



# 時系列テンソルデータストリームに対する関心の拡散パターンの抽出手法

東口慎吾

大阪大学 産業科学研究所  
shingo88@sanken.osaka-u.ac.jp

松原靖子

大阪大学 産業科学研究所  
yasuko@sanken.osaka-u.ac.jp

川畑光希

大阪大学 産業科学研究所  
koki@sanken.osaka-u.ac.jp

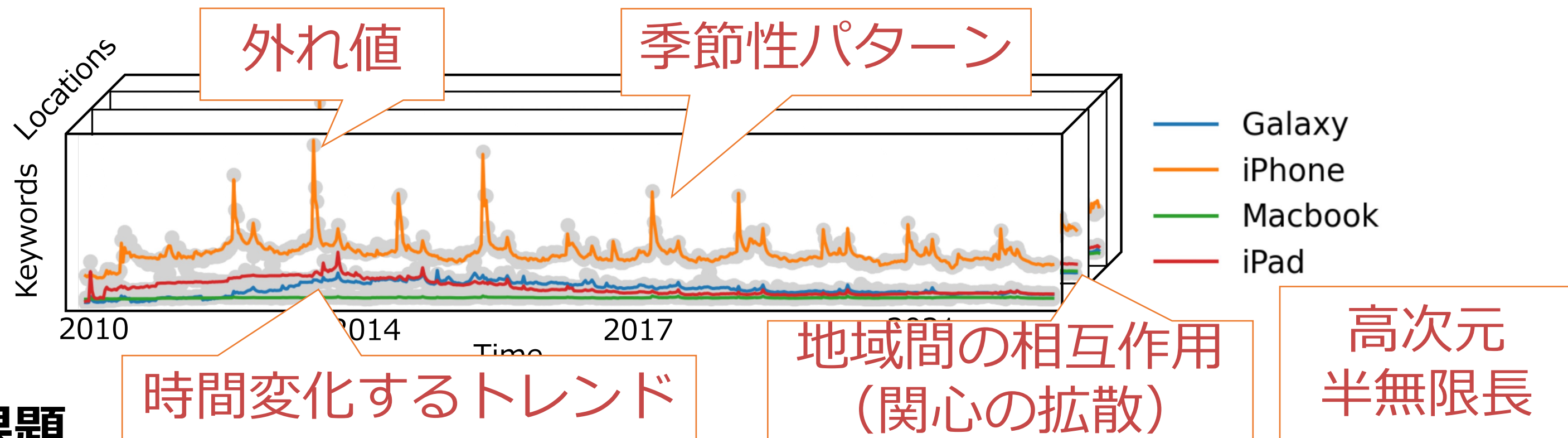
櫻井保志

大阪大学 産業科学研究所  
yasushi@sanken.osaka-u.ac.jp

## モチベーション

**Given:** 社会活動テンソルデータストリーム

e.g., Google検索数データ {Time, Keyword, Location}



課題

(1)複合的なパターン: トレンド・季節性・外れ値・地域間の拡散

(2)高次元かつ半無限長、(3)パターンの時間変化: 流行の変遷

**Goal:** 時間変化する複合的なパターンを抽出し、将来予測を行う

## 結論

データストリームの継続的なモデリングと将来予測を行う手法

**D-Tracker** を提案。提案手法は以下のような特徴を持つ

1. **解釈可能:** トレンドや拡散、季節性の変動を解釈可能な形で抽出
2. **スケーラブル:** 計算時間はデータストリームの長さに依存しない
3. **全自動:** 提案手法はハイパーパラメータを持たない

