Fast and Multi-aspect Mining of Complex Time-stamped Event Streams

Kota Nakamura, Yasuko Matsubara, Koki Kawabata,
Yuhei Umeda, Yuichiro Wada, Yasushi Sakurai

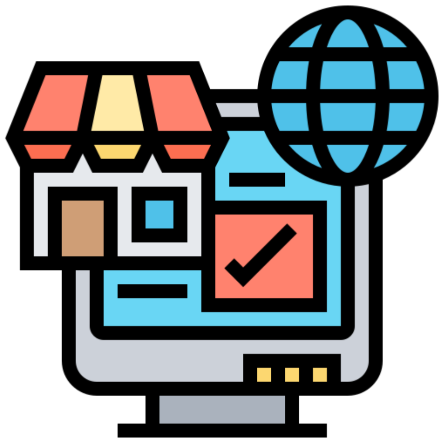
THE WEB CONFERENCE ACM

SANKEN OSAKA UNIVERSITY

FUJITSU

Complex Time-stamped Event Streams are Everywhere

- ❑ A huge, online stream of time-stamped events with multiple attributes



E-commerce

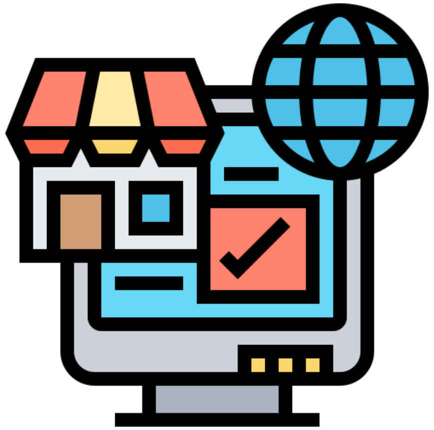
Local mobility

CyberSecurity

Complex Time-stamped Event Streams are Everywhere

❑ A huge, online stream of time-stamped events with multiple attributes

3 attributes (M=3)



E-commerce

TimeStamp	Brand	Item category	Price
2023-04-30-21:01	Tefal	Kettle	$45
2023-04-30-21:01	Bosch	Refrigerator	$200
2023-04-30-21:02	Samsung	TV	$650
2023-04-30-21:03	Sony	Portable audio	$200
2023-04-30-21:08	LG	TV	$400
2023-04-30-21:11	Dell	Monitor	$90
2023-04-30-21:13	Philips	Headphones	$190

Complex Time-stamped Event Streams are Everywhere

❑ A huge, online stream of time-stamped events with multiple attributes

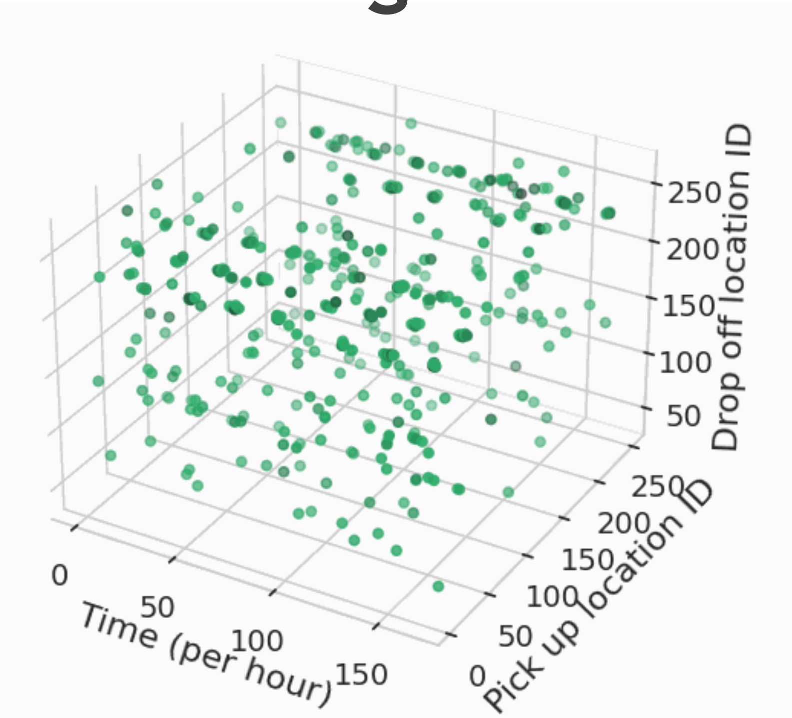
2 attributes (M=2)

TimeStamp	Pick-up location	drop-up location
2023-04-30-20:01	Museum C	Museum B
2023-04-30-21:02	Cinema A	Street C
2023-04-30-21:06	School D	Restaurant A
2023-04-30-21:18	Office A	Station A
2023-04-30-22:08	Street A	University D
2023-05-01-09:11	Hotel B	Airport A
2023-05-01-11:13	Station C	Street B

Limitations & Challenges

Complex time-stamped event streams …

derail existing methods and even our interpretation



3rd -order tensor stream:
each aspect indicates each attributes

Because this is…

High-order tensor streams

- **High-dimensional**
- **Sparse**
- **Semi-infinite**

Our Questions

Q. How can we summarize large, dynamic high-order tensor streams?
Q. How can we see any hidden patterns, rules, and anomalies?

Our Questions

Q. How can we summarize large, dynamic high-order tensor streams?
Q. How can we see any hidden patterns, rules, and anomalies?

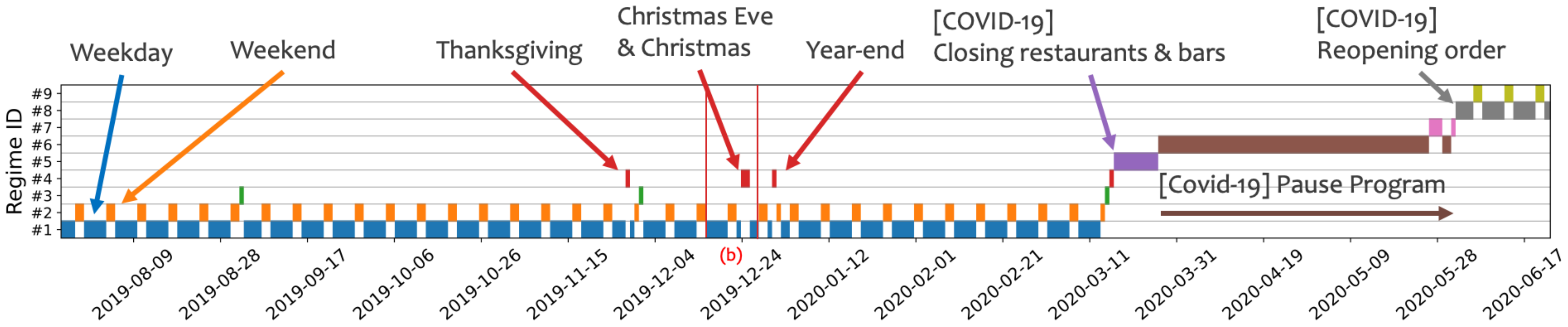
Our answer is …

to focus on two types of patterns,
Regimes and **Components**

Our Answer: Regimes and Components

Regimes: Distinct time-evolving patterns

Local mobility
(NY-taxi rides)



