Non-Linear Mining of Social Activities in Tensor Streams

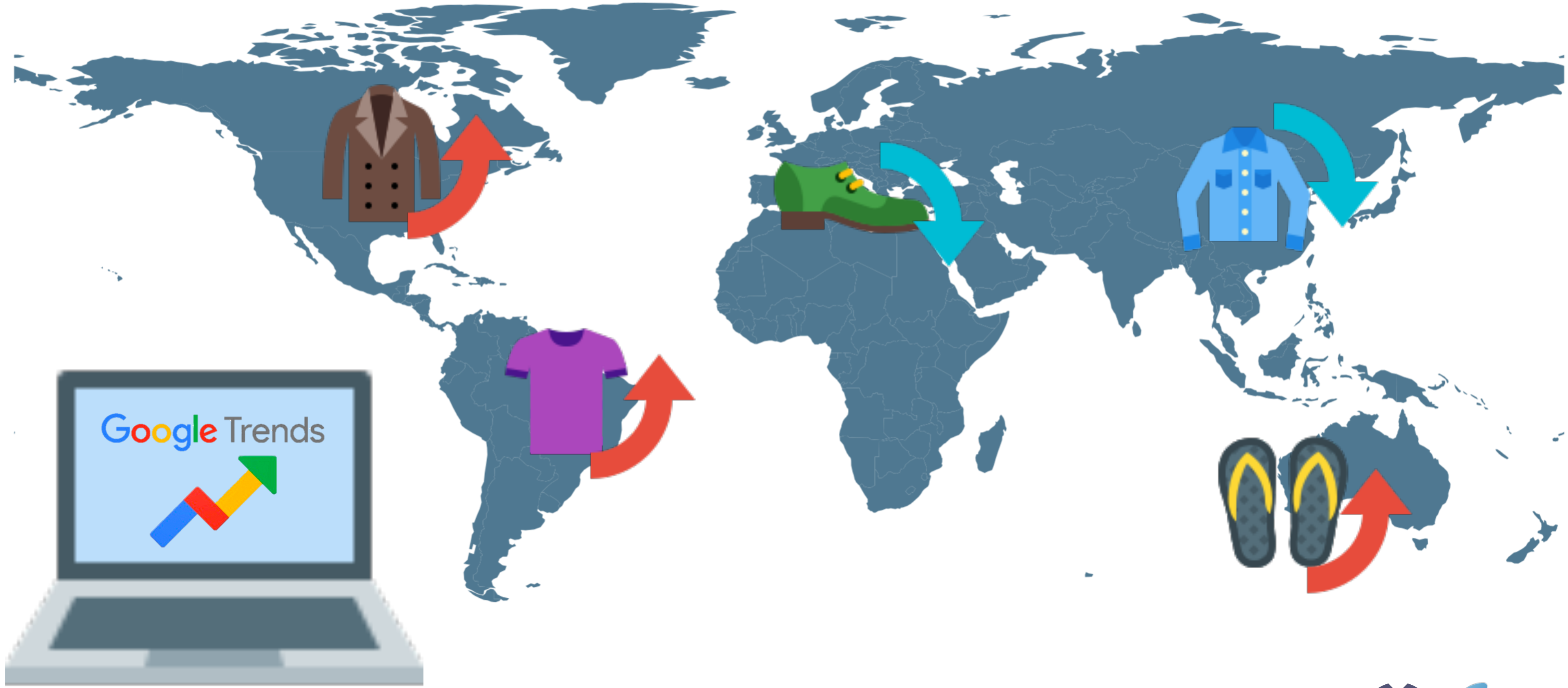
Koki Kawabata, Yasuko Matsubara, Takato Honda, Yasushi Sakurai

AIRC - ISIR, Osaka University

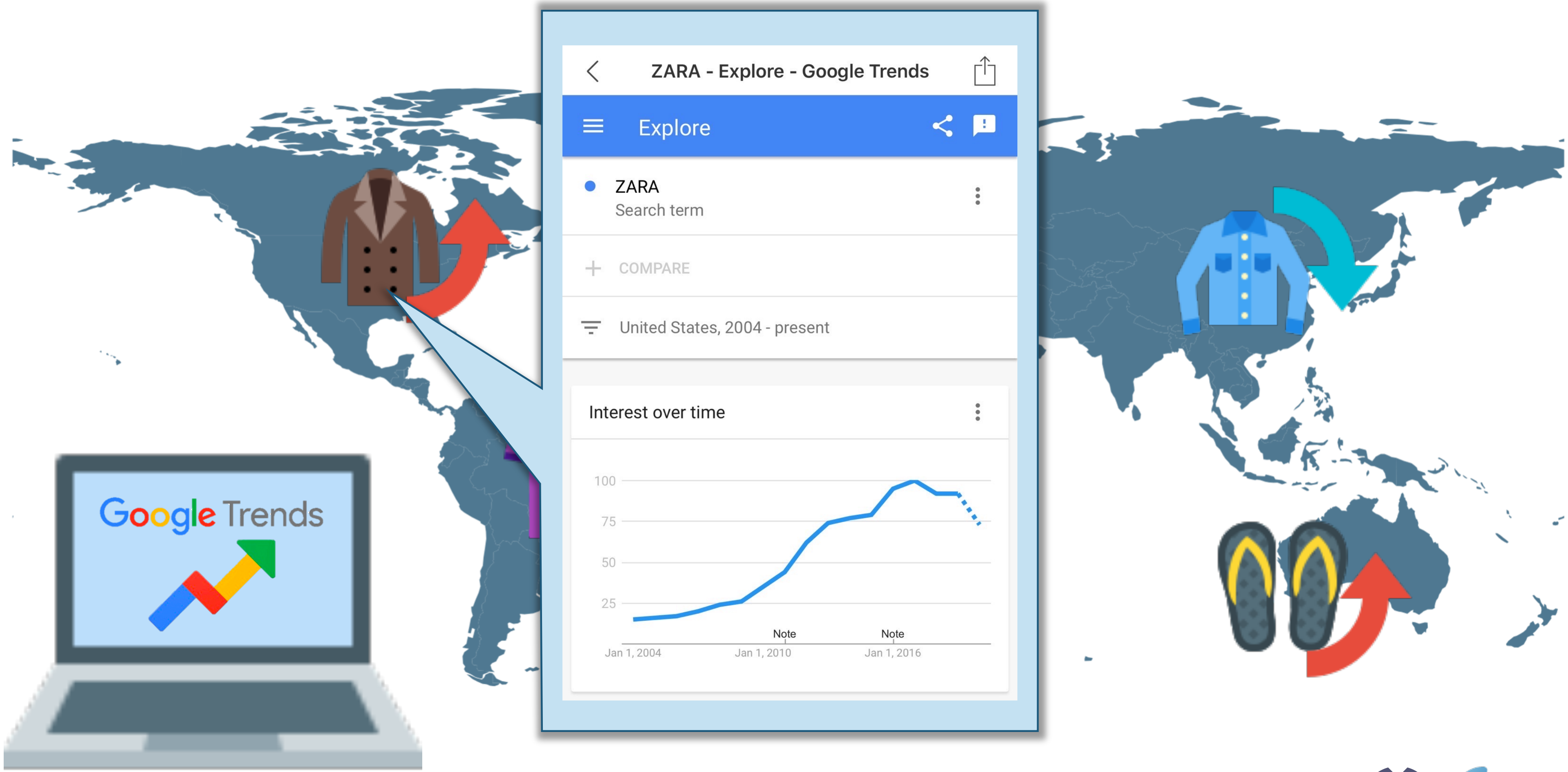
KDD

ISIR
OSAKA UNIVERSITY

Variety of Social Activities on the Web



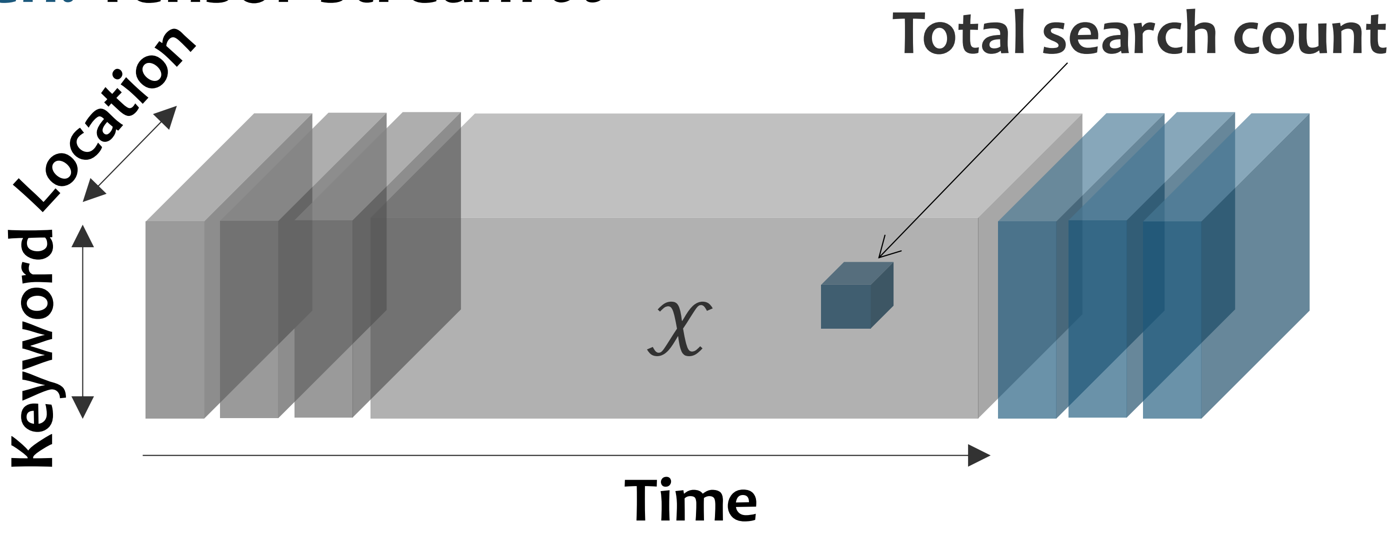
Variety of Social Activities on the Web



Variety of Social Activities on the Web

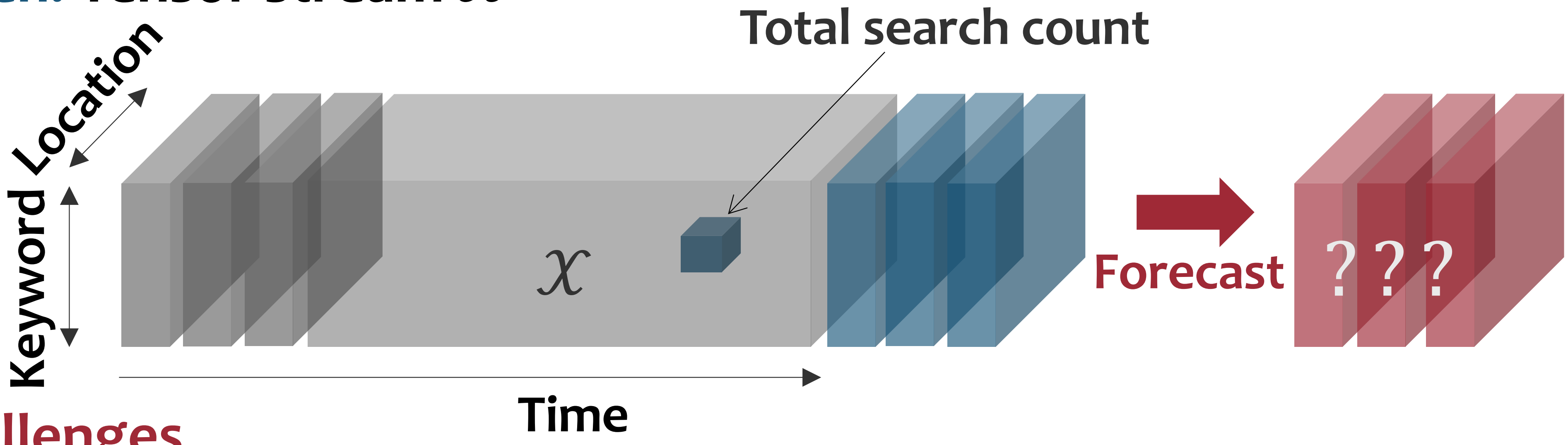
Motivation

Given: Tensor stream $\mathcal{X}$



Motivation

Given: **Tensor stream $\mathcal{X}$**

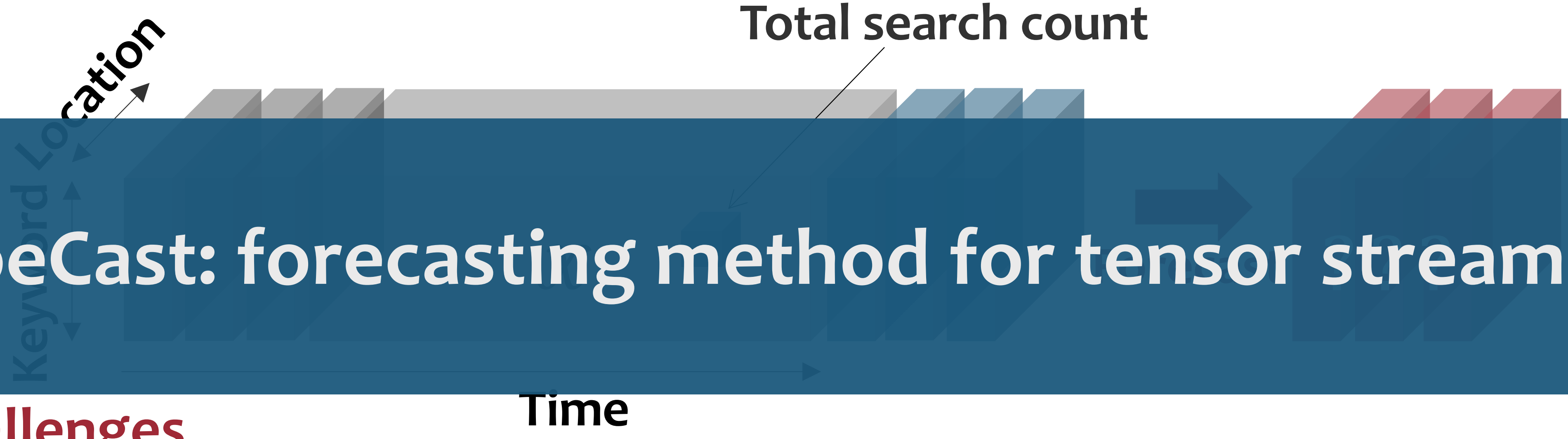


Challenges

- **Find** trends and seasonal patterns
- **Find** a set of groups of similar dynamics
- **Forecast** future values in a streaming fashion