## 提案モデル

### ①. 背景とモデル概要

**Background – 反応拡散方程式**

化学物質や熱量などの、局所的な増減 (反応項) や 地点間の拡散 (拡散項) を記述する数理モデル

$$\frac{\partial u_i}{\partial t} = f(u_i) + \sum_j D \cdot (u_j - u_i)$$

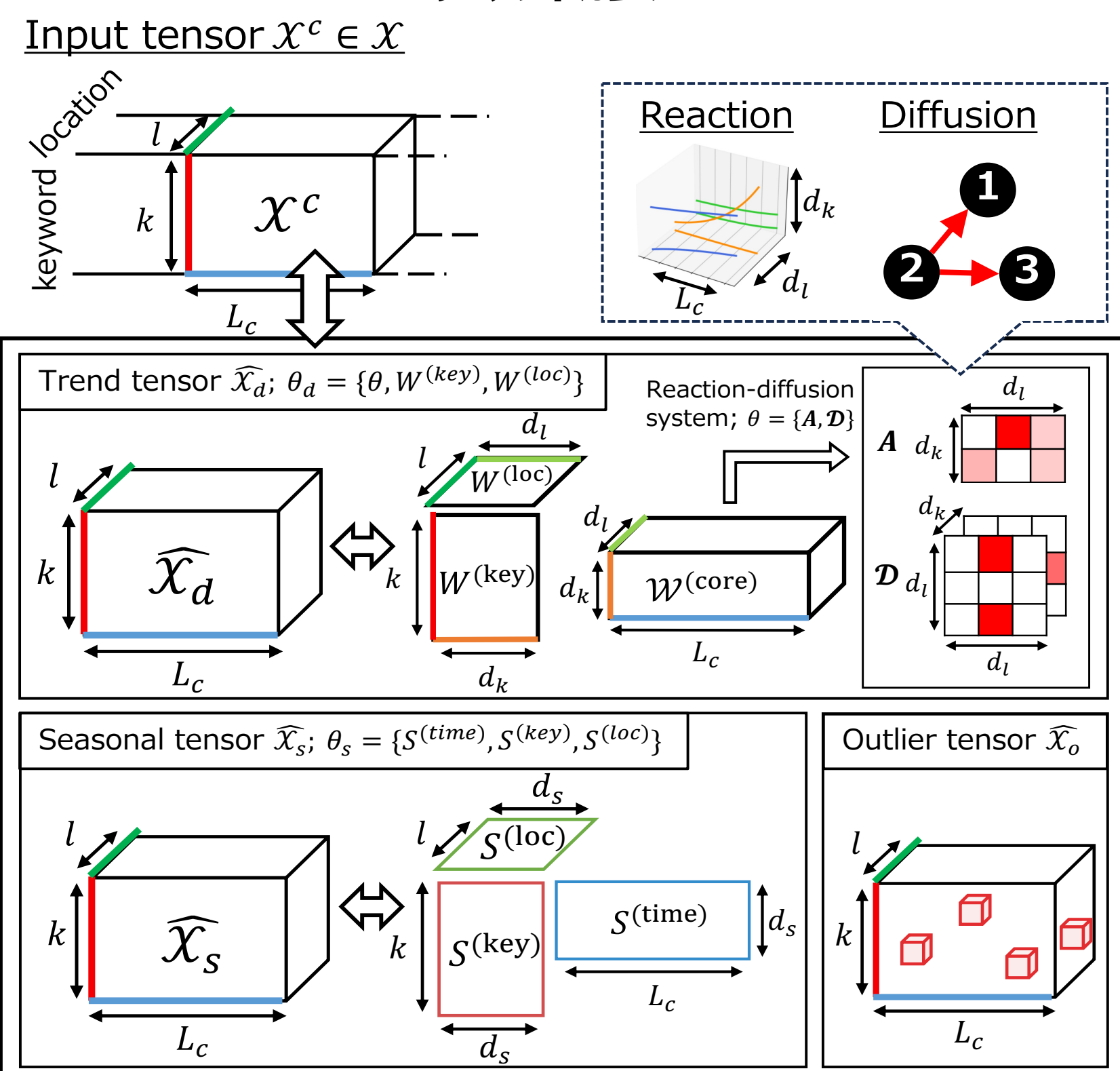
反応項      拡散項

### ② D-Tracker モデル

$$\mathcal{X}^c \approx \hat{\mathcal{X}}_d + \hat{\mathcal{X}}_s + \hat{\mathcal{X}}_o$$

トレンド・拡散      季節性      外れ値

### モデル概要



### トレンド・拡散

反応拡散システムにより生成された潜在ダイナミクスと、キーワード/地域ファクタによってトレンドテンソルを表現

$$\hat{\mathcal{X}}_d = \mathcal{W}^{(\text{core})} \times_{\text{key}} \mathcal{W}^{(\text{key})} \times_{\text{loc}} \mathcal{W}^{(\text{loc})} \quad \theta_d = \{\theta, \mathcal{W}^{(\text{key})}, \mathcal{W}^{(\text{loc})}\}$$