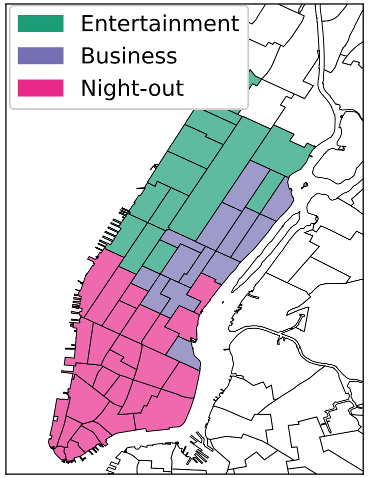
→Time

❑Summarize **semi-infinite** event stream into a handful number of segments

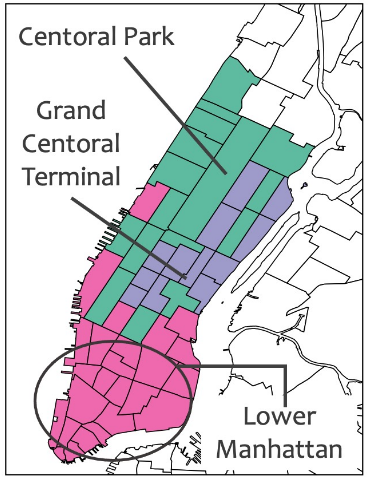
Our Answer: Regimes and Components

Components: Multi-aspect latent trends

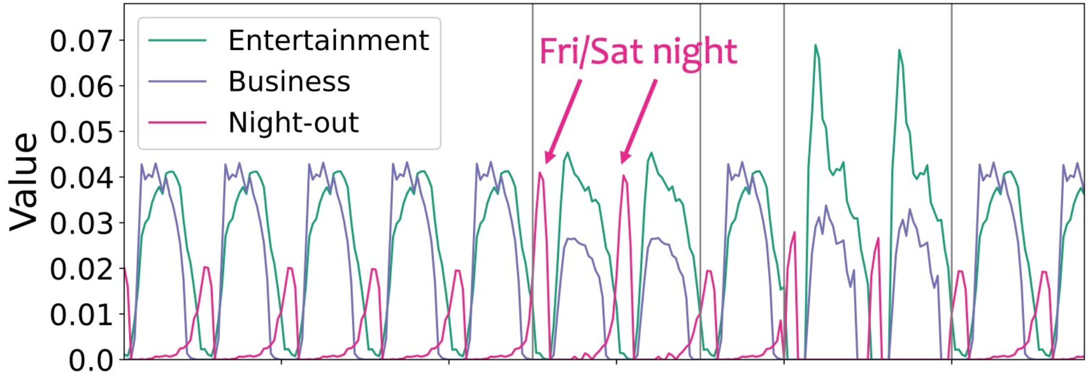
Local mobility
(NY-taxi rides)



Pick-up location



Drop-off location



Timestamp (Pick-up time)

- ❑Summarize **high-deimensional** and **sparse** events into major groups

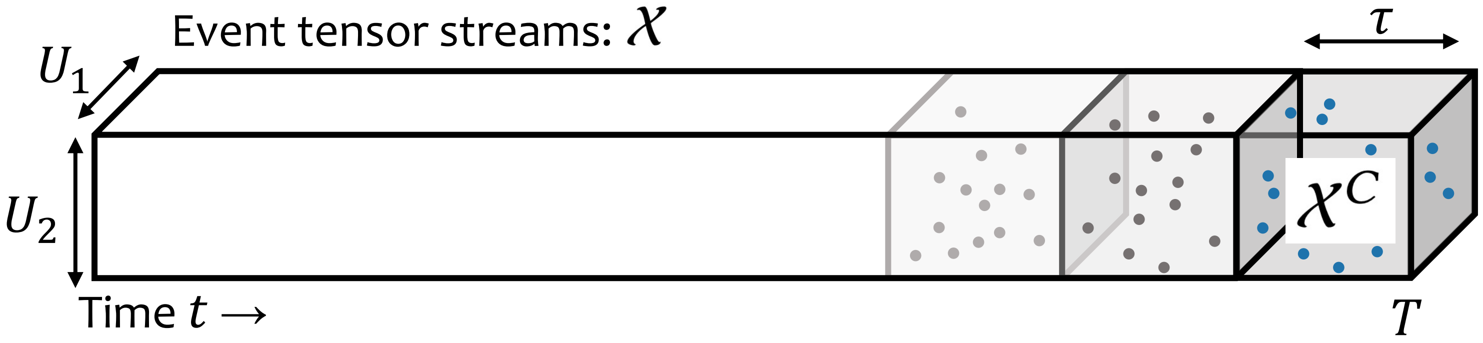
Outline

- Introduction
- ▶ Model
- Algorithm
- Experiments
- Conclusion

Our Settings: Complex Time-stamped Event Streams

- Event stream, which consist of {M attributes + Timestamp}
  → **M+1th-order tensor stream** $\mathcal{X} \in \mathbb{N}^{U_1 \times \cdots \times U_M \times T}$

- Continuously obtain **current tensors** $\mathcal{X}^C \in \mathbb{N}^{U_1 \times \cdots \times U_M \times \tau}$



