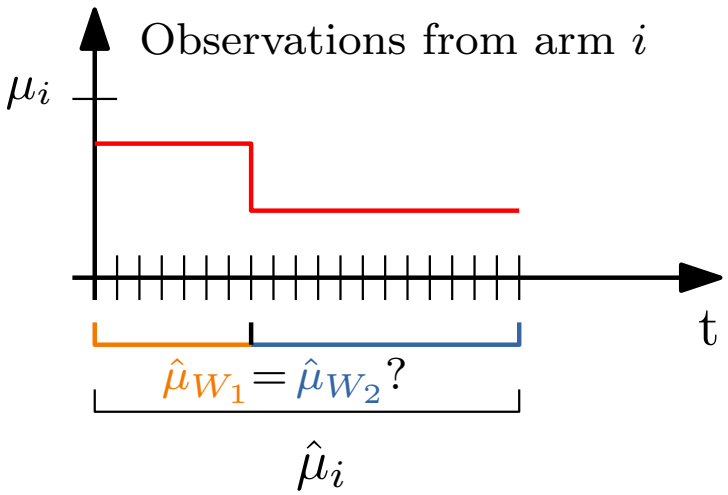
The goal of the forecaster is to maximize their gain, i.e., $\sum_{t=1}^{T} X_t$

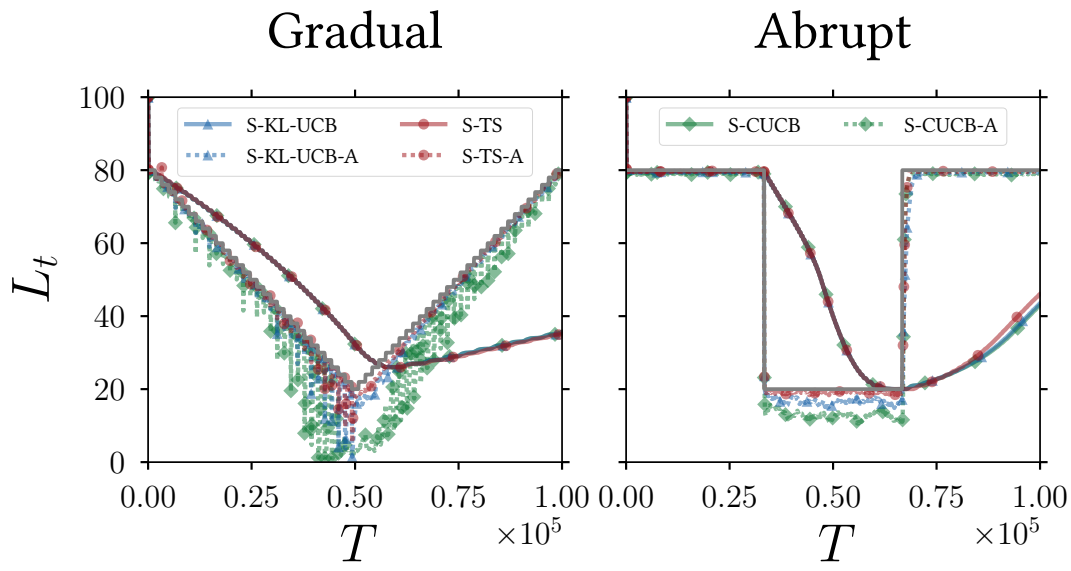
S-MAB with ADWIN

Problem: The expectations μ_i of each arm $i \in [K]$ might change.