Motivation

Given: **Tensor stream $\mathcal{X}$**



CubeCast: forecasting method for tensor streams

Challenges

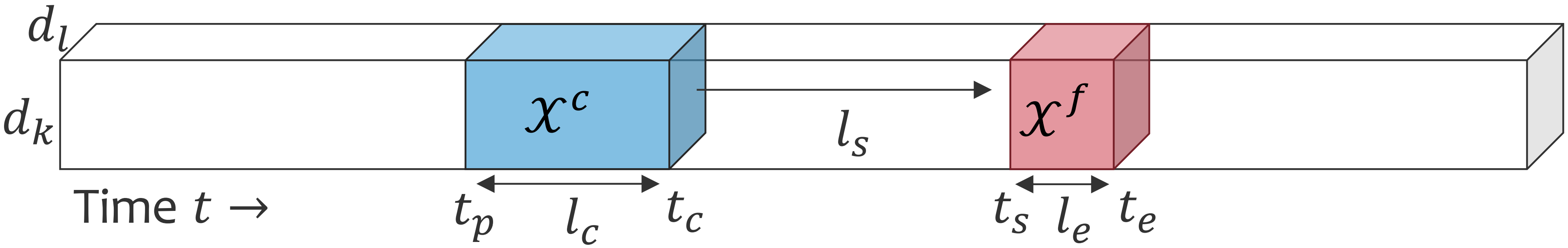
- **Find** trends and seasonal patterns
- **Find** a set of groups of similar dynamics
- **Forecast** future values in a streaming fashion

Outline

✓Motivation
- Problem definition
- Proposed model
- Proposed algorithm: CubeCast
- Experiments
- Conclusion

Problem definition

- **Given:** Tensor stream $\mathcal{X}$
- **Forecast:** l_s-step ahead values



$\mathcal{X}^c$	Current window
t_c	Current time point
d_l	Number of locations
d_k	Number of keywords

$\mathcal{X}^f$	Forecasting window
l_c	Size of current window
l_s	Size of time points to forecast
l_e	Size of report intervals

Outline

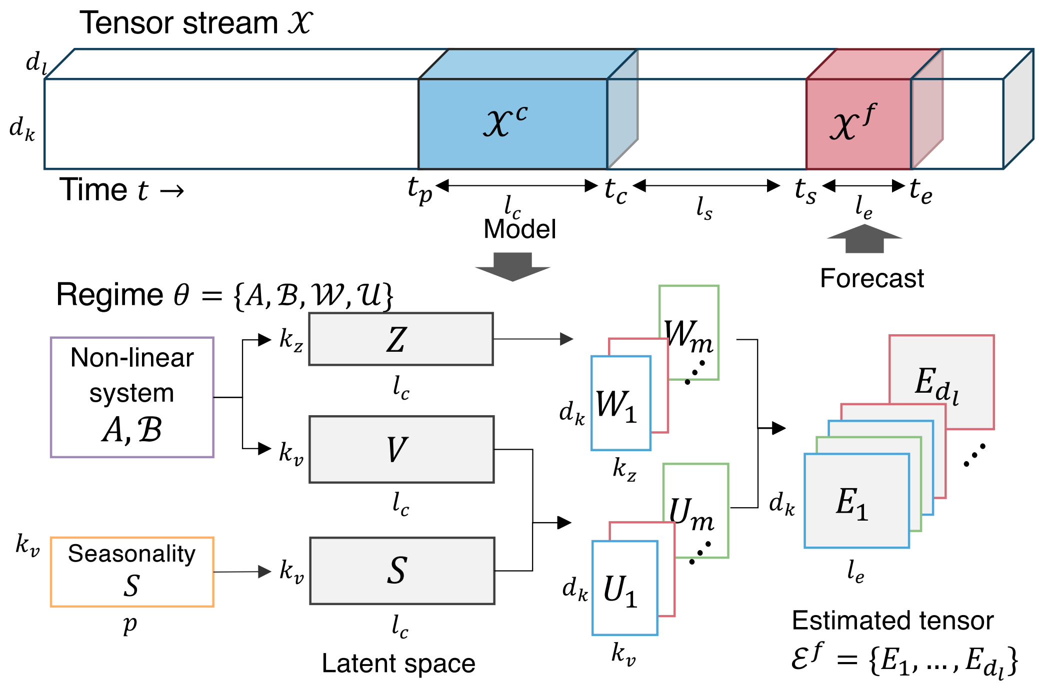
✓Motivation

✓Problem definition

- Proposed model
- Proposed algorithm: CubeCast
- Experiments
- Conclusion

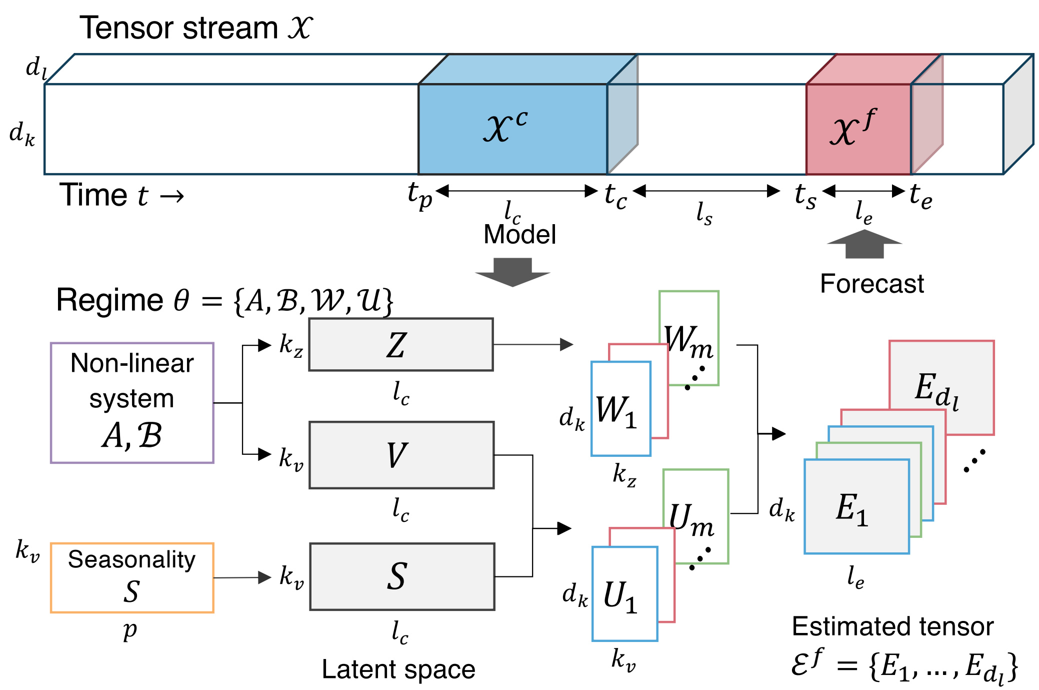
Proposed model

- **Overview**



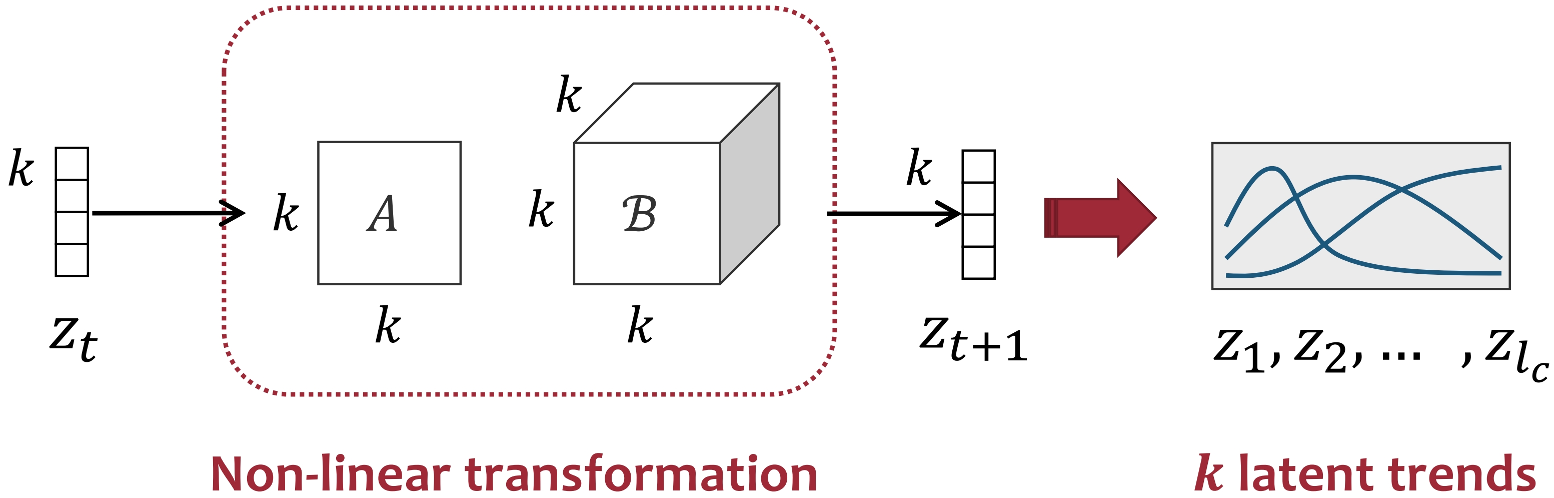
Proposed model

- <u>**Overview**</u>
  1. **Non-linear latent dynamics**
  2. **Seasonality**
  3. **Location-specific patterns**
  4. **Regime: dynamical changes**