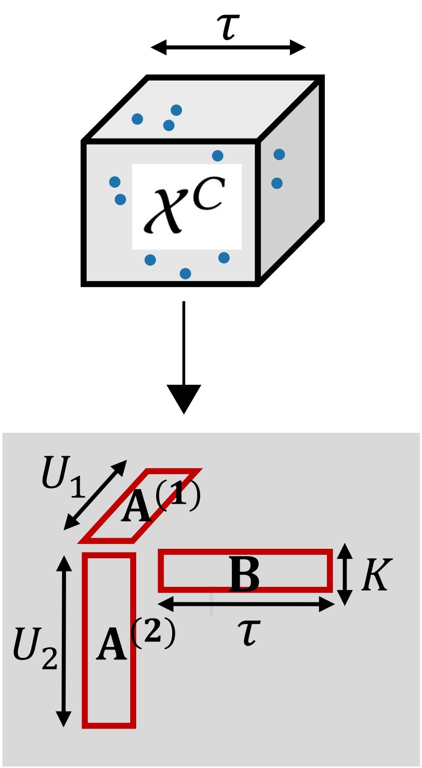
Proposed Model

Q1. What is the simplest mathmatical model for components?
Q2. How can we represent regimes and summarize the whole stream?
Q3. How can we formulate the summarization problem?

G1. Multi-aspect component factorization
G2. Compact description
G3. Problem formulation in a data compression paradigm

G1. Multi-aspect Component Factorization

Goal: to describe a high-dimensional and sparse tensor $\mathcal{X}^C$ as compact and interpretable model



Component matrices

Multi-aspect Component factorization

- ❑ Model the generative process of events
- ❑ Assume that there are K major trends/**components**
- ❑ k-th **component** is defined by probability distribution w.r.t. M attributes and time

$$\mathbf{A}_k^{(m)} \in \mathbb{R}^{U_m}, \mathbf{B}_t \in \mathbb{R}^K$$

$$\mathbf{A}_k^{(m)} \sim \text{Dirichlet}(\alpha^{(m)}),\ \mathbf{B}_t \sim \text{Dirichlet}(\beta)$$

G1. Multi-aspect Component Factorization

The generative process:

- For each component $k = 1, \ldots, K$:
  - For each attribute $m = 1, \ldots, M$:
    - $\mathbf{A}_k^{(m)} \sim \boxed{\mathrm{Dirichlet}(\Sigma_{l=1}^{L} \alpha^{(m)}{}_l \hat{\mathbf{A}}_k^{(m)})}$
- For each time $t = 1, \ldots, \tau$:
  - $\mathbf{B}_t \sim \boxed{\mathrm{Dirichlet}(\Sigma_{l=1}^{L} \beta_l \hat{\mathbf{B}}_t)}$
  - For each entry $j = 1, \ldots, N_t$:
    - $z_{t,j} \sim \mathrm{Multinomial}(\mathbf{B}_t)$ // Draw a latent component $z_{t,j}$
    - For each attribute $m = 1, \ldots, M$:
      - $e_{t,j}^{(m)} \sim \mathrm{Multinomial}(\mathbf{A}_{z_{t,j}}^{(m)})$, // Draw a unit in each attribute

Capture
temporal dependencies
without storing tensors

Details in paper

❑**Summarize** sparse activity into K components

❑**Mutli-aspect property:** handle arbitrary-order tensors

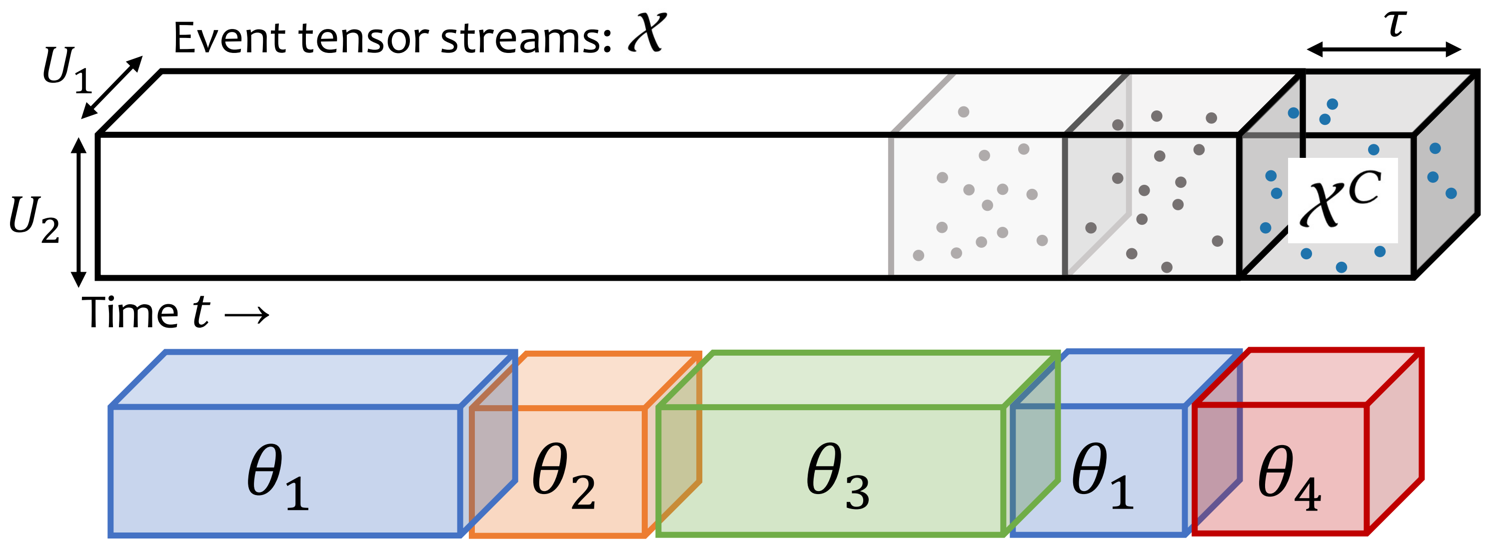
❑**Online setting:** capture temporal dependencies without storing tensors

G2. Compact description

Goal: to represent the whole stream $\mathcal{X}$, containing distinct dynamical patterns

Regime:

$$\theta = \{\{\mathbf{A}^m\}_{m=1}^{M}, \mathbf{B}\}$$



Compact description: $\mathcal{C} = \{R, \Theta, G, \mathcal{S}\}$

- ❑ the number of regimes R and the regime set Θ
- ❑ the number of segments G and the assignments $\mathcal{S}$

G3. Problem Formulation: Data Compression Paradigm

What is good summarization?