潜在ダイナミクス      キーワード/地域ファクタ (重み)

### 反応拡散システム (Reaction-diffusion system, RDS)

$$\frac{dw_{ij}}{dt} = a_{ij} w_{ij} + \sum_{j'} d_{ijj'} (w_{ij'} - w_{ij}), \quad (1 \leq i \leq d_k, 1 \leq j \leq d_l)$$

反応項      拡散項       $\theta = \{A, D\}$

$a_{ij}$ :  $w_{ij}$ の増加率       $d_{ijj'}$ :  $w_{ij'}$ から $w_{ij}$ への拡散の強さ  
潜在ダイナミクス  $\mathcal{W}^{(\text{core})}$ は反応拡散システムにより生成

### 季節性

潜在的な季節性ダイナミクスと、キーワード/地域ファクタによって季節性テンソルを表現

$$\hat{\mathcal{X}}_s = \mathcal{S}^{(\text{time})} \times_{\text{key}} \mathcal{S}^{(\text{key})} \times_{\text{loc}} \mathcal{S}^{(\text{loc})} \quad \theta_s = \{\mathcal{S}^{(\text{time})}, \mathcal{S}^{(\text{key})}, \mathcal{S}^{(\text{loc})}\}$$

潜在ダイナミクス      キーワード/地域ファクタ (重み)

Single model parameter:  $\theta = \{\theta_d, \theta_s, \hat{\mathcal{X}}_o\}$

複数のモデルを切り替えることで、時間変化するパターンを捉えてデータストリーム全体をモデリング

$$\theta_1, \theta_2, \theta_3, \dots, \theta_N \quad \text{Fullmodel parameter: } \mathcal{F} = \{\theta_1, \theta_2, \dots, \theta_N\}$$

## 最適化アルゴリズム

2つのサブアルゴリズムにより構成される

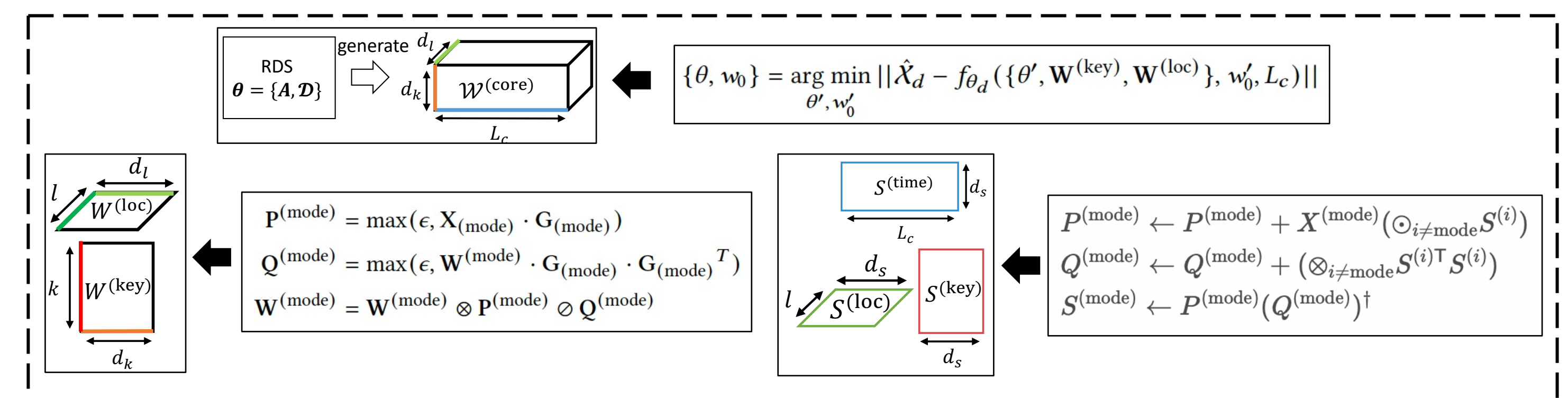
- ① **ModelEstimation:** 現在のウィンドウ ( $\mathcal{X}_c$ ) からモデル ( $\theta$ ) をスクラッチで推定
- ② **ModelUpdate:** 1で推定したモデルをフルパラメータ ( $\mathcal{F}$ ) に加えて新しいモデルにスイッチするかどうかを判定する (MDL理論に基づく目的関数)