1. Non-linear latent dynamics

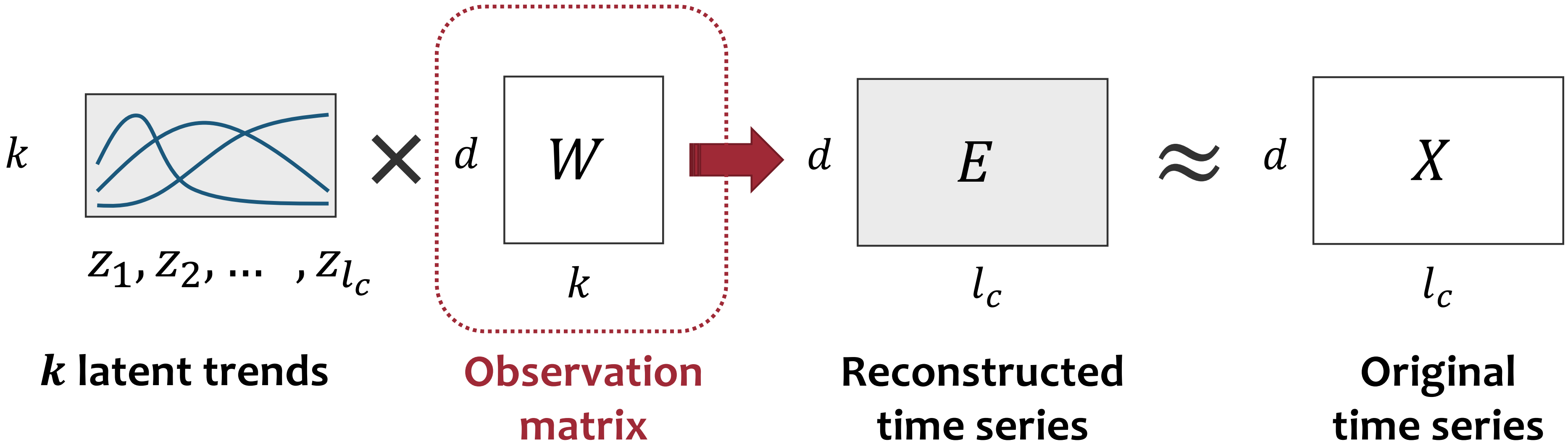
- **Non-linear dynamical system**



$$z_{t+1} = A z_t + \mathcal{B}\, z_t \otimes z_t$$

1. Non-linear latent dynamics

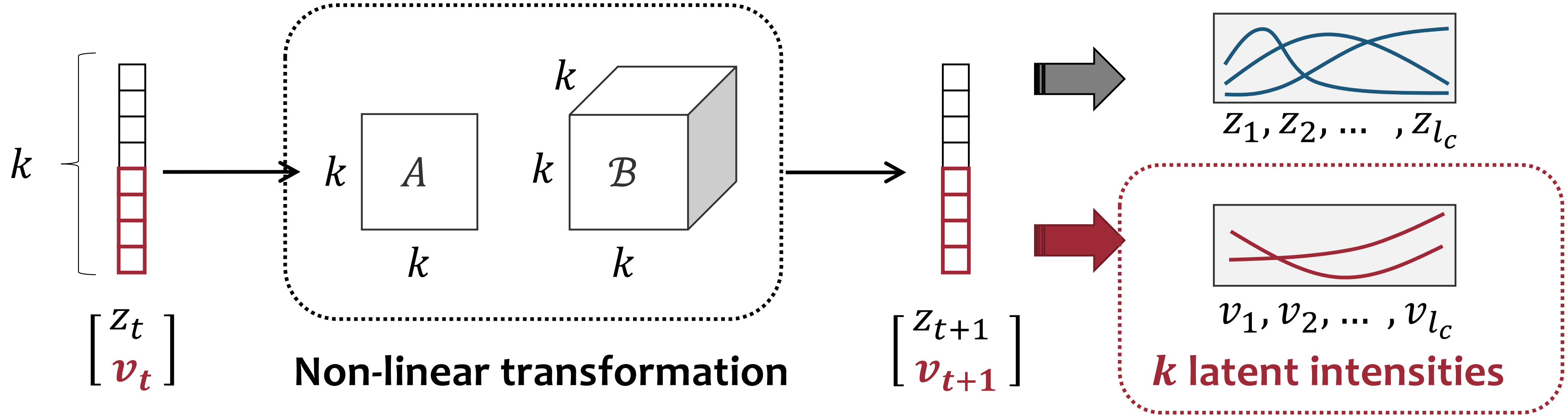
- **Non-linear dynamical system**



$$e_t = \boldsymbol{W} z_t$$

2. Seasonality

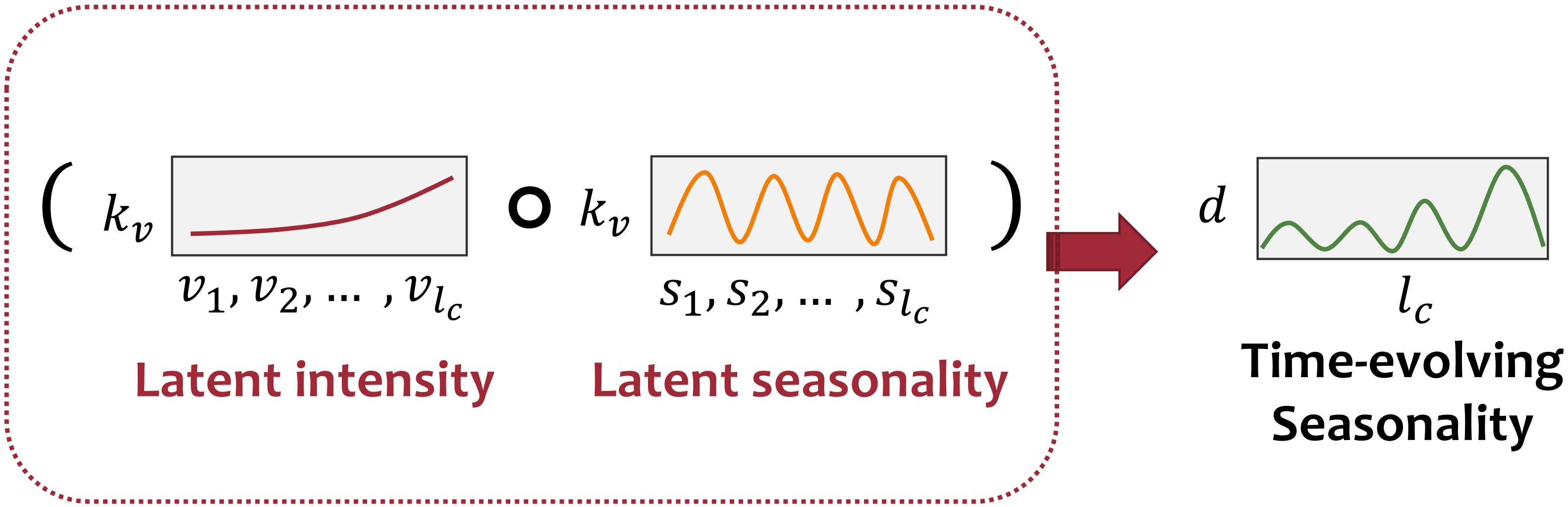
- **Non-linear dynamical system with seasonality**
  - Latent intensity, v_t



$$\begin{bmatrix} z_{t+1} \\ \boldsymbol{v_{t+1}} \end{bmatrix} = A \begin{bmatrix} z_t \\ \boldsymbol{v_t} \end{bmatrix} + \mathcal{B} \begin{bmatrix} z_t \\ \boldsymbol{v_t} \end{bmatrix} \otimes \begin{bmatrix} z_t \\ \boldsymbol{v_t} \end{bmatrix}$$

2. Seasonality

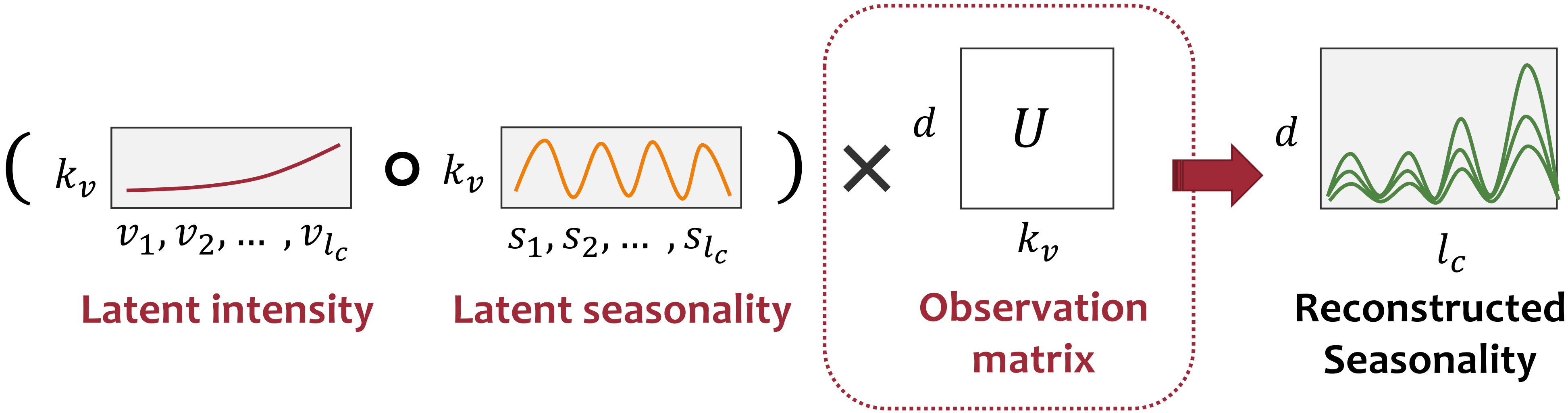
- **Non-linear dynamical system with seasonality**
  - Latent seasonality: S



$$e_t = Wz_t + \boldsymbol{U}\left(\boldsymbol{v}_t \circ \boldsymbol{S}_{(t \bmod p)}\right)$$

2. Seasonality

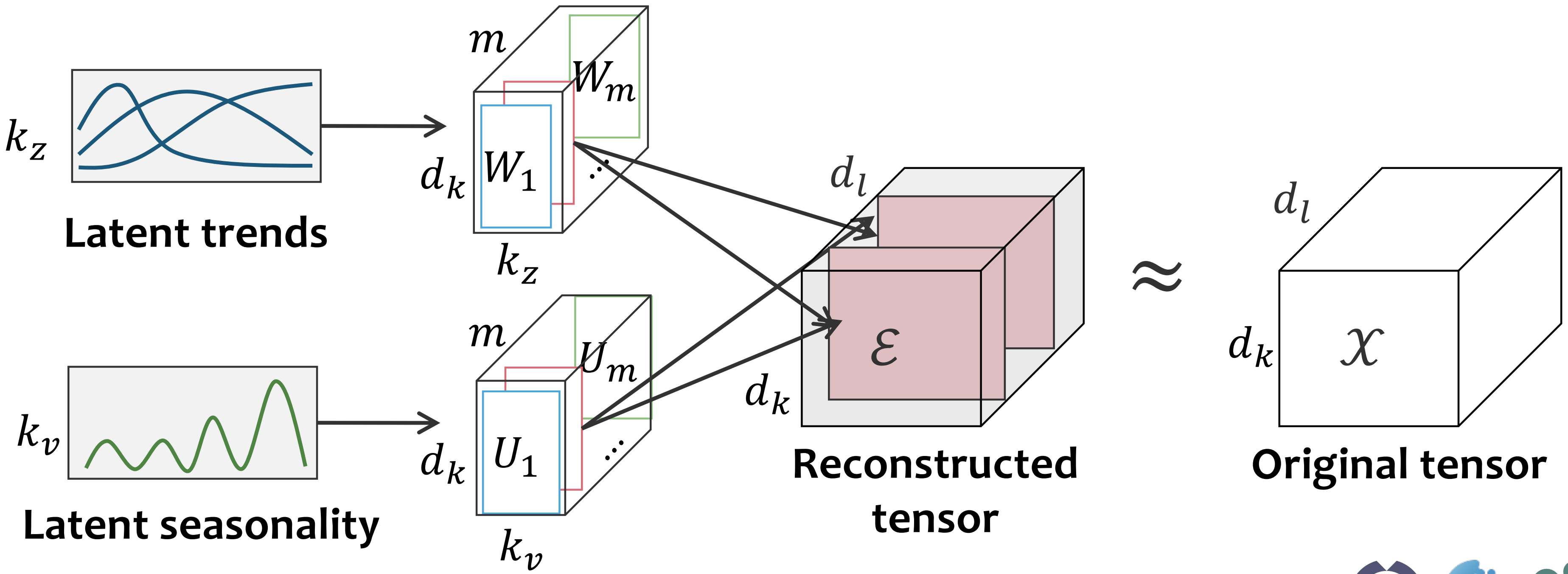
- **Non-linear dynamical system with seasonality**
  - Observation matrix: U



$$e_t = Wz_t + \boldsymbol{U}(\boldsymbol{v}_t \circ \boldsymbol{S}_{(t \bmod p)})$$

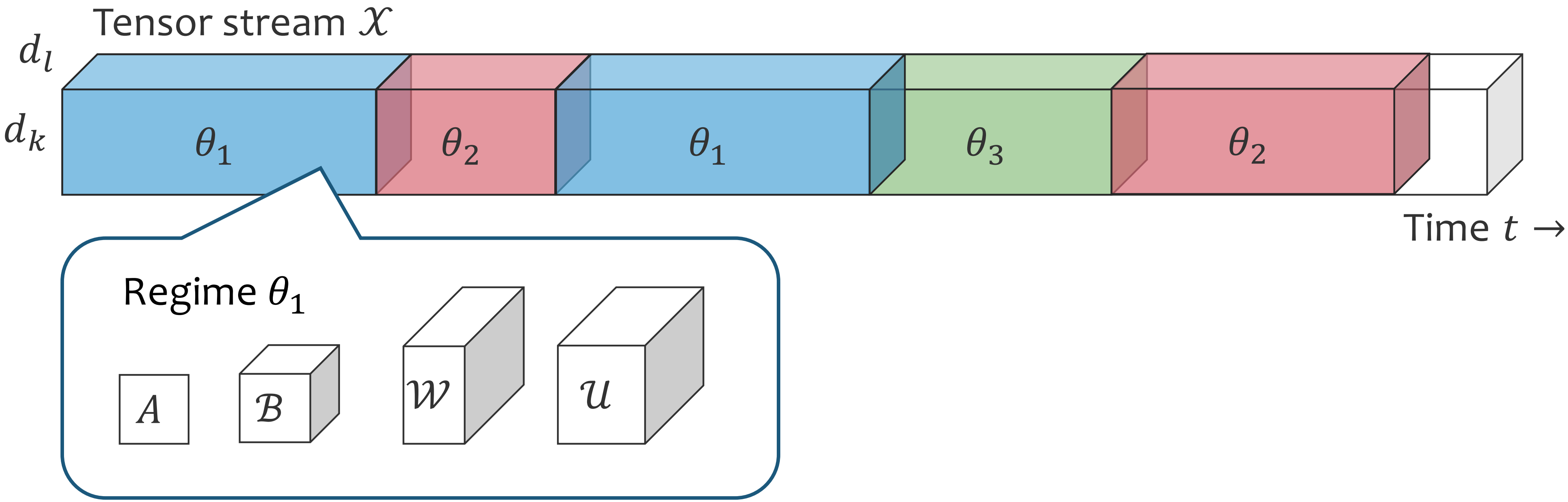
3. Location-specific patterns

- **Local groups in non-linear dynamical system**
  - m: number of local groups
  - Observation matrix set $\mathcal{W} = \{W_1, \cdots, W_m\}$, $\mathcal{U} = \{U_1, \cdots, U_m\}$