- ❑ **Minimum Description Length (MDL) principle:**
  "the more we can compress the data,
  the more we can learn about their underlying patterns"
- ❑ **Evaluate the total encoding cost**,
  which is used to losslessly compress the original data streams

Summarization Problem

Find the compact description $\mathcal{C}$, which minimizes the total encoding cost

$$< \mathcal{X}; \mathcal{C} > = \underbrace{< \mathcal{C} >}_{\text{Model coding cost}} + \underbrace{< \mathcal{X} | \mathcal{C} >}_{\text{Data coding cost}}$$

G3. Problem Formulation: Data Compression Paradigm

- **Model Coding Cost:** the number of bits needed to describe the model $\mathcal{C}$
- **Data Coding Cost:** the coding cost of data $\mathcal{X}$ given the model $\mathcal{C}$

Dimensionality

Number of regimes

Number of segments

Coding cost of each segment given regimes

$$< \mathcal{X}; \mathcal{C} > = < \mathcal{C} > + < \mathcal{X}|\mathcal{C} >$$
$$= \underline{< d >} + \underline{< R >} + \underline{< G >}$$
$$+ \sum_{r=1}^{R} \underline{< \theta >} + \sum_{g=1}^{G} \underline{< s_g >} + \underline{< \mathcal{X}|\mathcal{C} >} . \quad (6)$$

$$\boxed{\sum_{r=1}^{R} -\log P(\mathcal{X}[r]|\theta_r)}$$

Each regime

Each segment

Details in paper

Outline

CubeScope

SANKEN

Streaming Algorithm: CubeScope

Given:

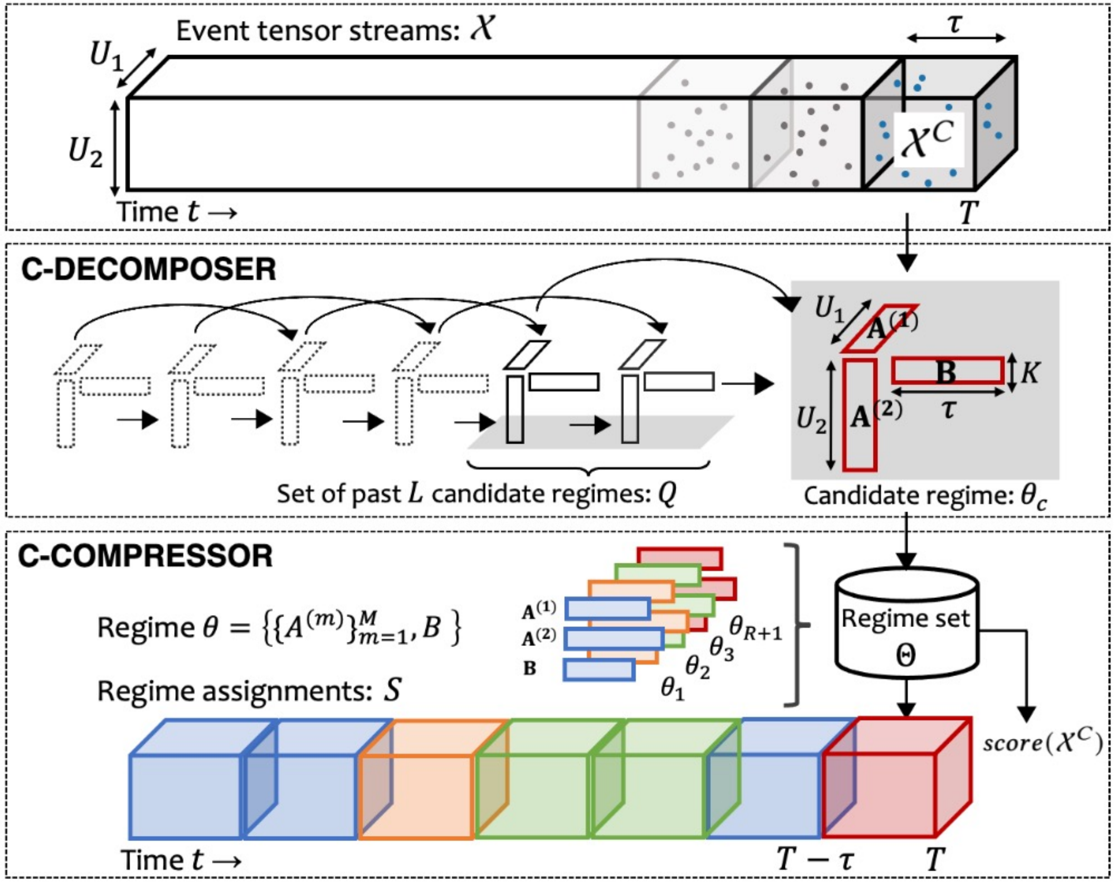
Complex time-stamped event streams

CubeScope

- ❑ **Finds**
  - ❑ Components (Multi-aspect latent trends/groups)
  - ❑ Regimes (Distinct time-evolving patterns)
- ❑ **Detects** anomalies and their types