We use Adaptive Windowing (ADWIN) [BG07]

- Maintain $\hat{\mu}_i$ for each arm over a sliding window of adaptive length





→ Scaling Thompson Sampling with ADWIN (S-TS-A)

Deploying S-MAB @ Bioliq

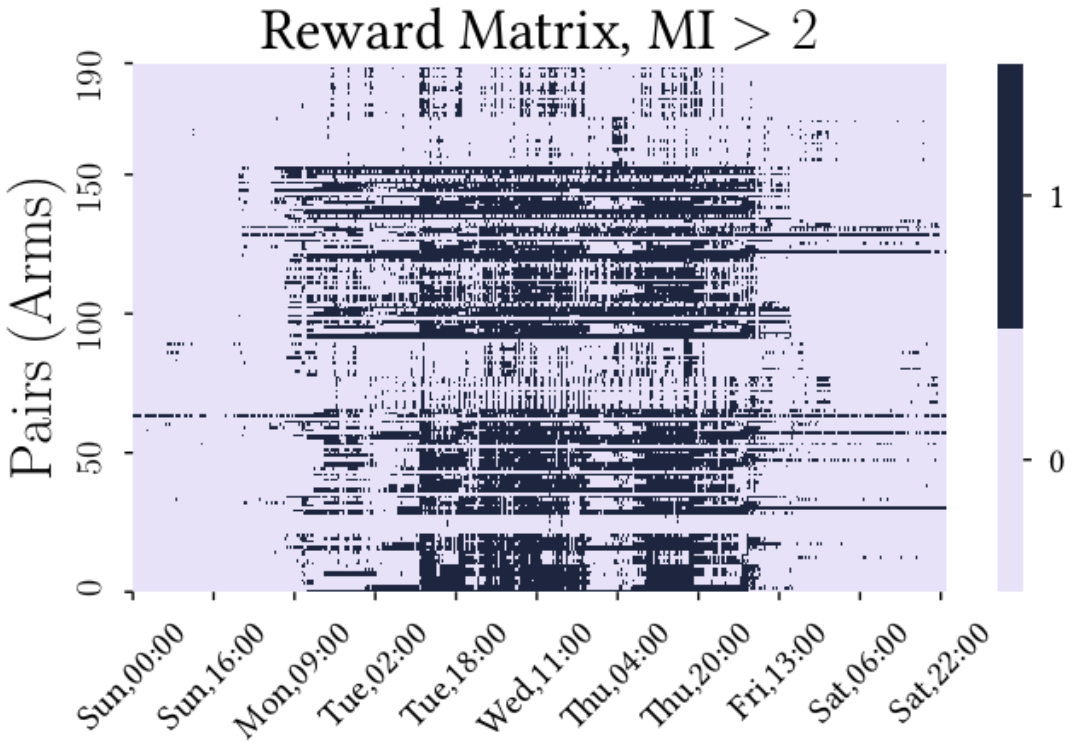
Bioliq: 20 sensors, 1 week data

- Mutual Information (MI) over sliding window
  - Window Size: 1000 points ($\sim$ 15 minutes)
  - Step size: 100 points

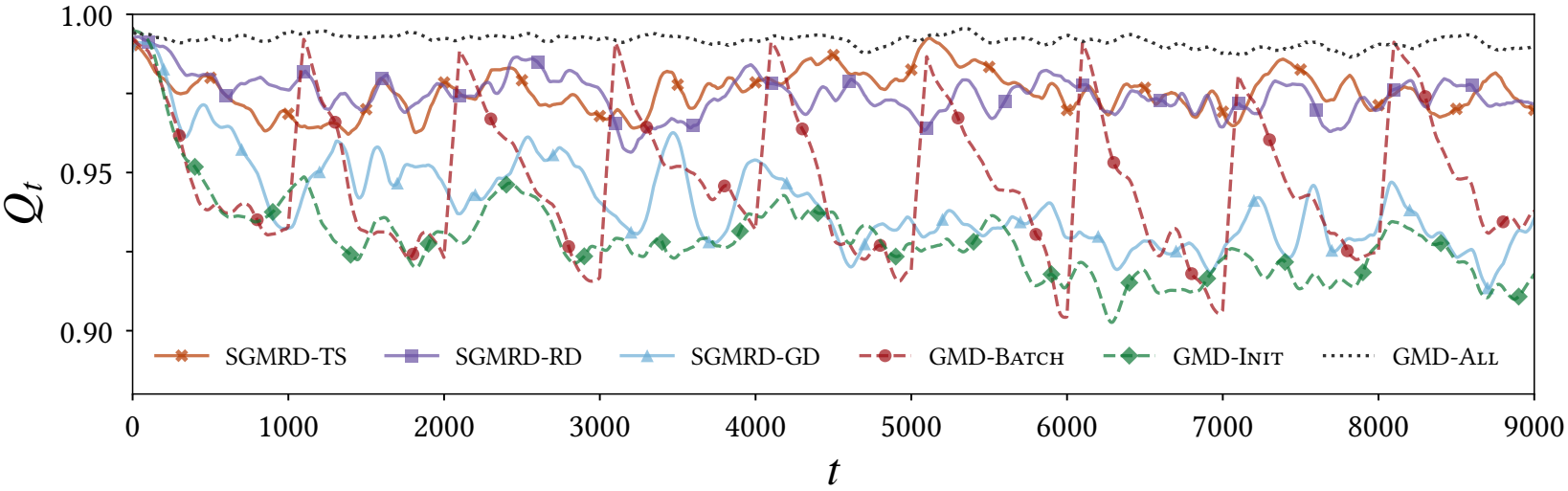
S-MAB as a "monitoring system"