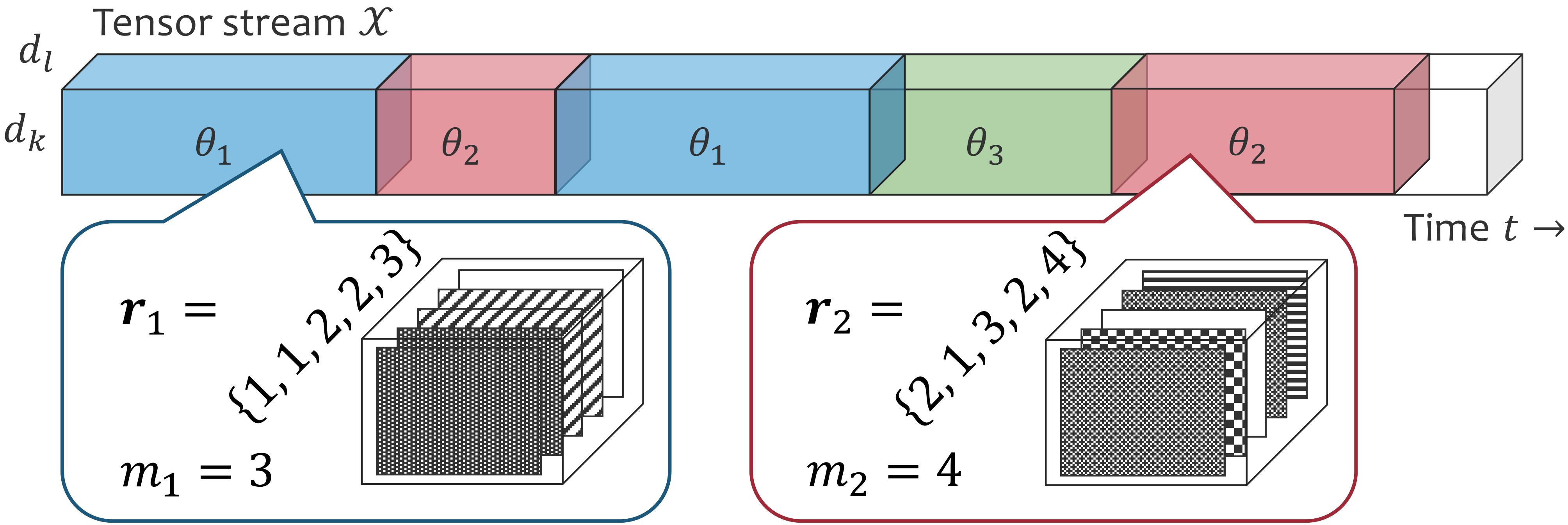
4. Dynamical changes of patterns

- **Regime: dynamical changes of non-linear dynamical systems**
  - Regime parameter set $\Theta = \{\theta_1, \cdots, \theta_n, S\}$, $\theta_i = \{A, \mathcal{B}, \mathcal{W}, \mathcal{U}\}$



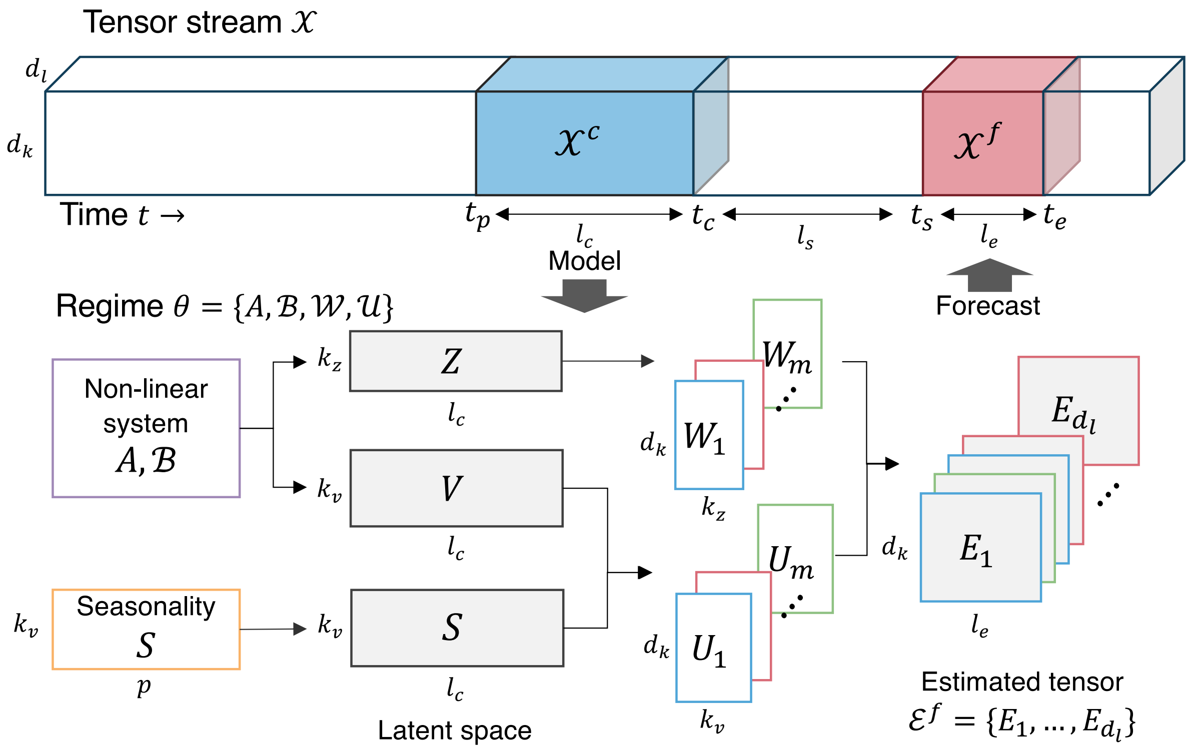
4. Dynamical changes of patterns

- **Regime: dynamical changes of non-linear dynamical systems**
  - Regime parameter set $\Theta = \{\theta_1, \cdots, \theta_n, S\}$, $\theta_i = \{\mathcal{A}, \mathcal{B}, \mathcal{W}, \mathcal{U}\}$
  - Regime assignment set $\mathcal{R} = \{\boldsymbol{r}_1, \cdots, \boldsymbol{r}_n\}$



Proposed model

- **CubeCast parameters**
  - **Regime parameter set:**
    - $\Theta = \{\theta_1, \cdots, \theta_n, S\}$
    - $\theta_i = \{\mathcal{A}, \mathcal{B}, \mathcal{W}, \mathcal{U}\}$
  - **Regime assignment set:**
    - $\mathcal{R} = \{\boldsymbol{r}_1, \cdots, \boldsymbol{r}_n\}$
    - $\boldsymbol{r}_i = \{r_1, \cdots, r_j, \cdots, r_{d_l}\}$
    - $r_j \in \{1, \cdots, m_i\}$



Outline

✓Motivation
✓Problem definition
✓Proposed model
- **Proposed algorithm: CubeCast**
- Experiments
- Conclusion

Proposed algorithm: CubeCast

Goal: estimate:

- Regime parameter set Θ
- Regime assignment set $\mathcal{R}$

Q1. How automatically can we find all components in CubeCast?

Q2. How effectively can we estimate non-linear parameters?

Q3. How efficiently can we find local groups and their switching?

Proposed algorithm: CubeCast

Goal: estimate:

- Regime parameter set Θ
- Regime assignment set $\mathcal{R}$

Q1. How automatically can we find all components in CubeCast?

◆**Applying data encoding scheme**

Q2. How effectively can we estimate non-linear parameters?

◆**Alternative optimization approach**

Q3. How efficiently can we find local groups and their switching?

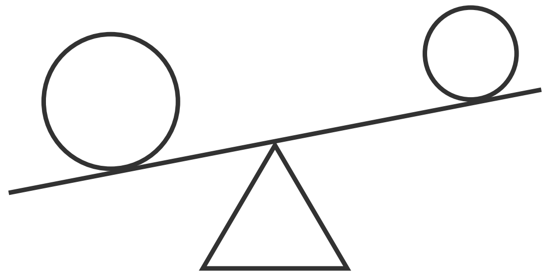
◆**Greedy approach to decide # of local groups and regimes**

Q1. What is a good summarization?

- **A. Minimum description length (MDL) principle**

$$\min\left(\boxed{<\Theta>} + \boxed{<\mathcal{X}|\Theta>} \right)$$

Model cost   Coding cost



Good compression ⟲ **Good description**

Q1. What is a good summarization?

- **Objective function**

$$< \mathcal{X}; \Theta > = < \Theta > + < \mathcal{X}|\Theta >$$

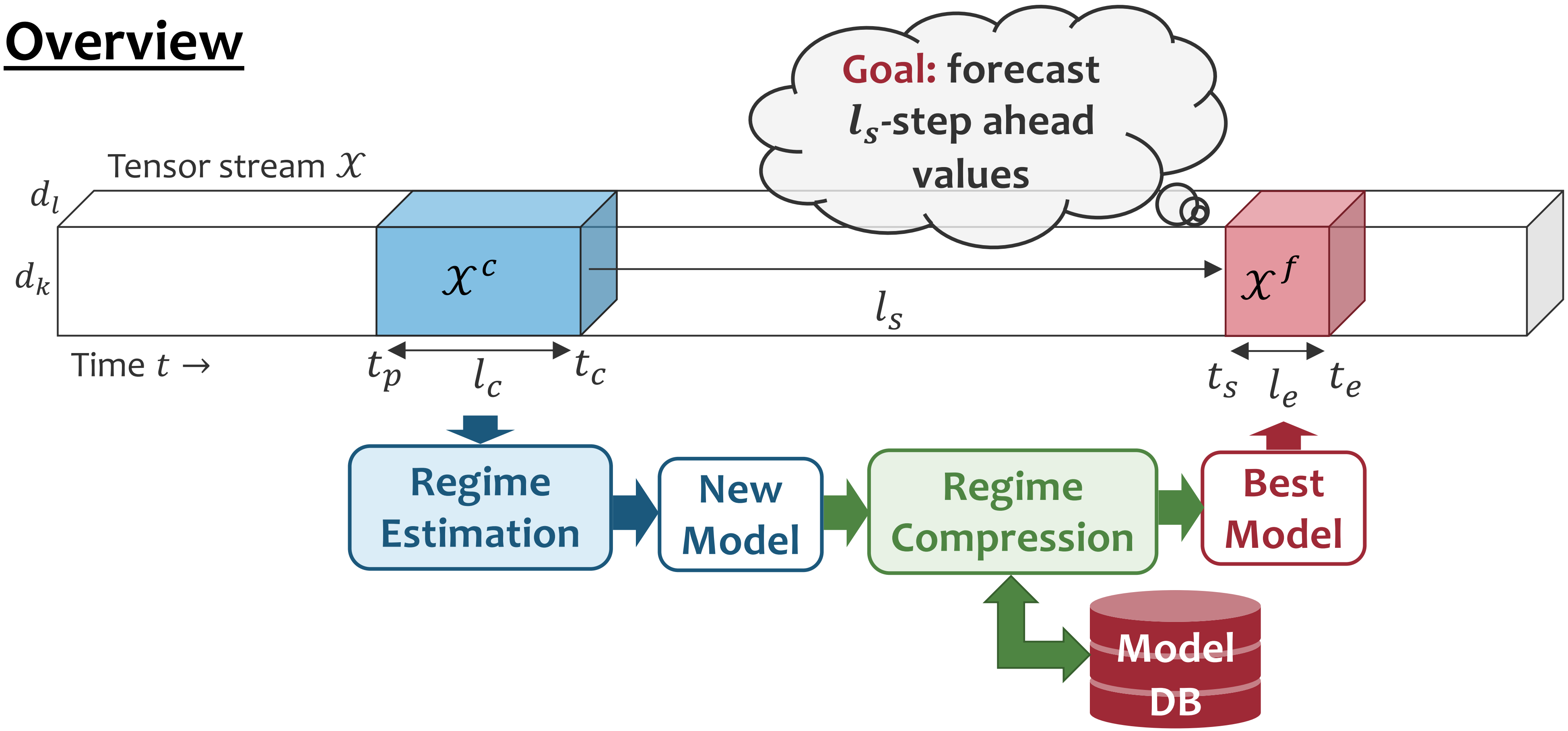
$$= < t_c > + < d_l > + < d_k > + < p >$$