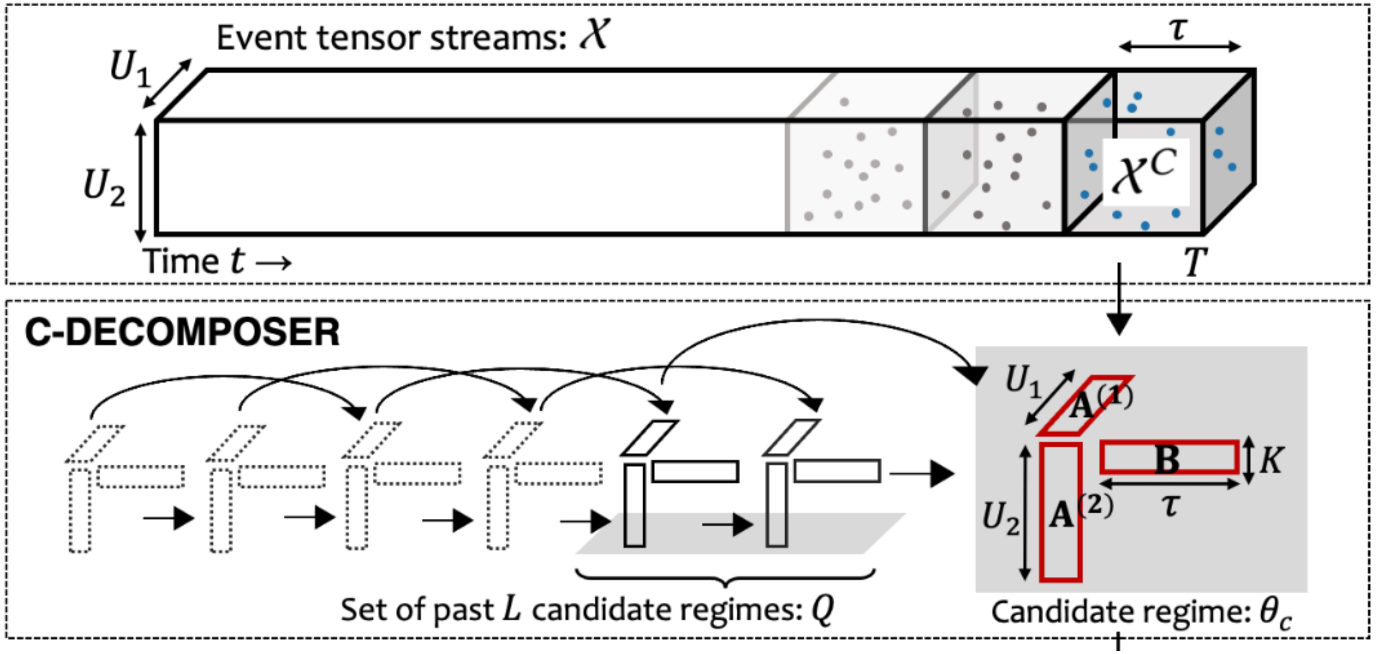
Streaming Algorithm: CubeScope

Our **CubeScope** consists of two sub-algorithms:



- ❑ **C-Decomposer:**
  - ❑ incrementally monitors $\mathcal{X}^C$
  - ❑ estimates a candidate regime θ_c
- ❑ **C-Compressor:**
  - ❑ Updates the compact description $\mathcal{C}$
  - ❑ Measures the anomalousness of $\mathcal{X}^C$

C-Decomposer



❑Regime estimation

with collapsed Gibbs sampling

$$p(z_{u_1,\ldots,u_M,t} = k \mid \mathcal{X}^C, \mathbf{B}', \hat{\mathbf{B}}, \beta, \{\mathbf{A}^{(m)\prime}, \hat{\mathbf{A}}^{(m)}, \alpha^{(m)}\}_{m=1}^{M})$$

$$\propto \frac{b'_{t,k} + \sum_{l=1}^{L} \beta_l \hat{b}_{t,k}}{\sum_{k=1}^{K} b'_{t,k} + L\beta} \cdot \prod_{m=1}^{M} \frac{a^{(m)\prime}_{u_m,k} + \sum_{l=1}^{L} \alpha^{(m)}{}_l \hat{a}^{(m)}_{u_m,k}}{\sum_{u=1}^{U_m} a^{(m)\prime}_{u,k} + L\alpha^{(m)}},$$

$$\tilde{a}^{(m)}_{u,k} \propto \frac{a^{(m)}_{u,k} + \sum_{l=1}^{L} \alpha^{(m)}{}_l \hat{a}^{(m)}_{u,k}}{\sum_{u=1}^{U_m} a^{(m)}_{u,k} + L\alpha^{(m)}}, \tilde{b}_{t,k} \propto \frac{b_{t,k} + \sum_{l=1}^{L} \beta_l \hat{b}_{t,k}}{\sum_{k=1}^{K} b_{t,k} + L\beta}$$

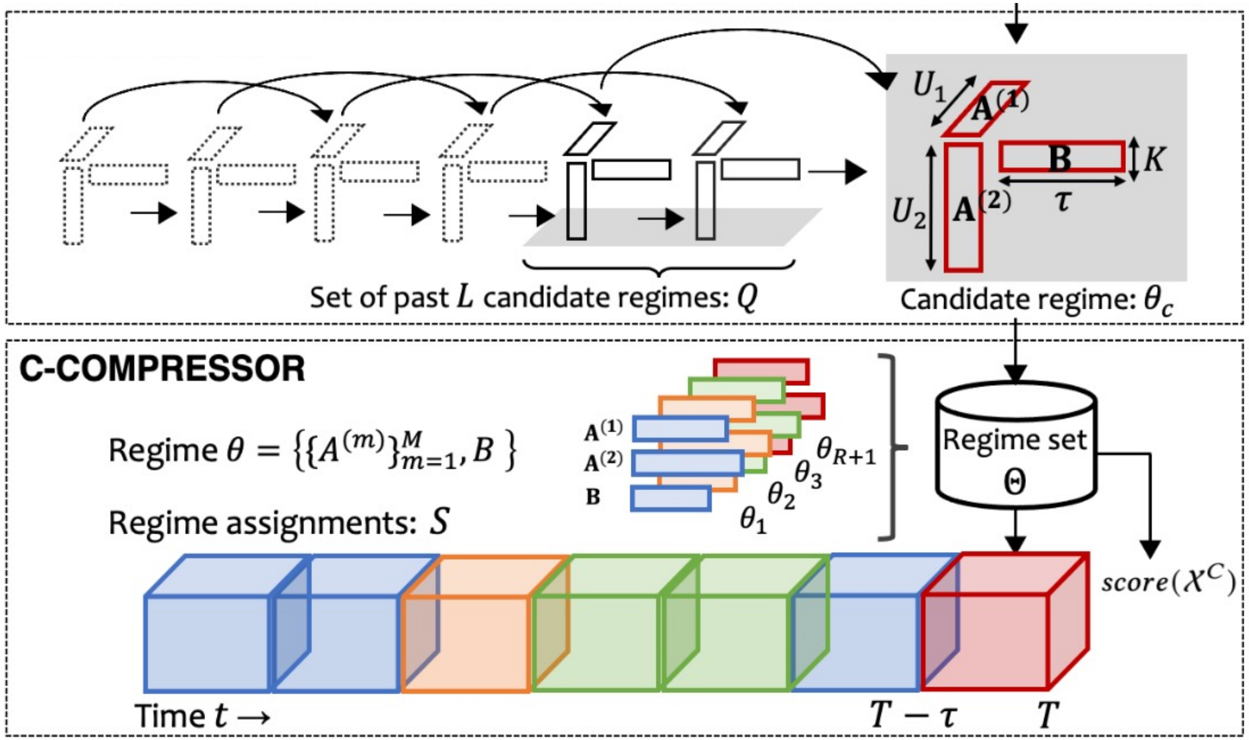
C-Decomposer is **Efficient**

- ❑ Independ on dimensionality, i.e., it takes $O(N)$, N: the number of events
- ❑ Conventional algorithms (e.g., ALS) are expensive for high-order tensor these scale w.r.t. all the attributes, i.e., take $O(\prod_{m=1}^{M} U_m)$