- If MI ≥ 2, Reward $= 1$

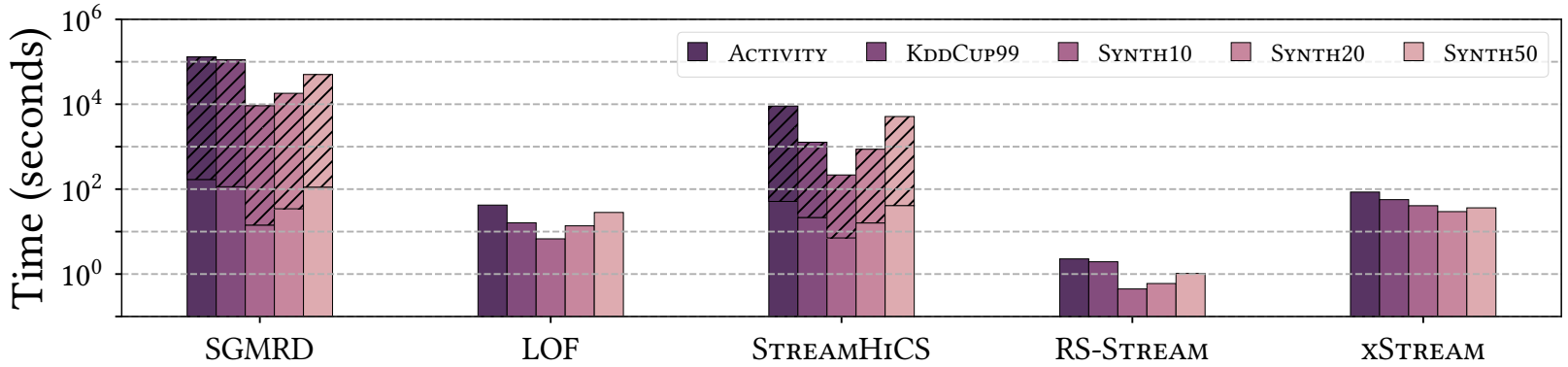
The environment is non-static !