$$+ < k_v > + < S > + \sum_{i=1}^{n} < \theta_i > + < \mathcal{X}|\Theta >$$

Details in paper

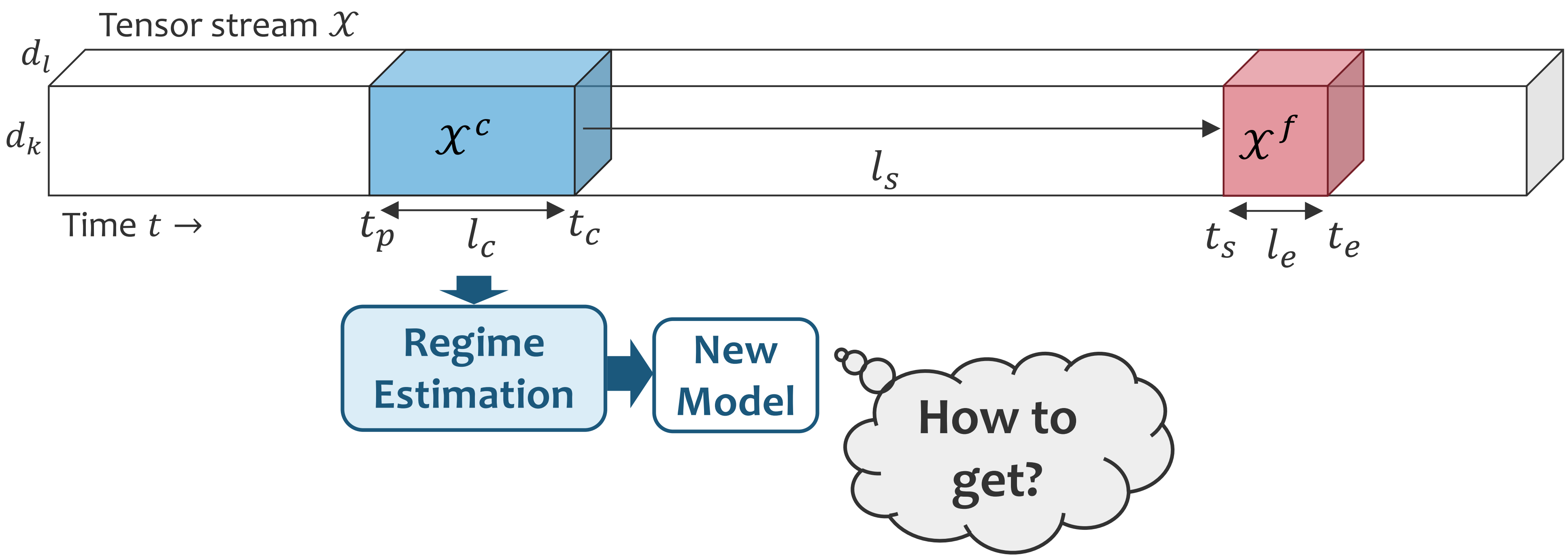
Proposed algorithm: CubeCast

- **Overview**



Proposed algorithm: CubeCast

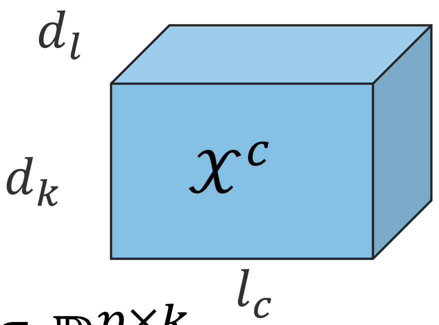
- **Overview**



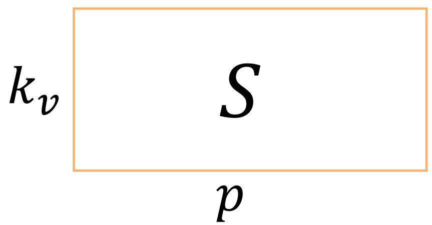
RegimeEstimation

Given:

- Current tensor $\mathcal{X}^c \in \mathbb{R}^{l_c \times d_l \times d_k}$

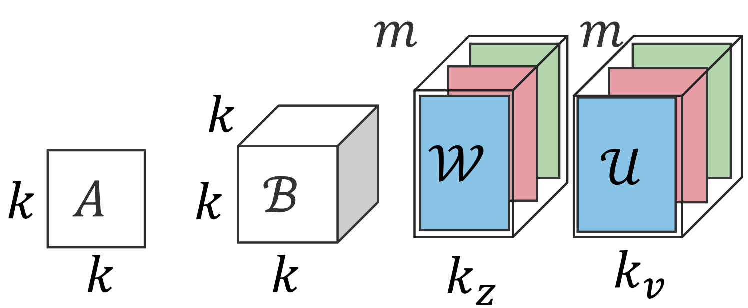


- Seasonality $S \in \mathbb{R}^{p \times k_v}$

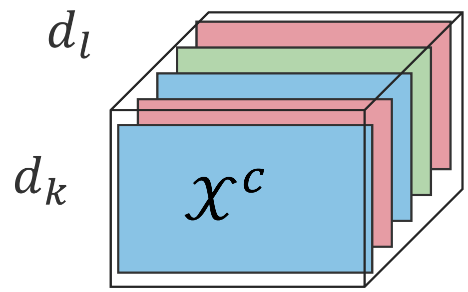


Find:

- Regime $\theta = \{A, \mathcal{B}, \mathcal{W}, \mathcal{U}\}$

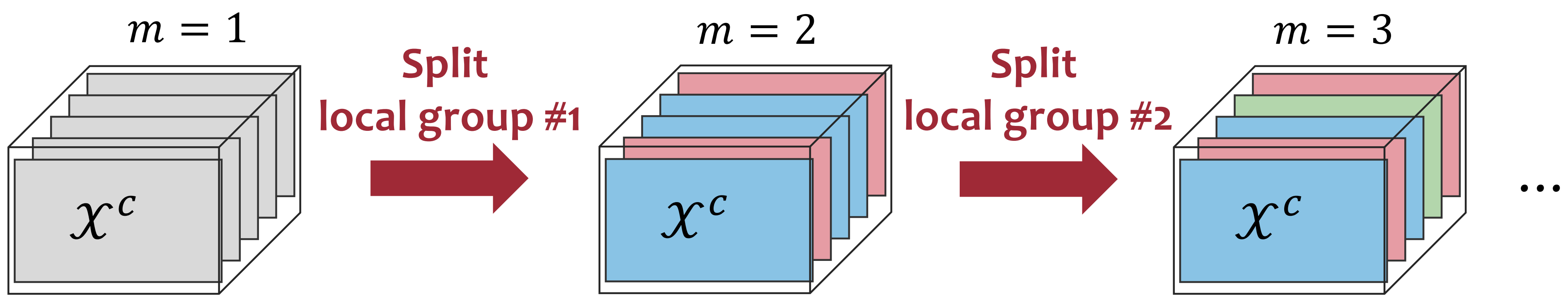


- Regime assignment $\boldsymbol{r} = \{r_1, \cdots, r_{d_l}\}$



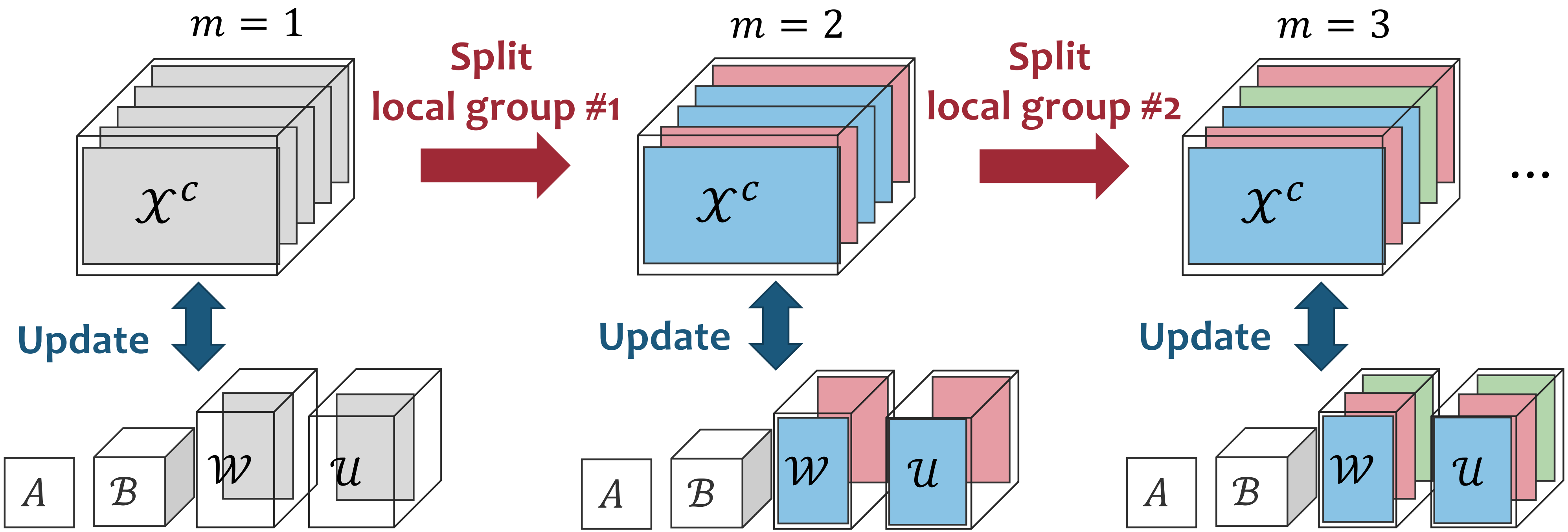
RegimeEstimation

- **<u>Main loop:</u>** Greedy approach to decide # of local group m
  - **minimize MDL score**



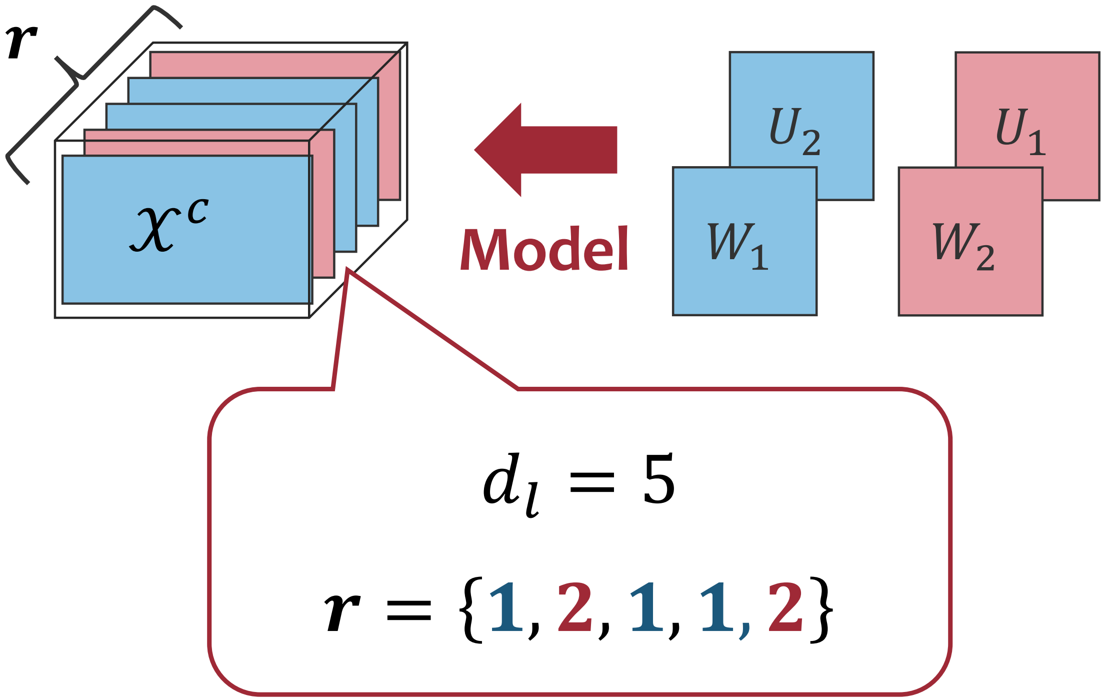
RegimeEstimation

- **<u>Main loop:</u>** Greedy approach to decide # of local group m
  - **minimize MDL score**