C-Compressor

❑Insertion-based algorithm:

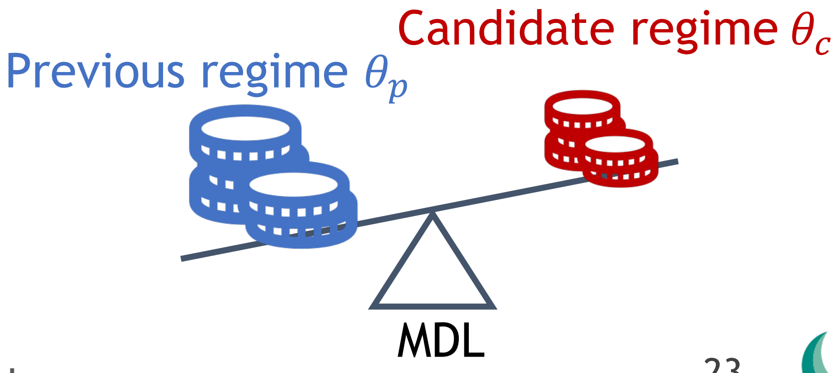
Maintains a resonable description $\mathcal{C}$ for $\mathcal{X}$ and generates new regime if necessary



❑ Compares encoding costs $< \mathcal{X}^C; \theta_* >$ between θ_c and θ_p

$$< \mathcal{X}^C; \theta_* > = \Delta < \mathcal{C} > + < \mathcal{X}^C | \theta_* >, \tag{9}$$

$$\begin{aligned} \Delta < \mathcal{C} > &= \log^*(R+1) - \log^*(R) + < \theta_* > \\ &\quad + \log^*(G+1) - \log^*(G) + < s >, \end{aligned} \tag{10}$$



C-Compressor: Stream Anomaly Detection

❑ Compression-based anomaly detection

- ❑ Higher compression cost → higher anomalousness score

$$norm = \arg\max_{r \in R} |\mathcal{S}_r^{-1}|,$$

$$score(\mathcal{X}^C) = < \mathcal{X}^C | \theta_{norm} >,$$

C-Compressor is Adaptive

- ❑ The concept of **normal changes** over time
  → Adaptively change the baseline to judge incoming tensors
- ❑ Data streams **contain multiple anomalies** over time
  → Discard anomalies from the baseline

Outline

Introduction

Model

Algorithm

▶ Experiments

Conclusion

Experimental Questions

We aim to evaluate that *CubeScope* has …

Q1. Effectiveness:
How successfully does it discover meaningful patterns?

Q2. Accuracy:
How accurately does it achieve modeling, clustering, and anomaly detection?

Q3. Scalability:
How does it scale in terms of computational time?

Experimental Setup

12 datasets
(8 real-world datasets + 4 synthetics)

Dataset	The form of entry	Order
Local Mobility: Ride information attributes & timestamp → #rides		
#1 *NYC-Taxi* [8]	*(Pick-up/Drop-off location ID, Time)*	3
#2 *Bike-Share* [2]	*(User's age, Start/End station ID, Time)*	4
E-commerce: Purchase information attributes & timestamp → #purchases		
#3 *Jewelry* [4]	*(Price, Brand, Gem, Accessory type, Time)*	4
#4 *Electronics* [3]	*(Brand, Item category, Time)*	3
Network traffic/intrusion: Access detail attributes & timestamp → #accesses		
#5 *AirForce* [5]	*(Protocol type, Service, Flag, Land, Duration Src/Dst bytes, Wrong fragment, Urgent, Time)*	10
#6 *External* [1]	*(Proto, Src/Dst IP Addr, Src/Dst Pt, Flags,Duration,Packets,Bytes, Time)*	10
#7 *OpenStack* [1]	”	10
#8 *Kyoto* [9]	*(Src/Dst bytes, Count, Same srv/Serror/Srv serror rate, Dst host serror rate/same src port rate/srv serrors rate, Dst host count/srv count, Duration,Service,Flag,Time)*	15

12 Baselines

Probabilistic generative models
- ❑ LDA
- ❑ NTM
- ❑ TriMine

Clustering approaches for time series, tensor, and data streams
- ❑ K-means
- ❑ TICC
- ❑ CubeMarker
- ❑ T-LSTM
- ❑ DBSTREAM