SGMRD: Subspace Monitoring

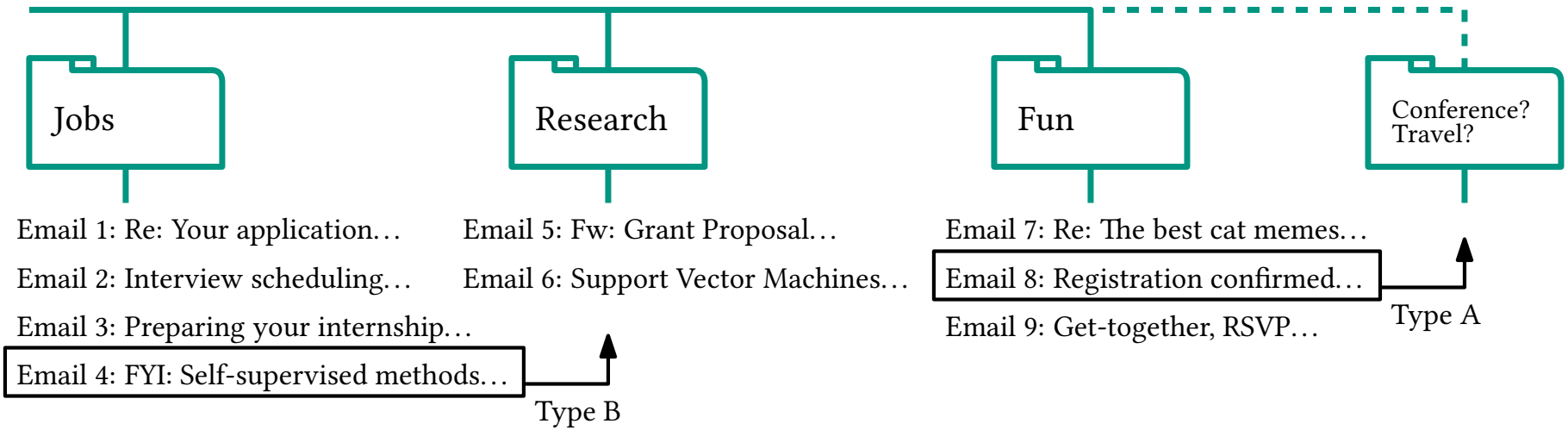


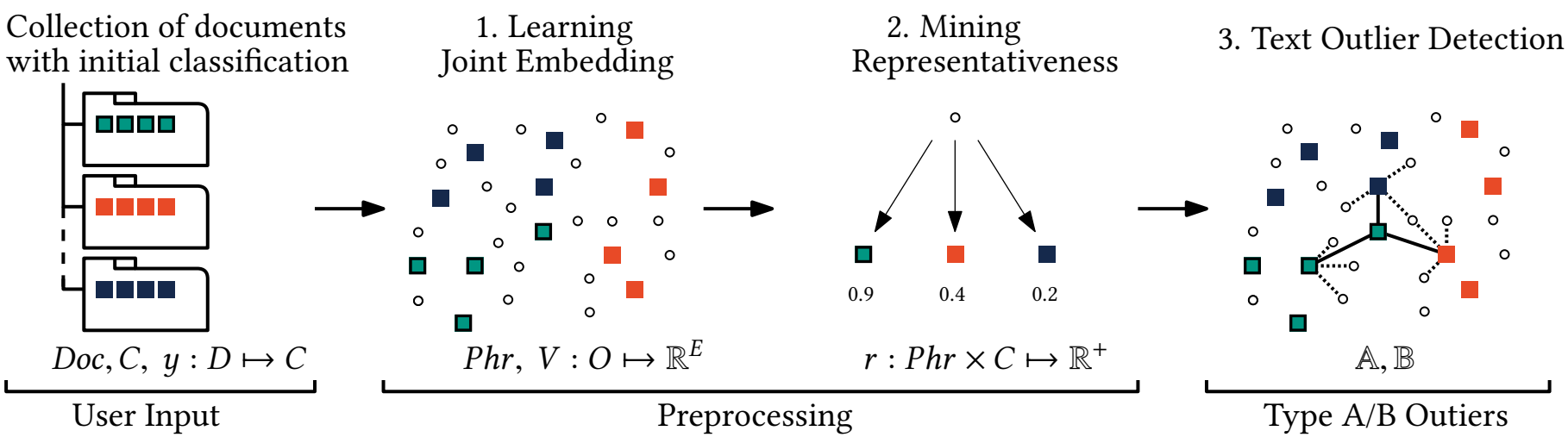
Average quality of monitored subspaces



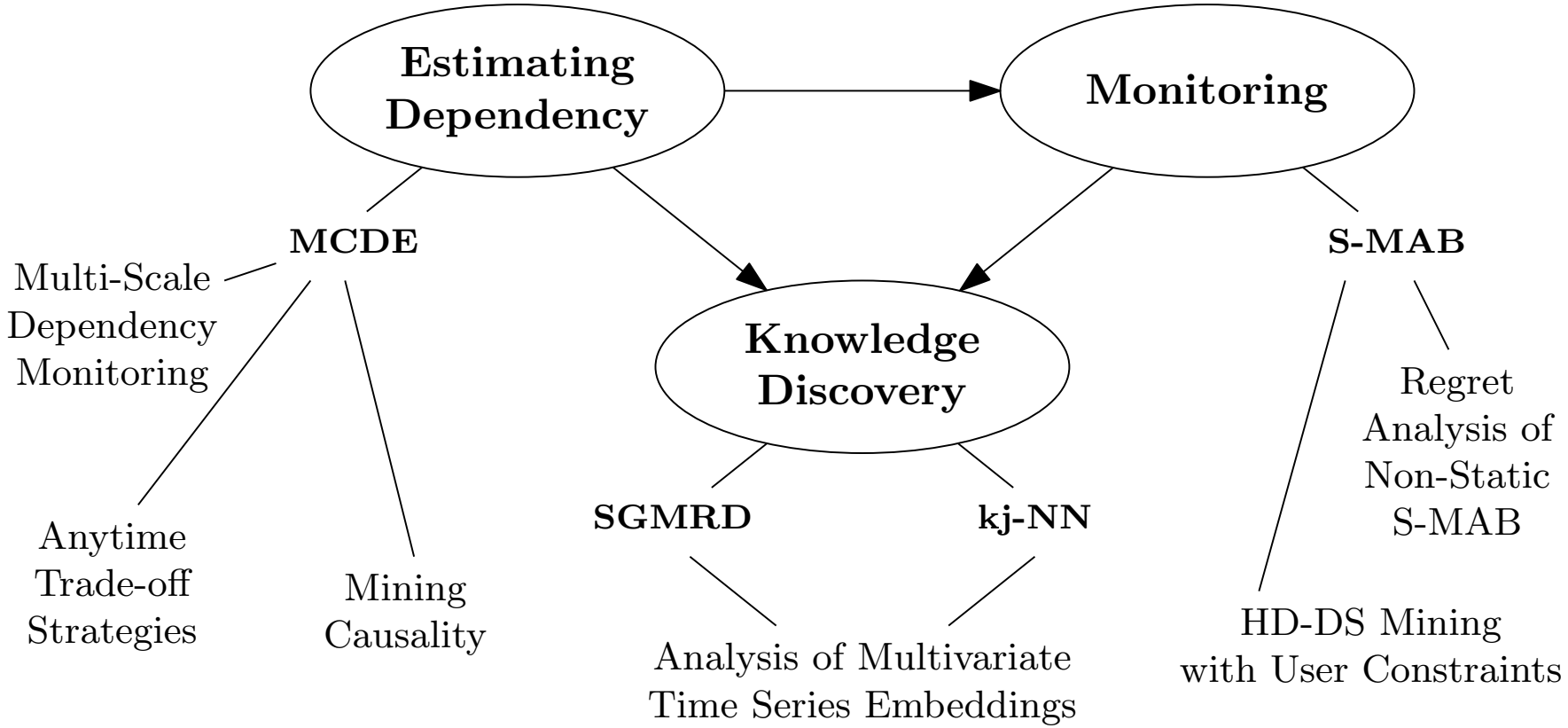
Outlier detection runtime

Text Outlier Detection (kj-NN)





Outlook


Estimating Dependency
Monitoring
MCDE
S-MAB
Multi-Scale Dependency Monitoring
Knowledge Discovery
Regret Analysis of Non-Static S-MAB
Anytime Trade-off Strategies
Mining Causality
SGMRD
kj-NN
HD-DS Mining with User Constraints
Analysis of Multivariate Time Series Embeddings

Mining Causality

- Dependencies may be spurious
  - e.g., see Simpson's paradox
- With MCDE, we may detect such spurious relationships.
  - Build a "dependency network"
  - In HD-DS