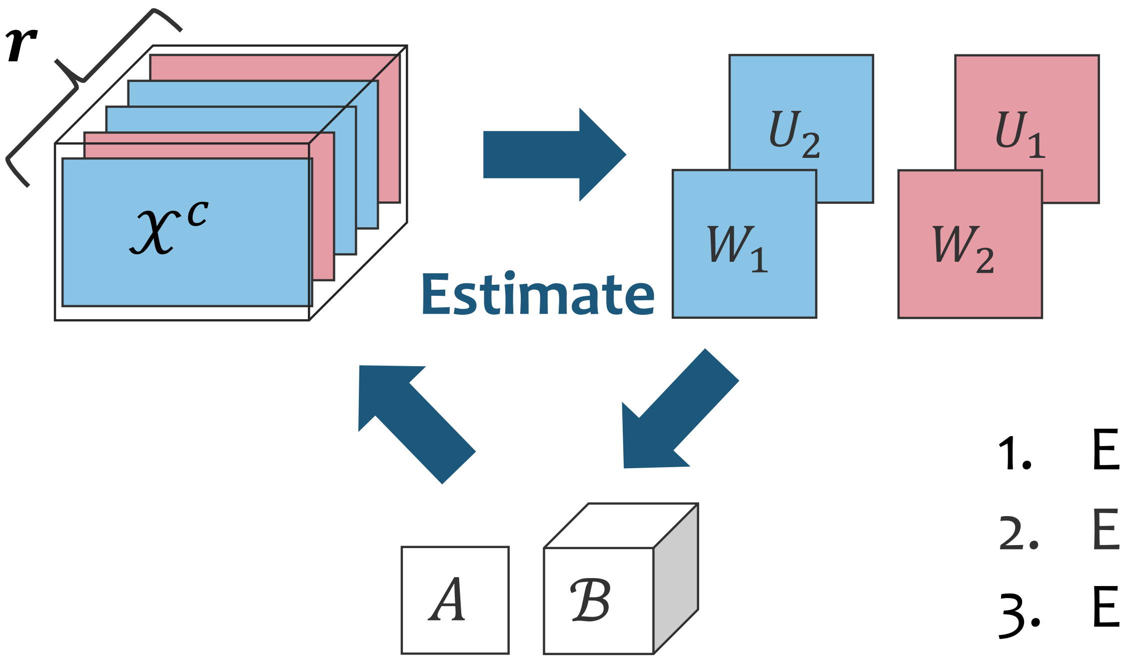
RegimeEstimation

- **<u>Sub loop:</u>** Estimate regime parameter θ
  - **minimize reconstruction error**



RegimeEstimation

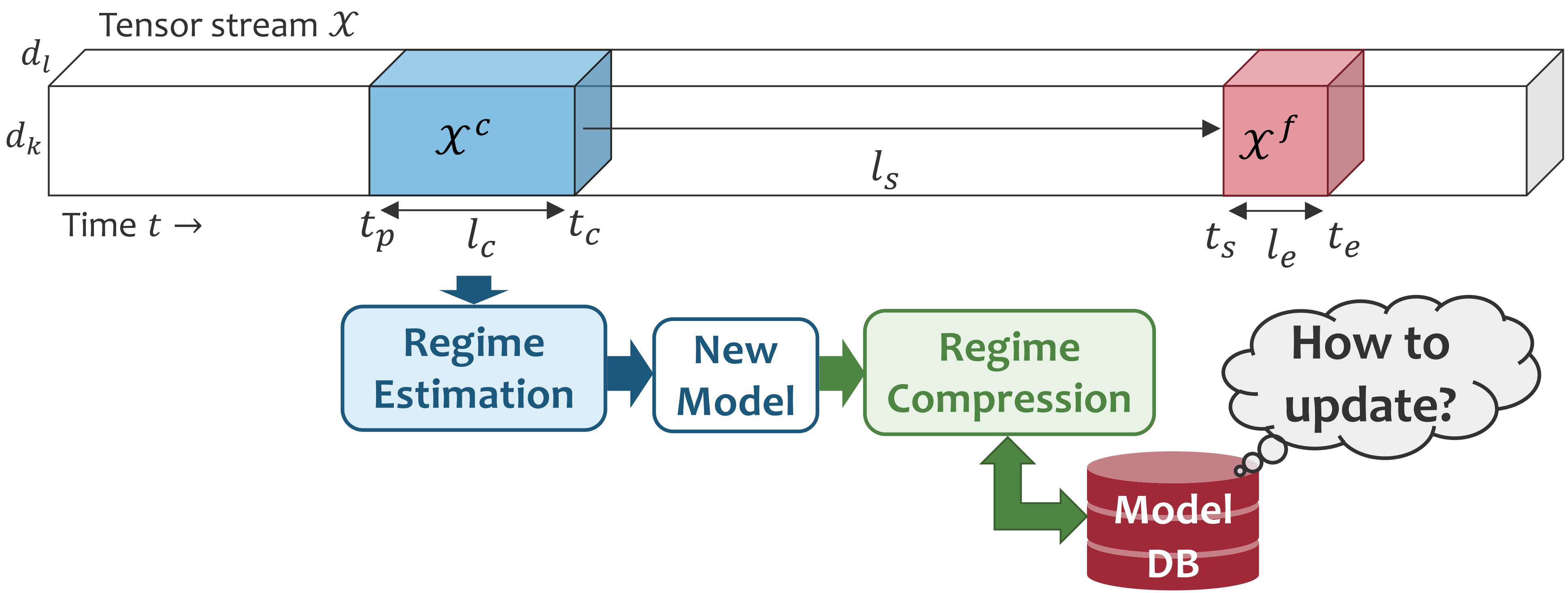
- **<u>Sub loop:</u>** Estimate regime parameter θ
  - **minimize reconstruction error**



1. Estimate $\boldsymbol{r} = \{r_1, \cdots, r_{d_l}\}$
2. Estimate W_1, U_1 and W_2, U_2
3. Estimate $A, \mathcal{B}$

Proposed algorithm: CubeCast

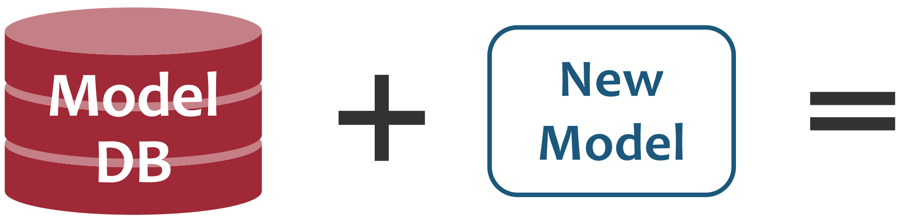
- **Overview**



RegimeCompression

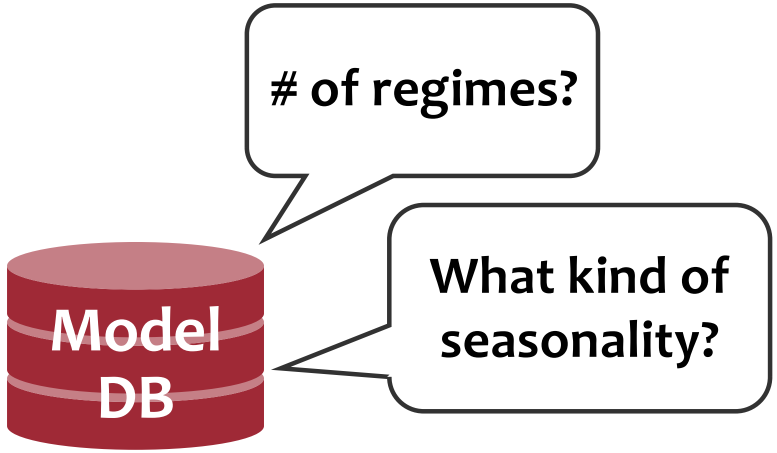
Given:

- Current tensor $\mathcal{X}^c \in \mathbb{R}^{l_c \times d_l \times d_k}$
- Full parameter set Θ and $\mathcal{R}$
- Candidate parameters θ and $\boldsymbol{r}$



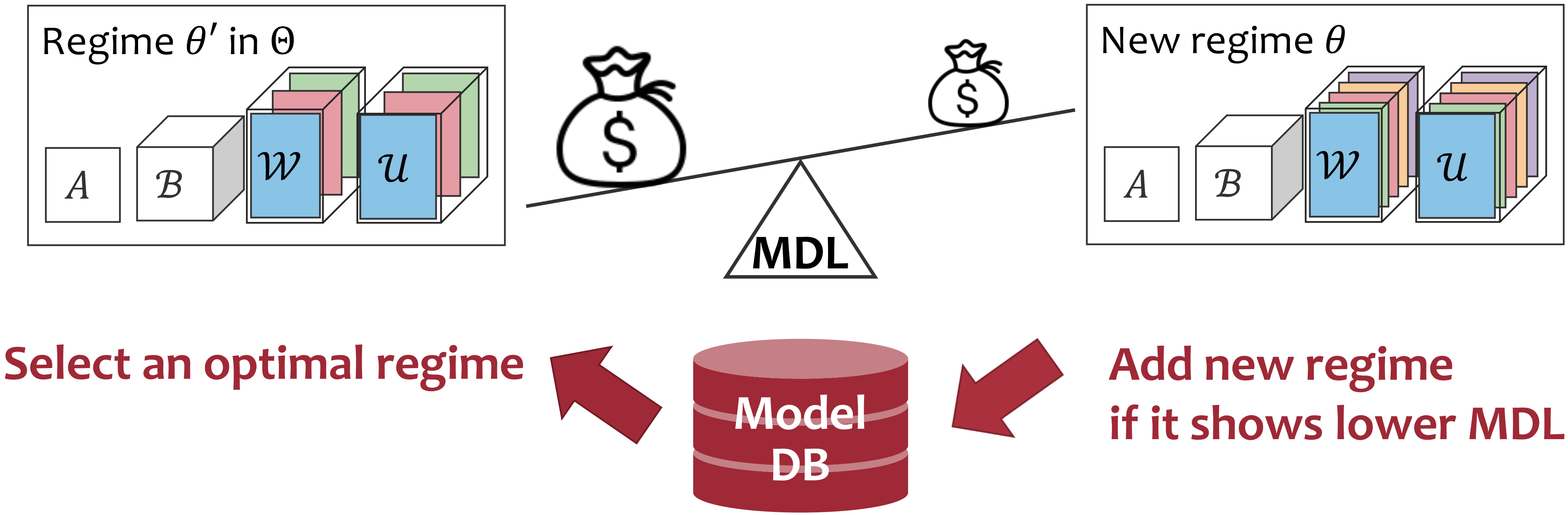
Find:

- Updated parameter set Θ' and $\mathcal{R}'$



RegimeCompression

- Decide whether or not to employ new regime θ

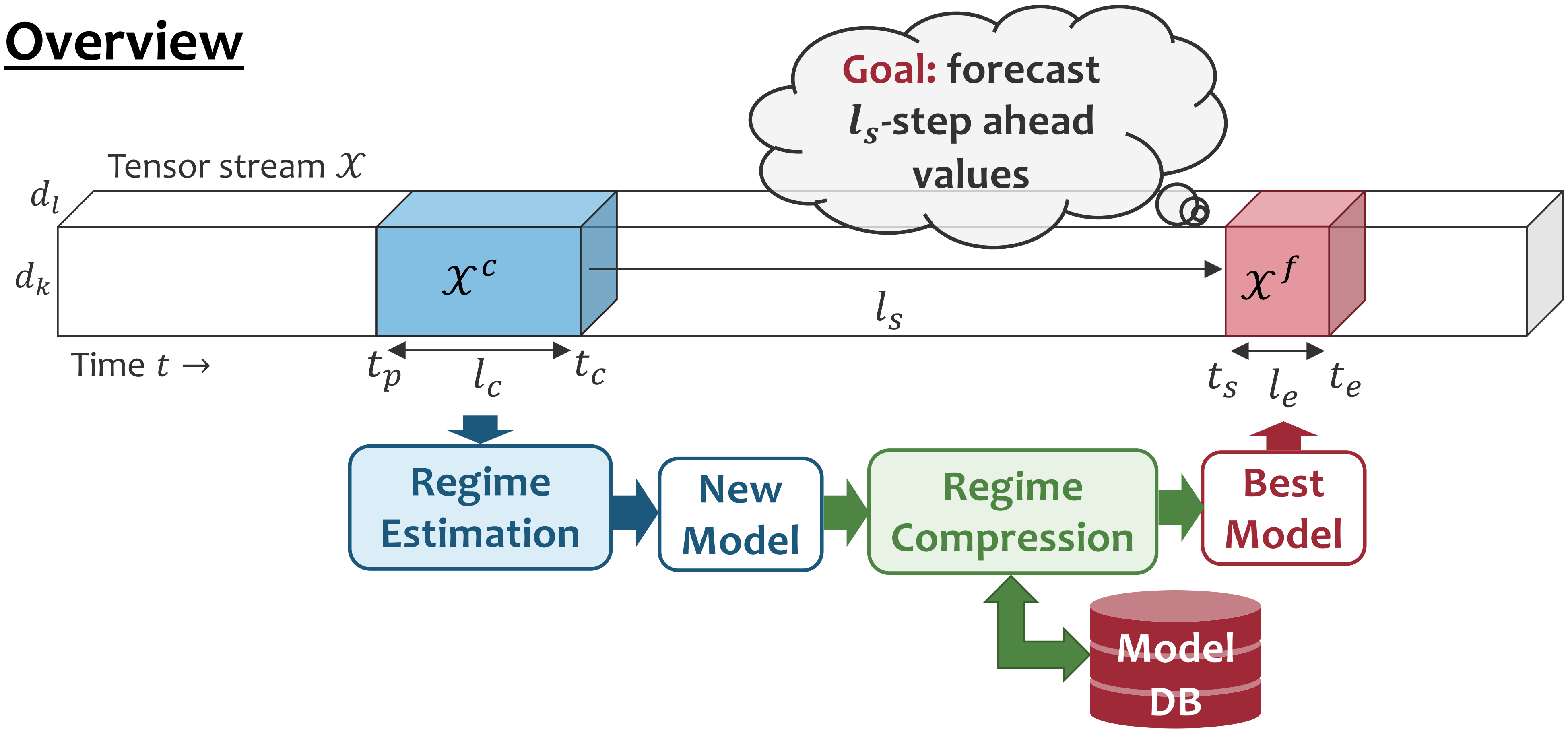


RegimeCompression

- Estimate seasonality with the best regime alternatively

Proposed algorithm: CubeCast

- **Overview**



Outline

✓Motivation
✓Problem definition
✓Proposed model
✓Proposed algorithm: CubeCast
- Experiments
- Conclusion

Experiments

- **Q1. Effectiveness**
  - How well does our method extract latent dynamical patterns?
- **Q2. Accuracy**
  - How accurately does our method predict future values?
- **Q3. Scalability**
  - How does our method scale in terms of computational time?