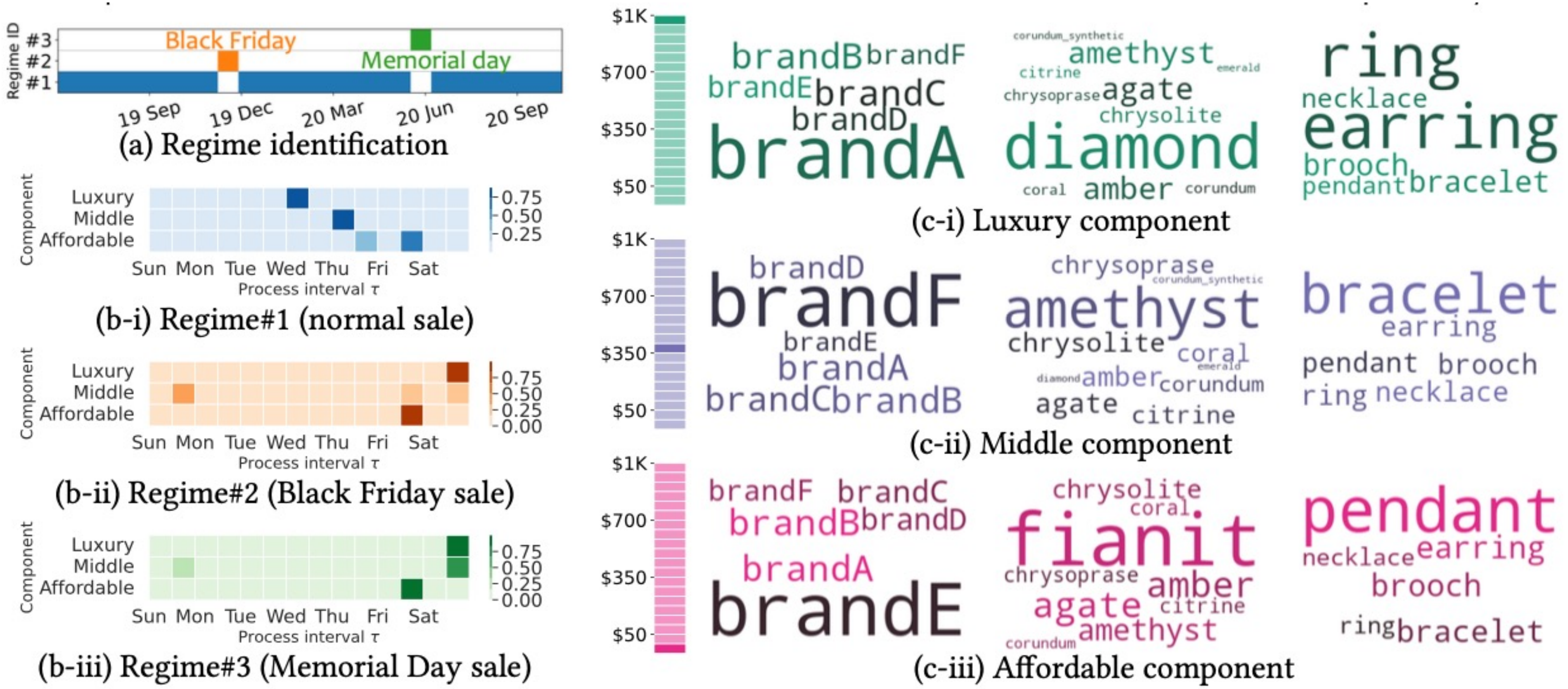
Unsupervised anomaly detection methods
- ❑ LOF
- ❑ iForest
- ❑ RRCF
- ❑ MemStream

SANKEN

Q1. Effectiveness: Online Marketing Analytics

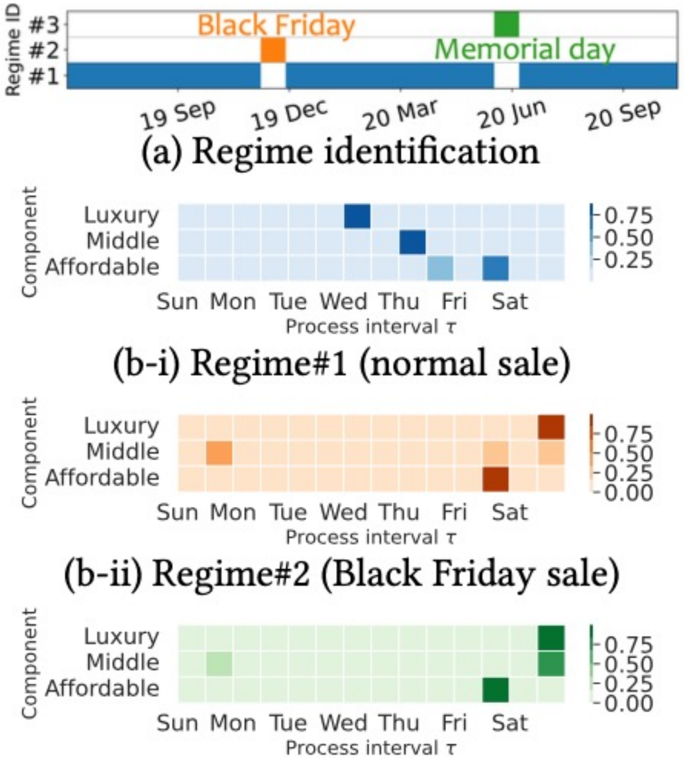
Jewerly Dataset: 4rd-order tensor stream

{Time, Price, Brand, Gem, Accessory type}



Q1. Effectiveness:

Jewerly Dataset: 4rd-order
{Time, Price, Brand, Gem,



(a) Regime identification
(b-i) Regime#1 (normal sale)
(b-ii) Regime#2 (Black Friday sale)
(b-iii) Regime#3 (Memorial Day sale)

Regimes:
Distinct dynamical patterns

Changes in Purchase behavior

Components:

multi-aspect latent trends

User preferences

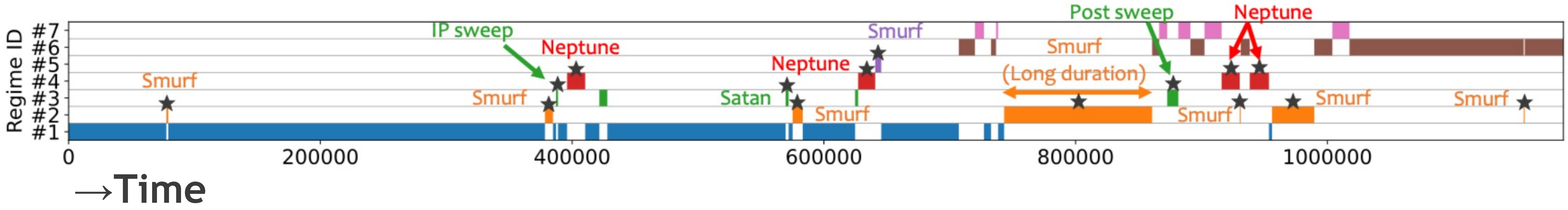
tensor stream

Accessory type}

Q1. Effectiveness: Cybersecurity

AirForce Dataset: 10th-order tensor stream

{Time,Protocol type, Service, Flag, Land, Duration, Src/Dst bytes, Wrong fragment, Urgent}



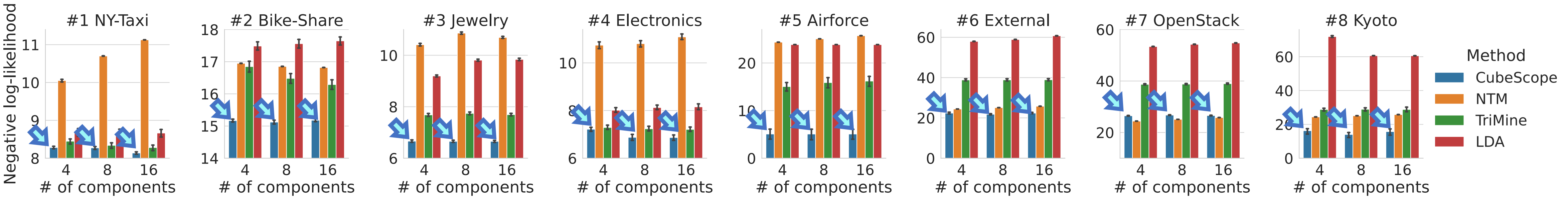
found **Regimes** that most corresponded to actual intrusions