### ① ModelEstimation

**Given:** 入力テンソル  $\mathcal{X}_c$

**Object:**  $\min_{\theta_d, \theta_s, \hat{\mathcal{X}}_o} \|\mathcal{X} - \hat{\mathcal{X}}_d - \hat{\mathcal{X}}_s - \hat{\mathcal{X}}_o\|$

**Idea:** 交互最小二乗法 (ALS) により推定



### ② ModelUpdate

新しいデータが到着した時、

**新しいモデルにスイッチする or 現在のモデルを使う**

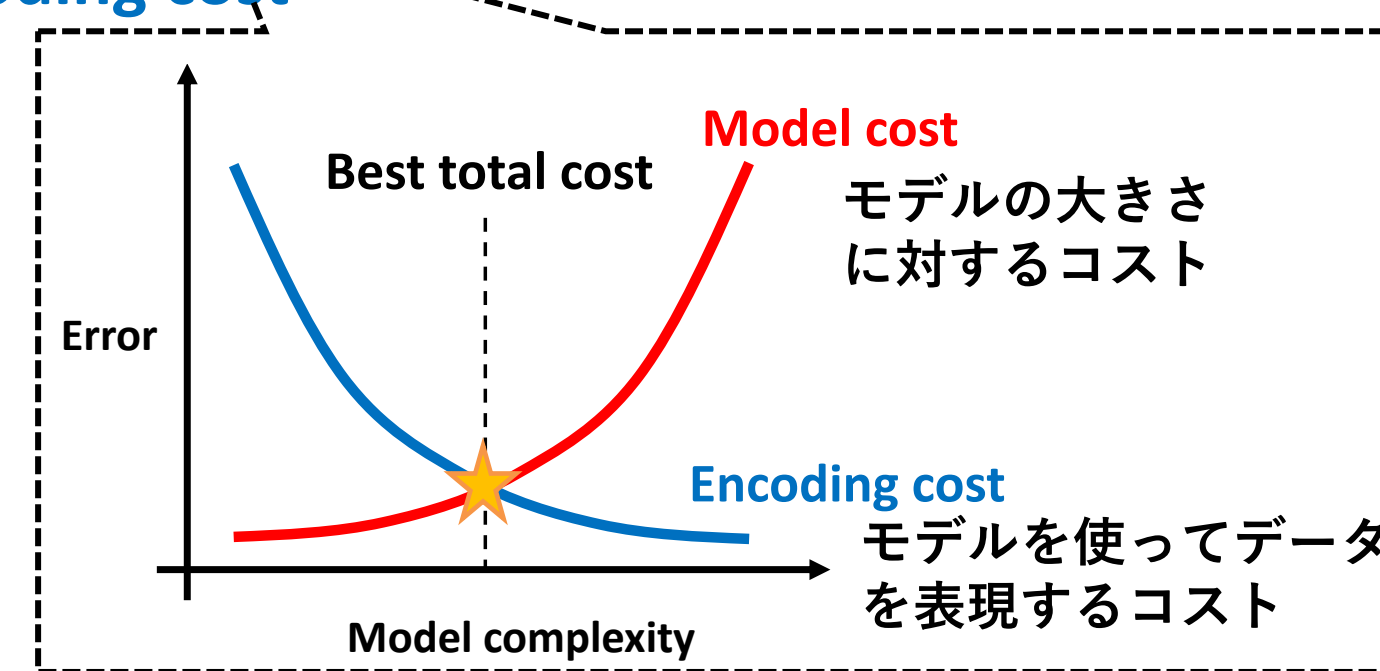
をMDL理論に基づく目的関数より判定

$$\text{目的関数} \quad \langle \mathcal{X}; \mathcal{F} \rangle = \langle \mathcal{F} \rangle + \langle \mathcal{X} | \mathcal{F} \rangle$$

Total cost      Model cost      Encoding cost

**Model Cost:**  $\langle \mathcal{F} \rangle = \sum_{\theta \in \mathcal{F}} \langle \theta \rangle$ ,  $\langle \theta \rangle = \langle \theta_d \rangle + \langle \theta_s \rangle$

$$\begin{aligned} \langle \theta_d \rangle &= \langle \mathcal{W}^{(\text{key})} \rangle + \langle \mathcal{W}^{(\text{loc})} \rangle + \langle \theta \rangle \\ \langle \mathcal{W}^{(\text{key})} \rangle &= |\mathcal{W}^{(\text{key})}| \cdot (\log d_k + \log k + c_F) + \log^* |\mathcal{W}^{(\text{key})}| \\ \langle \mathcal{W}^{(\text{loc})} \rangle &= |\mathcal{W}^{(\text{loc})}| \cdot (\log d_l + \log l + c_F) + \log^* |\mathcal{W}^{(\text{loc})}| \\ \langle \theta \rangle &= \langle A \rangle + \langle D \rangle \\ \langle A \rangle &= |A| \cdot (\log d_k + \log d_l + c_F) + \log^* |A| \\ \langle D \rangle &= |D| \cdot (\log d_k + 2 \log d_l + c_F) + \log^* |D| \\ \langle \theta_s \rangle &= \langle \mathcal{S}^{(\text{time})} \rangle + \langle \mathcal{S}^{(\text{key})} \rangle + \langle \mathcal{S}^{(\text{loc})} \rangle \\ \langle \mathcal{S}^{(\text{time})} \rangle &= |\mathcal{S}^{(\text{time})}| \cdot (\log d_s + \log L_c + c_F) + \log^* |\mathcal{S}^{(\text{time})}| \\ \langle \mathcal{S}^{(\text{key})} \rangle &= |\mathcal{S}^{(\text{key})}| \cdot (\log d_s + \log k + c_F) + \log^* |\mathcal{S}^{(\text{key})}| \\ \langle \mathcal{S}^{(\text{loc})} \rangle &= |\mathcal{S}^{(\text{loc})}| \cdot (\log d_s + \log l + c_F) + \log^* |\mathcal{S}^{(\text{loc})}| \end{aligned}$$



**Data Encoding Cost:**  
Negative log-likelihood

$$\langle \mathcal{X} | \mathcal{F} \rangle = \sum_{x \in \mathcal{X}} -\log_2 p_{\mu, \sigma}(x - \hat{x}_d - \hat{x}_s)$$

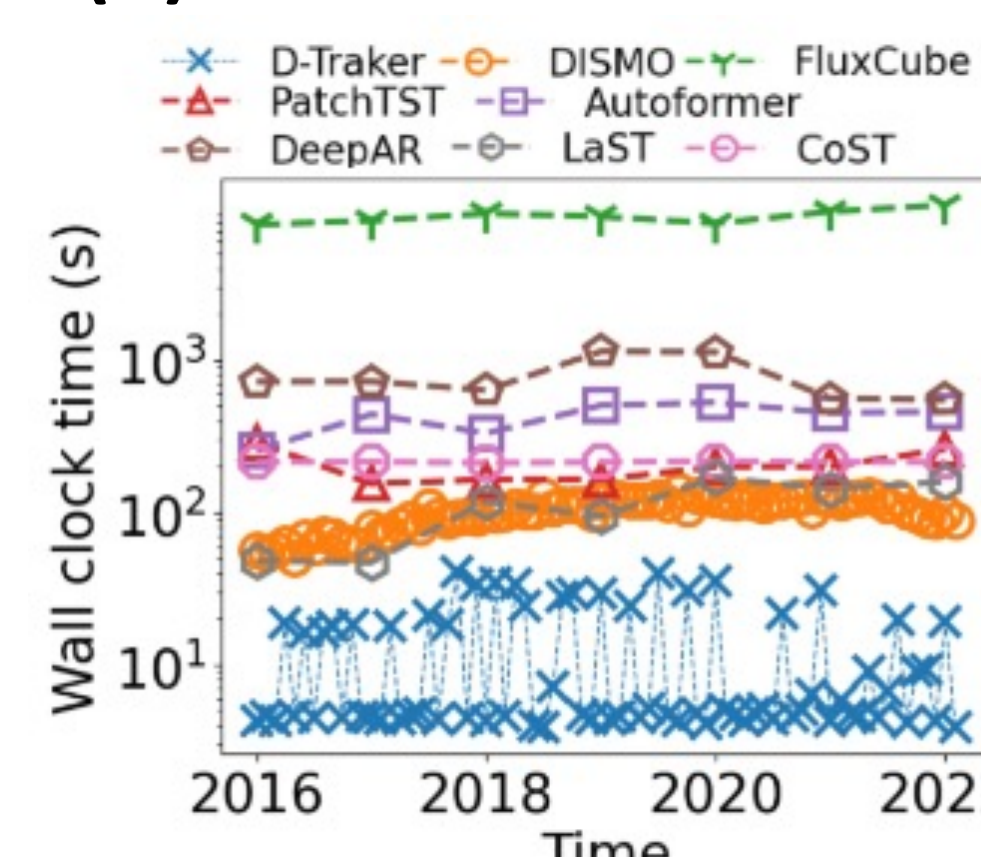
## 実験

### (a) 予測精度

Table 3: Forecasting performance comparison respectively ( $\times 10^{-2}$ ).

Method	D-Tracker	DISMO	FluxCube	PatchTST	Autoformer	DeepAR	LaST	CoST
#1 Device	13 0.84 1.82 1.08 2.18 3.86 7.20 1.39 3.20 1.21 2.03 2.14 2.99 1.91 4.00 1.63 3.45	26 0.90 1.95 1.19 2.45 3.85 7.26 1.54 3.19 1.31 2.13 2.21 3.16 2.07 4.31 1.80 3.79	39 0.97 2.18 1.22 2.60 3.97 7.63 1.21 2.76 1.34 2.22 2.37 3.34 2.37 5.83 1.74 3.77					
#2 Pythonlib	13 0.27 1.42 0.47 1.74 0.86 3.57 0.89 4.26 0.49 1.45 2.01 3.84 0.71 2.54 0.40 2.46	26 0.32 1.54 0.52 1.96 0.93 3.68 0.95 4.17 0.47 1.35 2.01 3.84 0.86 2.80 0.45 2.76	39 0.37 1.78 0.57 2.22 1.09 4.08 0.83 3.33 0.42 1.52 2.01 3.86 0.92 2.99 0.47 3.08					
#3 VoD	13 0.53 1.58 0.56 1.64 2.83 6.96 0.95 2.17 0.56 1.33 1.77 2.23 2.91 13.26 2.98 14.29	26 0.59 1.83 0.66 1.91 2.83 6.95 0.84 1.82 0.74 1.63 1.88 2.42 2.53 11.42 2.36 11.17	39 0.63 2.02 0.78 2.22 2.79 6.84 1.69 2.57 1.13 2.07 1.98 2.61 1.55 5.50 2.27 15.18					
#4 Chatapp	13 0.28 0.83 0.42 1.16 1.47 4.06 0.47 0.81 0.84 1.34 1.85 2.35 2.66 13.16 1.76 7.30	26 0.34 1.01 0.48 1.21 1.48 4.02 0.74 1.13 0.81 1.43 2.16 3.04 2.34 9.91 2.09 9.11	39 0.41 1.24 0.50 1.27 1.47 3.93 0.88 1.53 0.54 1.41 1.93 2.61 2.18 8.59 2.18 7.48					
#5 Language	13 0.31 0.97 0.94 2.25 1.38 3.34 0.41 1.47 0.37 0.85 1.98 3.63 0.74 1.94 0.40 1.50	26 0.35 1.14 1.12 2.68 1.42 3.33 0.50 1.97 0.41 0.93 2.03 3.65 0.80 2.13 0.47 1.73	39 0.40 1.32 1.36 3.19 1.51 3.44 0.51 2.50 0.54 1.08 2.01 3.55 0.81 2.13 0.49 1.79					
#6 Covid-19	7 0.47 1.40 1.12 2.87 1.95 2.75 1.60 5.55 1.10 2.39 4.60 5.41 2.39 7.91 1.23 7.07	14 0.70 2.42 1.33 3.94 1.97 2.85 2.04 7.64 1.18 2.97 2.30 5.88 2.10 7.78 1.45 8.50	21 0.98 3.65 1.58 5.83 1.98 3.08 2.24 17.30 1.25 3.25 2.61 4.40 1.74 5.71 1.72 9.14					

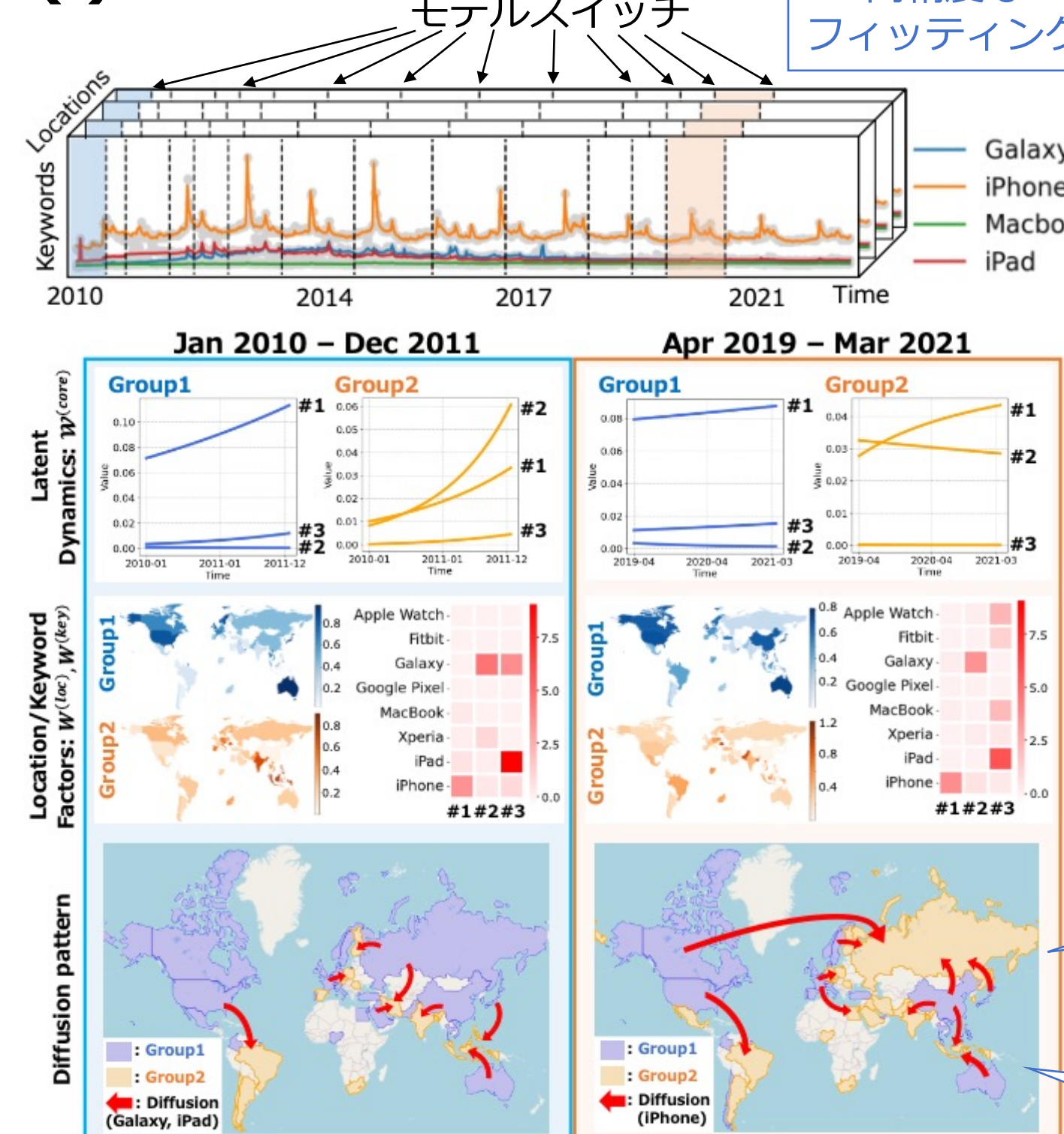
### (b) スケーラビリティ



D-Trackerの計算時間はデータストリームの長さに依存しない

## ケーススタディ

### (i) Device



### (ii) Python library