Experiments

- **Datasets:** GoogleTrends
  - Weekly aggregated search counts
  - 14 years (2004 – 2018)
  - 50 countries
  - 6 keyword sets

#1 Apparel
- ZARA
- UNIQLO
- H&M
- GAP
- Primark

#2 ChatApps
- Facebook
- LINE
- Slack
- Snapchat
- Twitter
- Telegram
- Viber
- Whatsapp

#3 Hobby
- Soccer
- Baseball
- Basketball
- Running
- Yoga
- Crafts

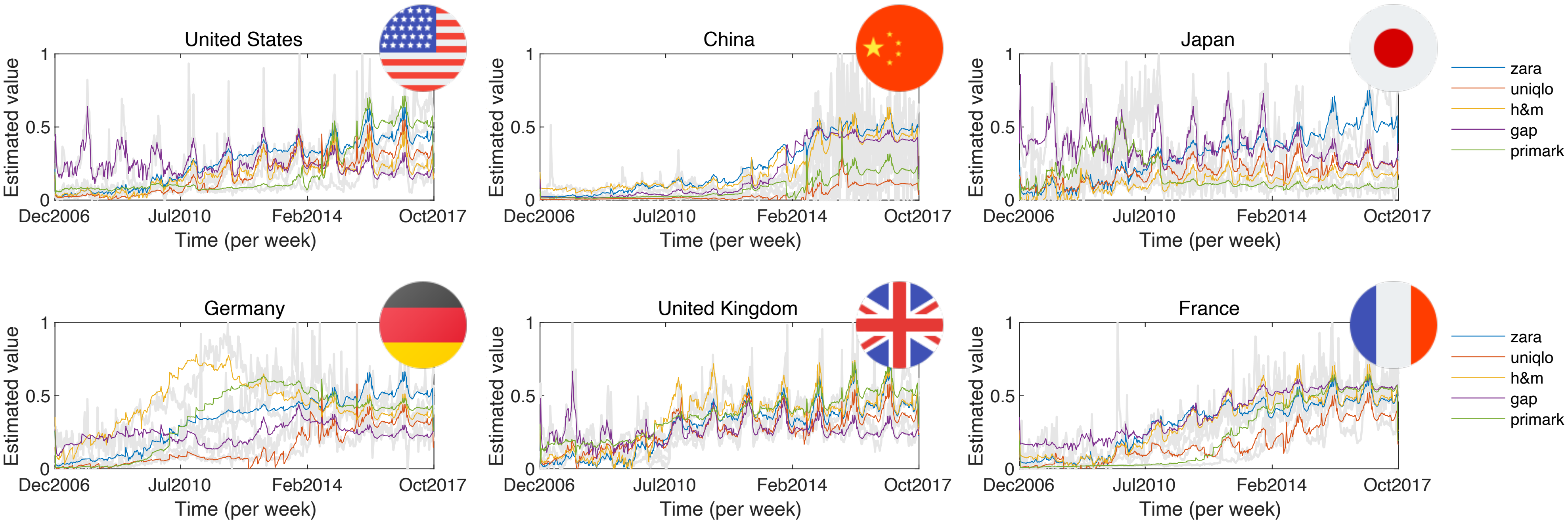
#4 LinuxOS
- Debian
- Ubuntu
- CentOS
- Redhat
- Fedora
- OpenSUSE
- SteamOS
- Raspbian
- Kubuntu

#5 PythonLib
- Numpy
- Scipy
- Sklearn
- Matplotlib
- Plotly
- TensorFlow

#6 Shoes
- Booties
- Flats
- Heels
- Loafers
- Pumps
- Sandals
- Sneakers

Q1. Effectiveness

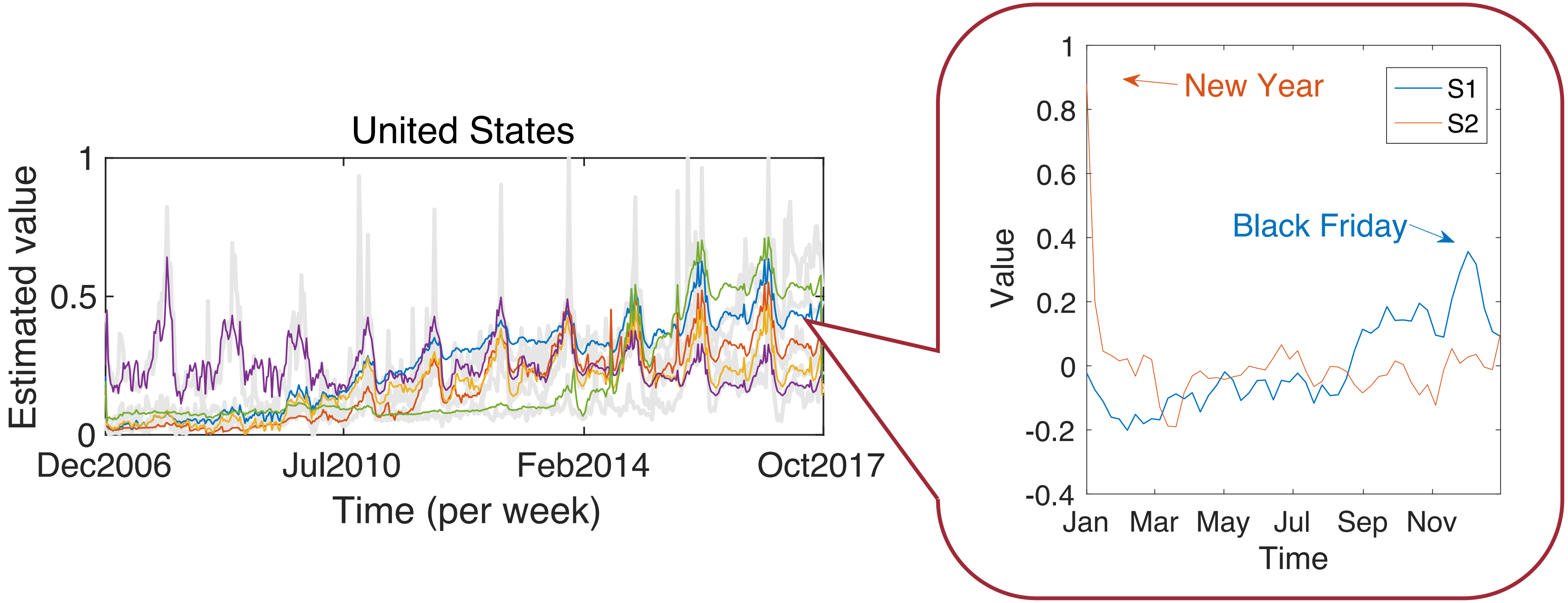
- Apparel company dataset: **Overall fitting results**



CubeCast can model non-linear dynamics with seasonality

Q1. Effectiveness

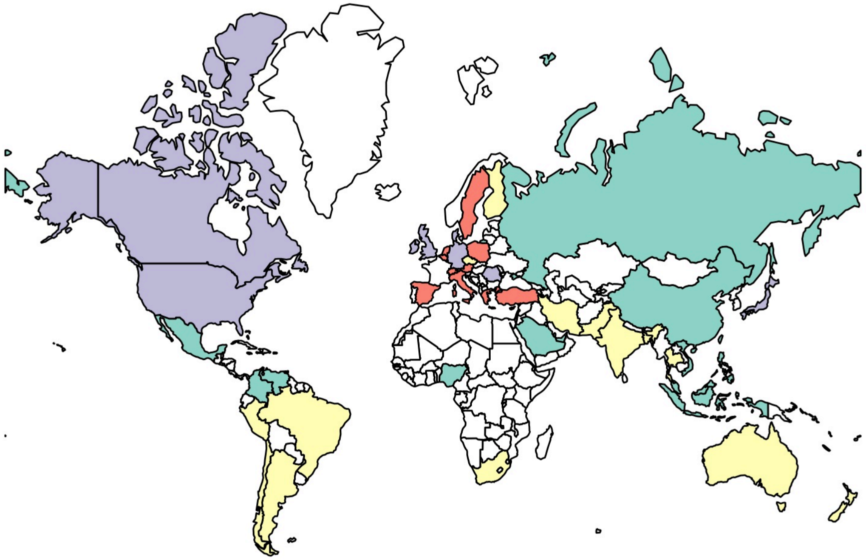
- Apparel company dataset: **Seasonality**



CubeCast can extract important seasonality

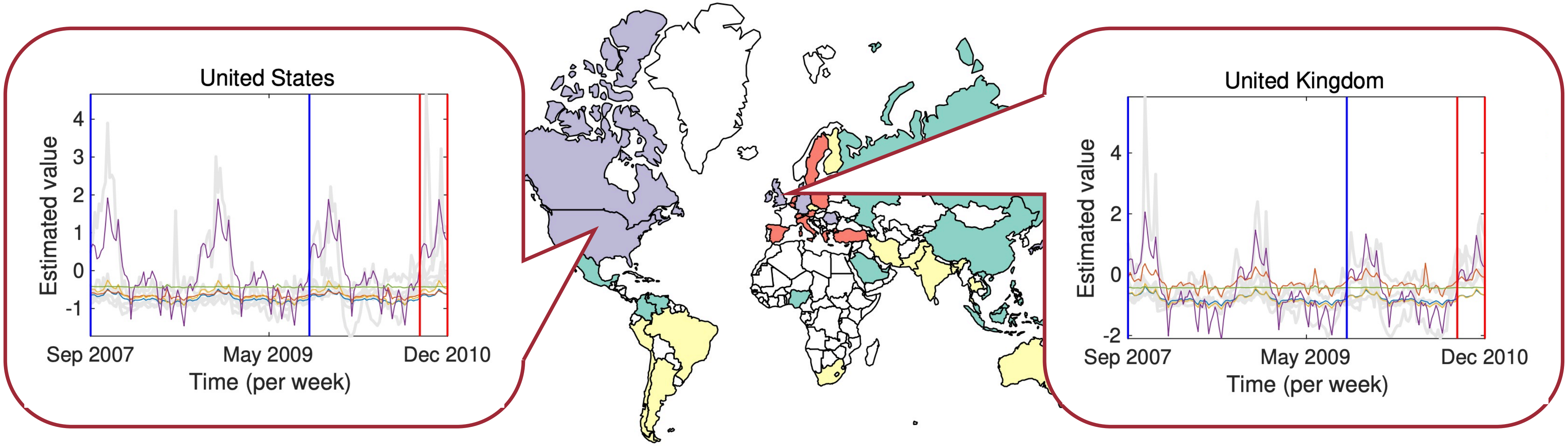
Q1. Effectiveness

- Apparel company dataset: **Local groups**
  - Number of groups $m = 4$



Q1. Effectiveness

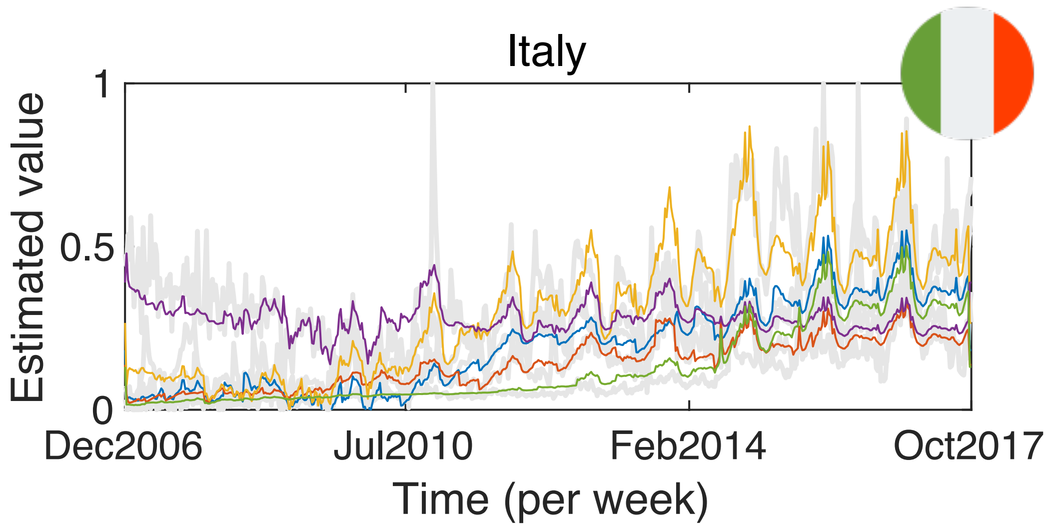
- Apparel company dataset: **Local groups**
  - Number of groups $m = 4$



CubeCast successfully finds local groups

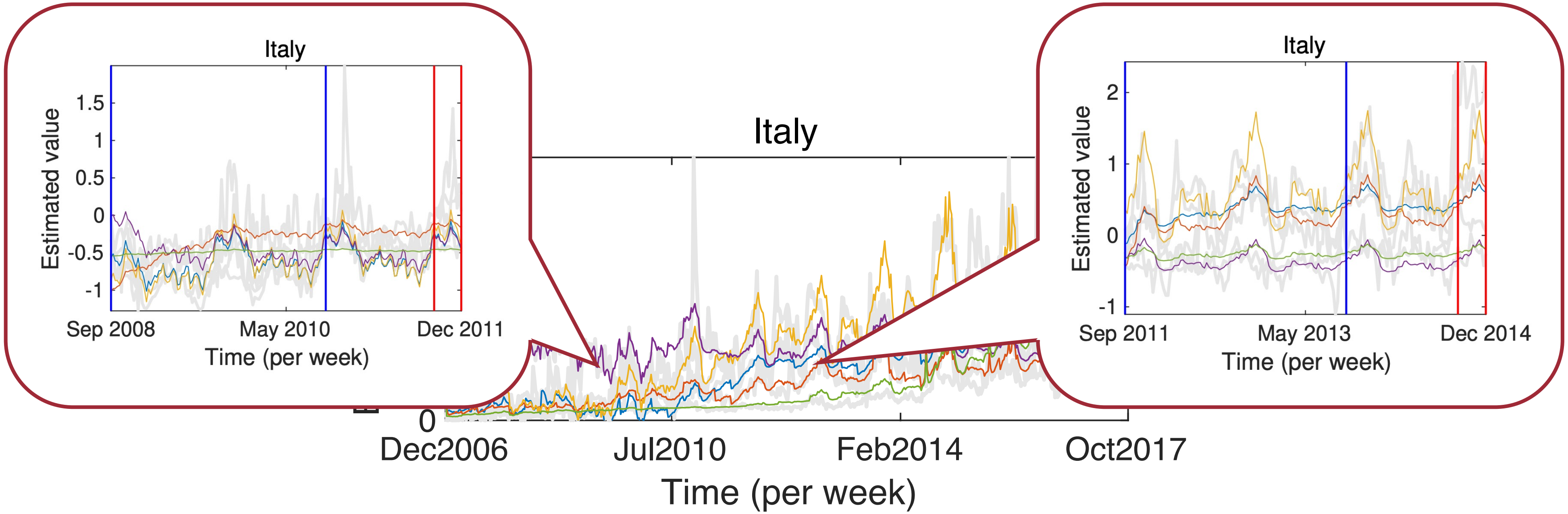
Q1. Effectiveness

- Apparel company dataset: **Dynamical changes**
  - May 2010 → May 2013



Q1. Effectiveness

- Apparel company dataset: **Dynamical changes**
  - May 2010 → May 2013

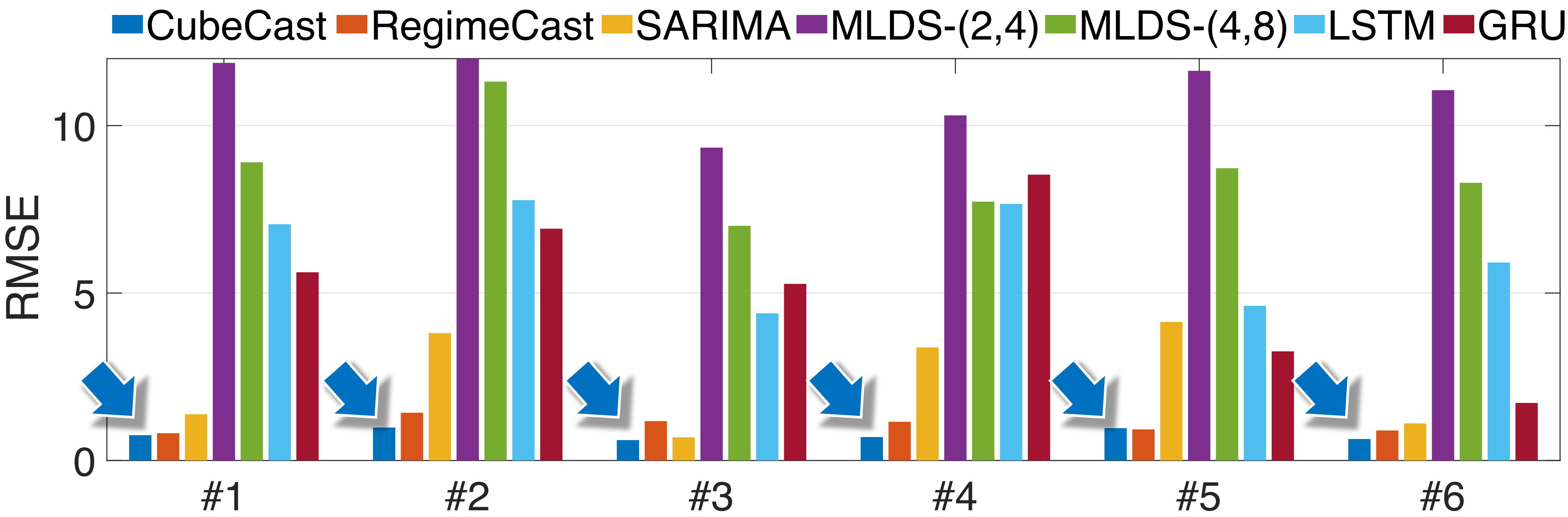


CubeCast can detect dynamical changes automatically

Q2. Accuracy

- **<u>Average RMSE:</u>** lower is better

52-step ahead forecasting

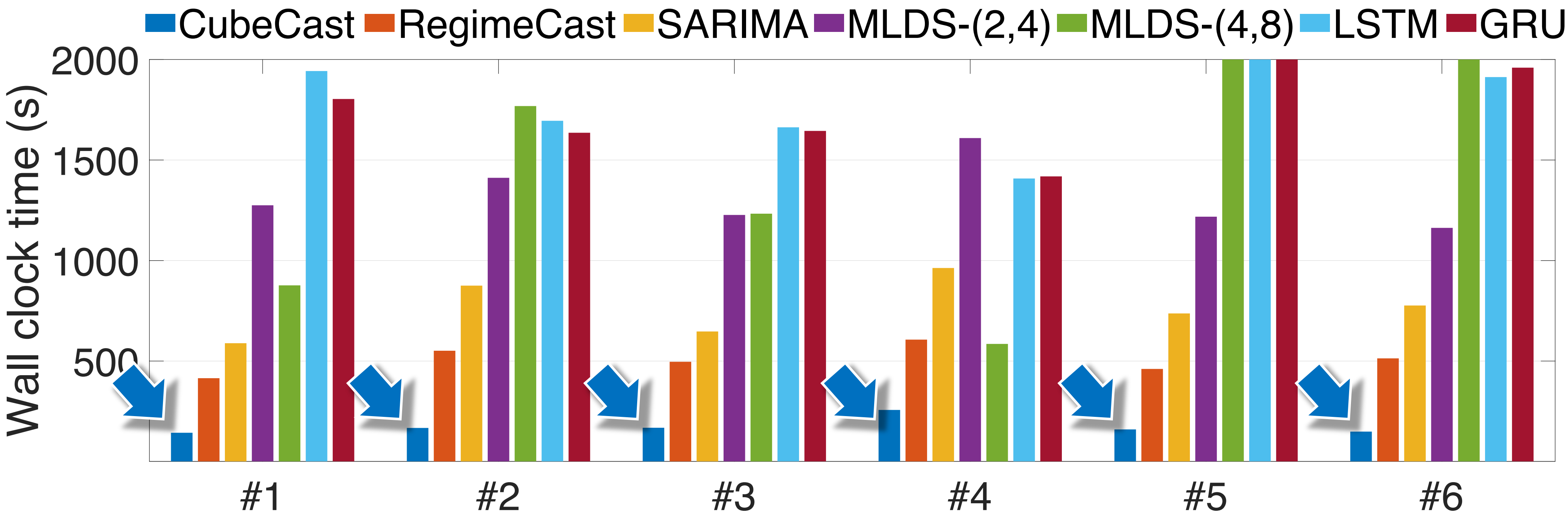


CubeCast can accurately forecast future values

Q3. Scalability

- **<u>Average time:</u>** lower is better

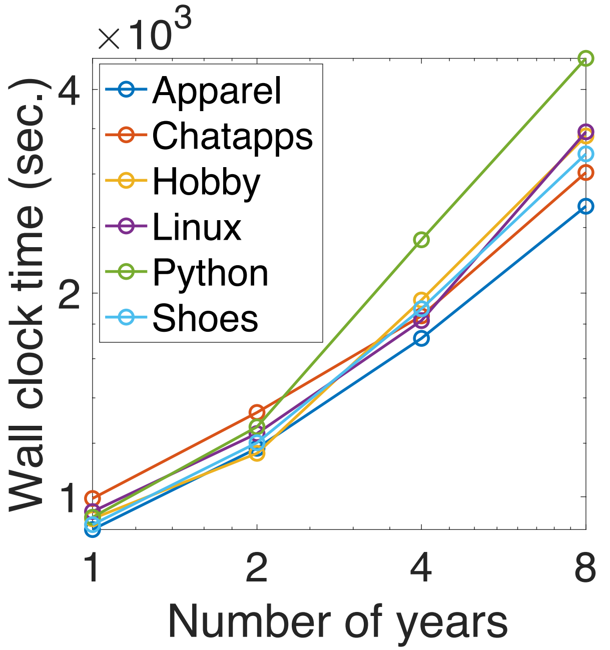
104-steps tensor processing

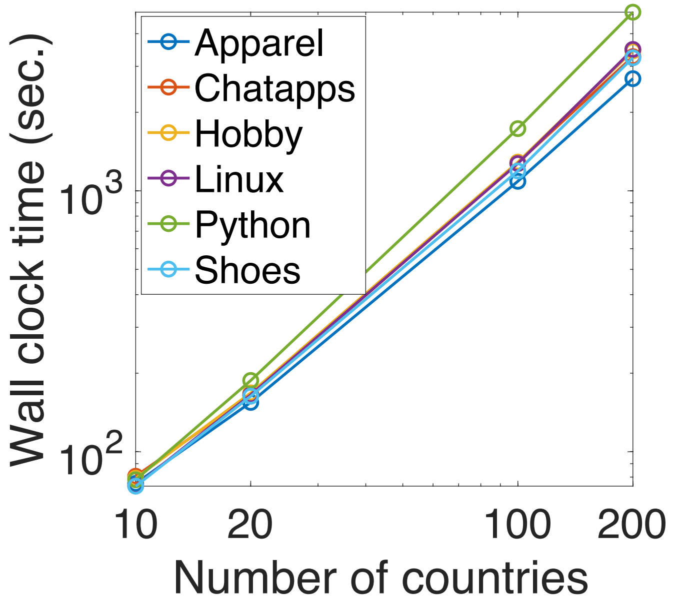


CubeCast can accurately forecast future values

Q3. Scalability

- **<u>Computation time when varying dimensionality</u>**





CubeCast scales linearly

Outline

✓Motivation

✓Problem definition

✓Proposed model

✓Proposed algorithm: CubeCast

✓Experiments

- Conclusion

Conclusion

Effective

It captures complex non-linear dynamics for tensor time series when forecasting long-term future values

Automatic

It automatically recognizes all the components in regimes and their temporal/structural innovations

Scalable

The computation time of CubeCast is independent of the time series length

EOF

Thank you for watching !

- **CubeCast**