- ❑ These intrusions arise over time and thus their numbers, durations, and features are unknown in advance

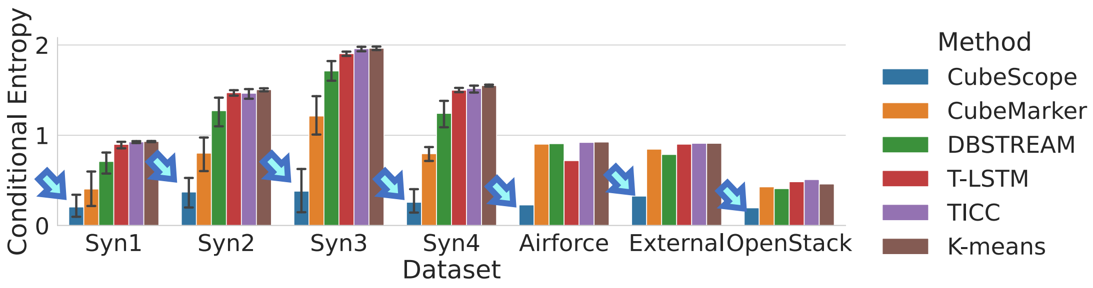
Q2. Accuracy: Modeling, Clustering, Anomaly Detection

"How does *CubeScope* achieve
modeling, **clustering**, and **anomaly detection**?"

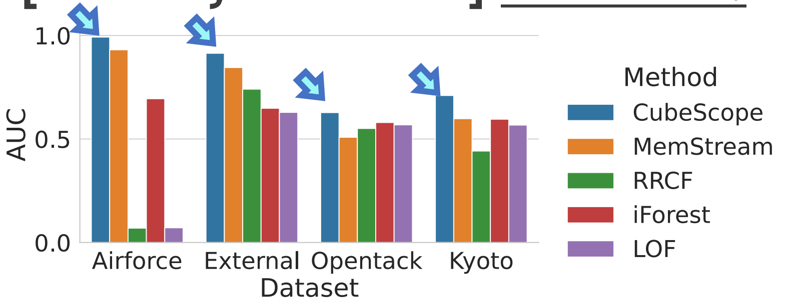
[Modeling] <u>Negative log-likelihood:</u> lower is better

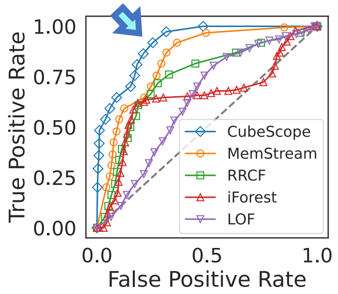


[Clustering] <u>CE score:</u> lower is better



[Anomaly Detection] <u>AUC score:</u> higher is better

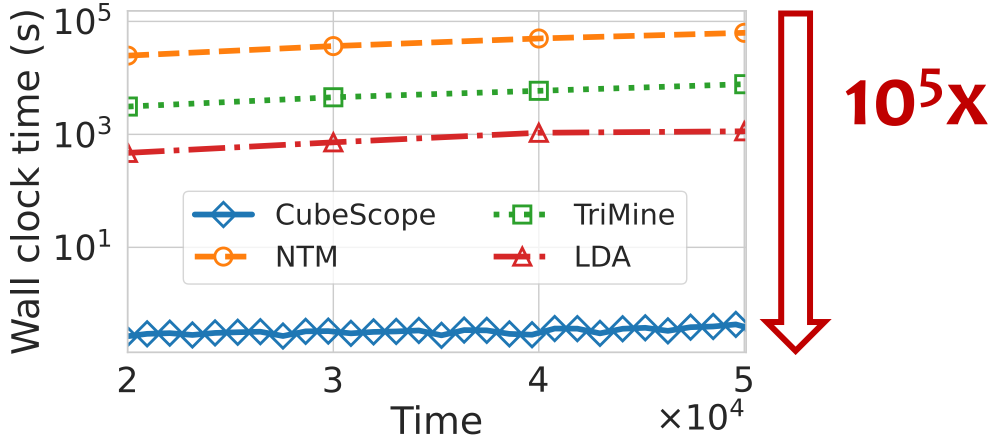


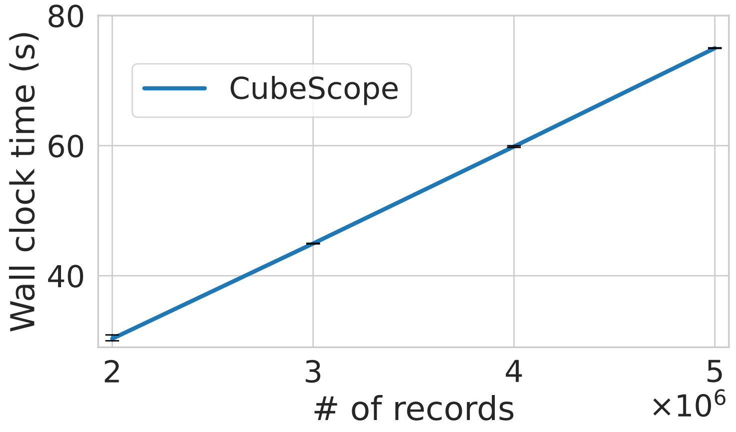


CubeScope **consistently outperforms** its baselines

Q3. Scalability

“How does CubeScope scale in terms of computational time?”





CubeScope is **up to 312,000x faster** than baselines
and **scales linearly**

Outline

Introduction

Model

Algorithm

Experiments

▶ Conclusion

Conclusion

Effective

- Introduce regimes and components
- Formulate the summarization problem for capturing these patterns
- Design *CubeScope* to solve the summarization problem

General

- Perform data compression, pattern discovery, and anomaly detection
- Practical in multiple domains,
  such as local mobility, online market analytics, and cybersecurity

Scalable

- Fast and constant computational time
  w.r.t. the entire stream length and its dimensionality

Thank you!

CubeScope

Data&Code: