The Thirty-Eighth Annual Conference on Neural Information Processing Systems (NeurIPS 2024)

NEURAL INFORMATION PROCESSING SYSTEMS

Continuous Temporal Domain Generalization

Zekun Cai[1,4], Guangji Bai[2], Renhe Jiang[1*], Xuan Song[3,4], Liang Zhao[2]

[1] The University of Tokyo [2] Emory University

[3] Jilin University [4] Southern University of Science and Technology

TL;DR

- This study formalizes Continuous Temporal Domain Generalization (CTDG), where **domain distribution evolve continuously, and domains are observed at arbitrary times**.
- We introduces a novel method to **generate neural networks at any given time**, aligning with the evolving data distributions.

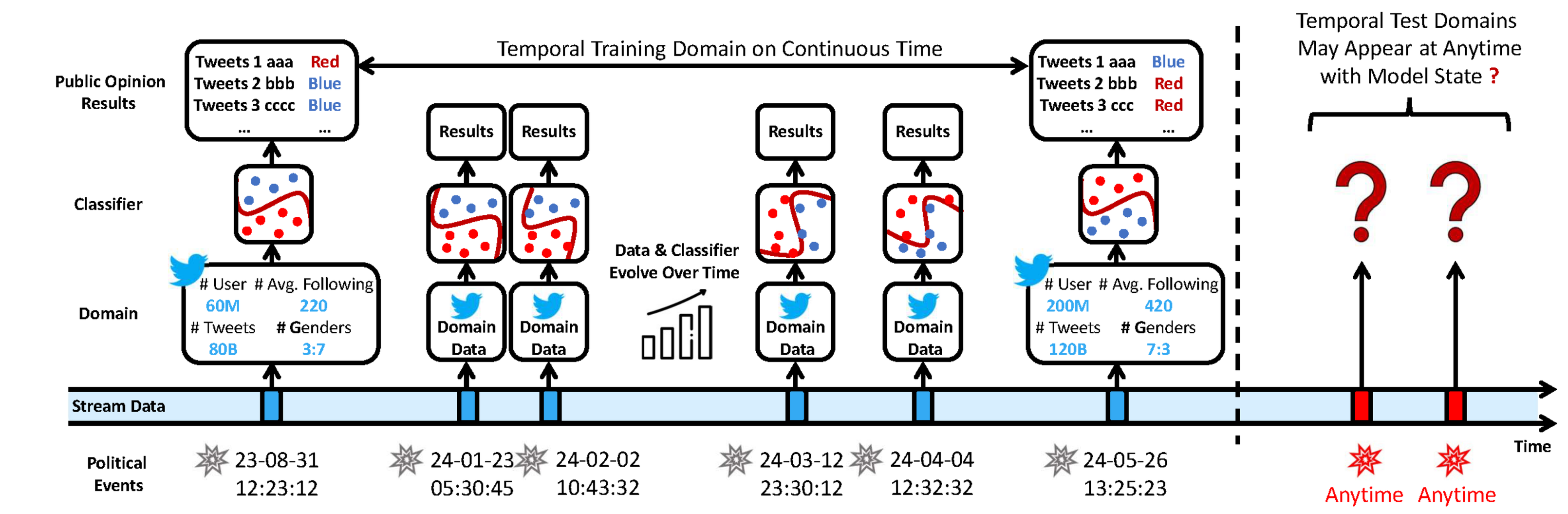


Fig 1. An example of continuous temporal domain generalization. Consider training classification models for public opinion prediction via tweets, where the training domains are only available at specific political events (e.g., presidential debates), we wish to generalize the model to any future based on the underlying data distribution drift within the time-irregularly distributed training domains.

Core Challenges

Traditional Temporal Domain Generalization

- × Temporal domains are constrained to fixed time intervals.
- × Primarily focus on single-step generalization.

Continuous Temporal Domain Generalization

- ✓ **Domains are randomly and sparsely distributed along a continuous timeline.**
- ✓ **The model can seamlessly generalize to any given point in time.**
- ✓ **Controlling the generalization process by inductive bias.**

Critical Hurdles:

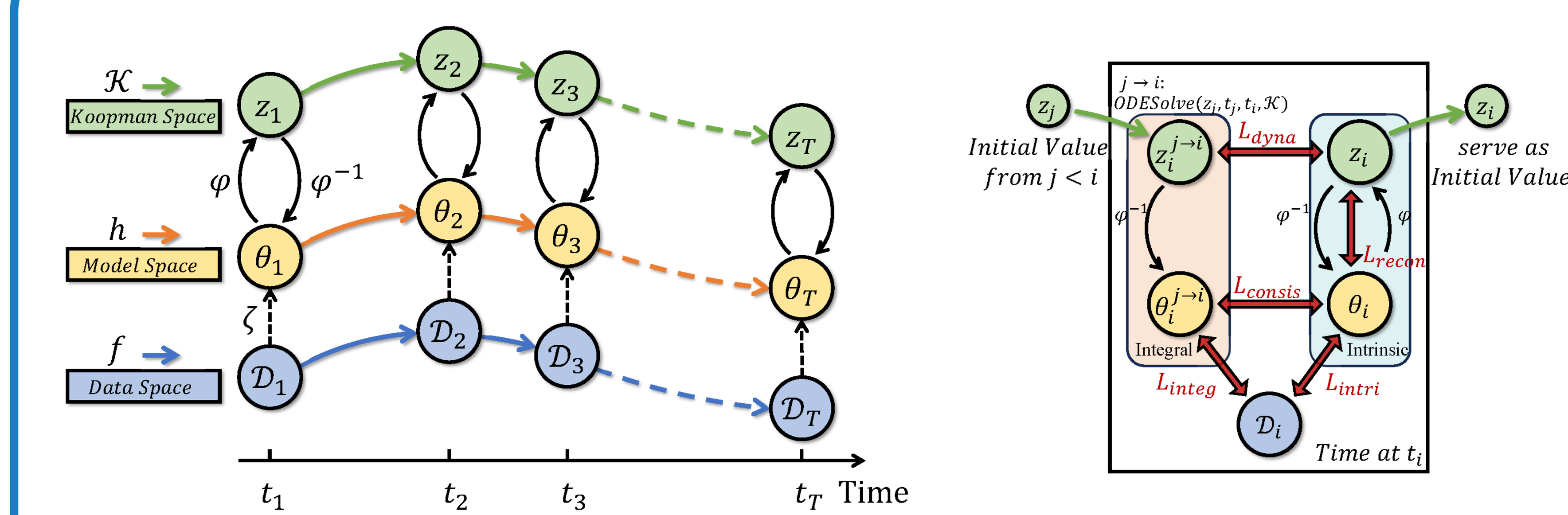
- How to model model dynamics and synchronize them with data dynamics?
- How to capture the dominant dynamics within over-parametrized model?
- How to ensure stability and controllability for long-term generalization?

Problem Definition

Continuous Temporal Domain Generalization (CTDG):

- In CTDG, a domain $\mathcal{D}(t)$ represents a dataset collected at time t, consisting of instances $\{(x_i^{(t)}, y_i^{(t)})\}_{i=1}^{N(t)}$, where $x_i^{(t)} \in X(t)$, $y_i^{(t)} \in Y(t)$ and $N(t)$ denotes the feature, target and the number of instances.
- The focus is on **gradual concept drift in continuous time**, where the conditional probability distribution $P(Y(t)|X(t))$ evolves smoothly over time.
- During training, the model observes a series of domains $\{\mathcal{D}(t_1), \mathcal{D}(t_2), \dots, \mathcal{D}(t_T)\}$ collected at irregular time points $\mathcal{T} = \{t_1, t_2, \dots, t_T\}$. At each $t_i \in \mathcal{T}$, the model learns a predictive function $g(\cdot; \theta(t_i))$ for domain $\mathcal{D}(t_i)$. The goal of CTDG is to model the dynamic evolution of $\theta(t_i)$, enabling the prediction of model parameters $\theta(s)$ at any given time $s \notin \mathcal{T}$.

Koodos: Koopman operator driven CTDG framework



(a) Three Spaces and Parallel Flows in Continuous Temporal Domains (b) State Constraints at a Given Moment

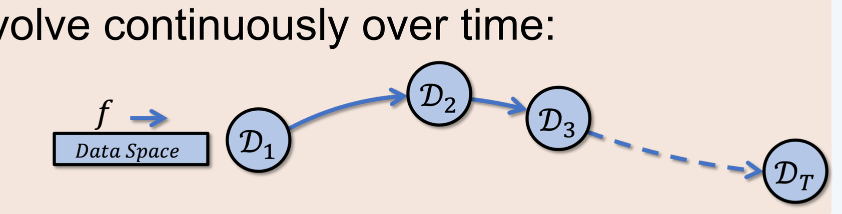
Fig 2. Macro-flows and micro-constraints in the proposed model framework.

Step 1. Characterizing the Continuous Dynamics of the Data and the Model

Key Assumption: Distribution Continuity

- We assume that data distributions evolve continuously over time:

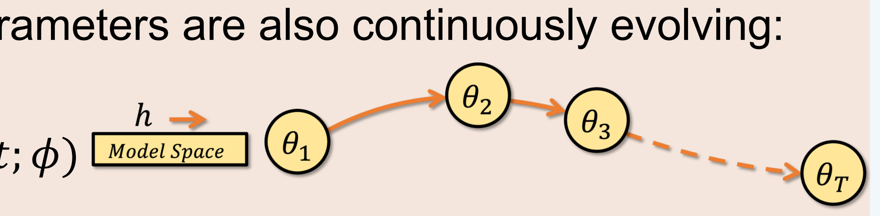
$$\frac{d}{dt}P_t(Y|X) = f(P_t(Y|X), t)$$



Key Theorem: Model Continuity

- It can be proved that the model parameters are also continuously evolving:

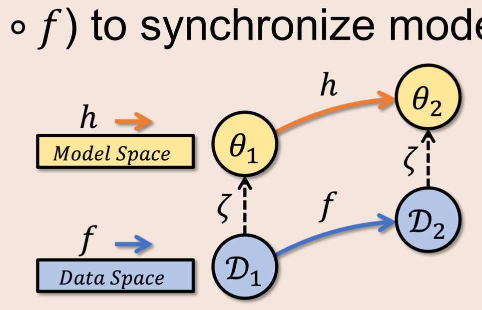
$$\frac{d}{dt}\theta_t = J_g(\theta_t)^{-1} f(g(\cdot;\theta_t), t) = h(\theta_t, t; \phi)$$



Key Methodology : Model Dynamic Systems

- Encouraging topological conjugation ($h \circ \zeta = \zeta \circ f$) to synchronize model dynamics with data dynamics.

$$\phi = \underset{\phi}{\operatorname{argmin}} \sum_{i=1}^{T}\sum_{j=1}^{i} \| \theta_i, \theta_j + \int_{t_j}^{t_i} h(\theta_\tau, \tau; \phi) d\tau \|^2$$



Intuitive: $\mathcal{D}_1 \rightarrow \mathcal{D}_2 \rightarrow \theta_2 = \mathcal{D}_1 \rightarrow \theta_1 \rightarrow \theta_2$

Step 2. Modeling Nonlinear Model Dynamics by Koopman Operators

Model h in high-dimensional θ spaces are computationally intensive and unstable!

- Solution: identify principal dynamics in Koopman Space.

$$z = \varphi(\theta) \qquad \frac{dz}{dt} = \mathcal{K}z \qquad \theta = \varphi^{-1}(z)$$

Koopman Theory provides a method for the global linearization of nonlinear dynamics.

Step 3. Joint Optimization of Model and its Dynamics with Prior Knowledge

- **Joint Optimization** (five constraints as shown in Fig. 2 Right.)
- **Combination of Prior Knowledge**

Dynamic system control techniques can be used for control model generalization!

Operation Interface: $\mathcal{K}$

- Analyzing the eigenvalue of $\mathcal{K}$
- Adding constraints to $\mathcal{K}$

Visualization

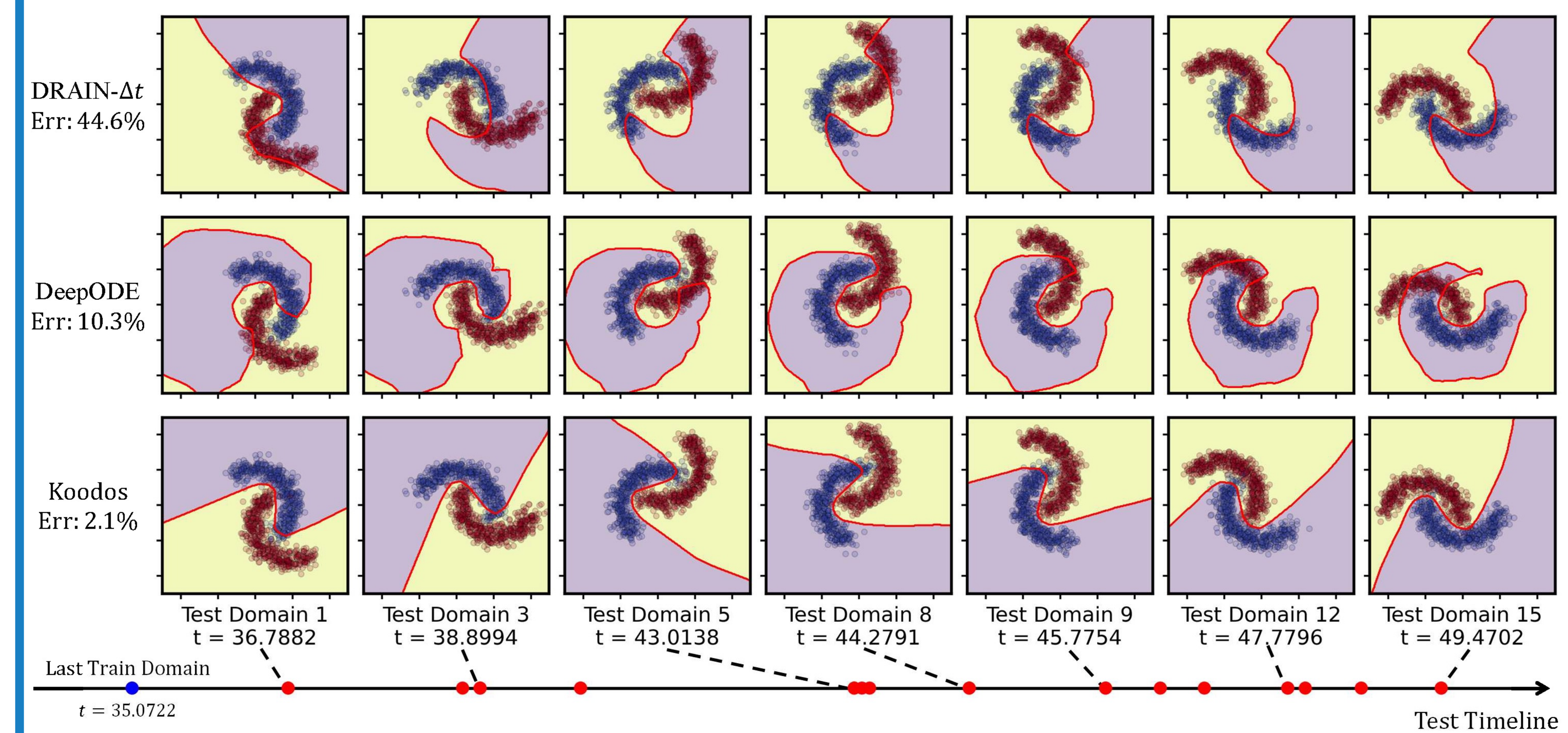


Fig 3. Visualization of decision boundary of the 2-Moons dataset (purple and yellow show data regions, red line shows the decision boundary). Top to bottom compares two baseline methods with ours; left to right shows partial test domains (all test domains are marked with red points on the timeline). All models are learned using data before the last train domain.

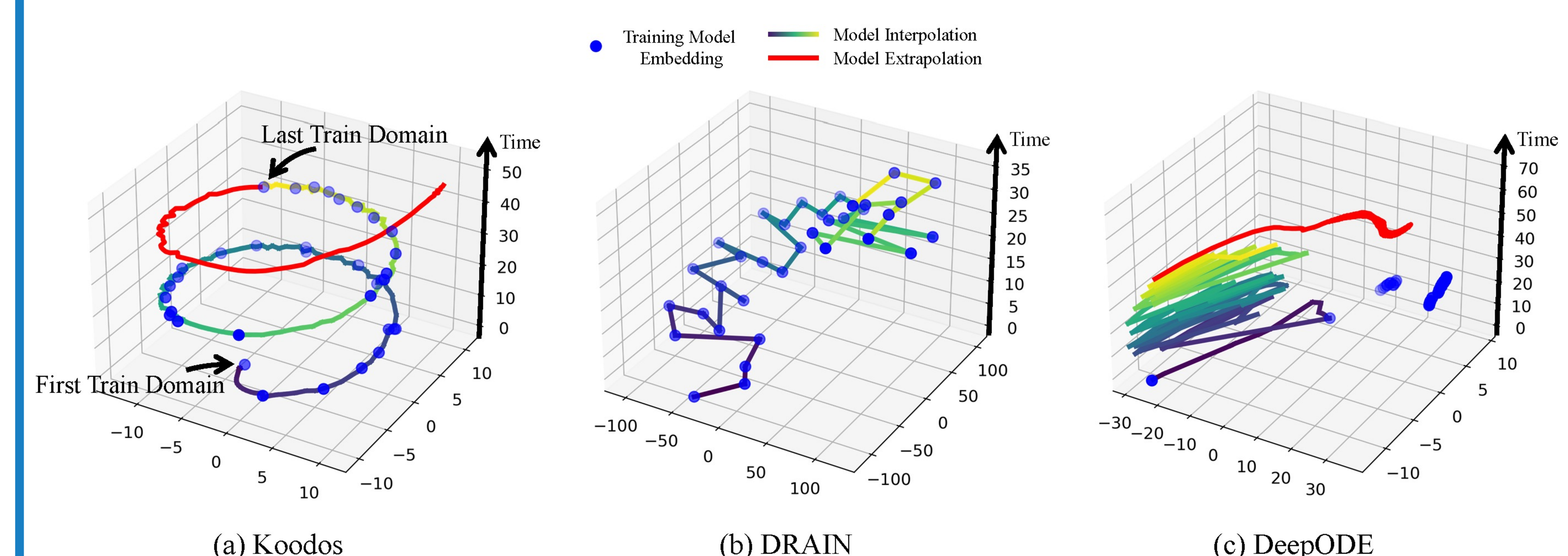


Fig 4. Interpolated and extrapolated predictive model trajectories. Left: Koodos captures the essence of generalization through the harmonious synchronization of model and data dynamics; Middle: DRAIN, as a probabilistic model, fails to capture continuous dynamics, which is presented as jumps from one random state to another. Right: DeepODE demonstrates a certain degree of continuity, but the dynamics are incorrect.

Results

Table 1: Performance comparison on continuous temporal domain datasets. The classification tasks report Error rates (%) except for the AUC for the Twitter dataset. The regression tasks report MAE. 'N/A' implies that the method does not support the task.

Model	Classification				Regression	
	2-Moons	Rot-MNIST	Twitter	Yearbook	Cyclone	House
Offline	13.5 ± 0.3	6.6 ± 0.2	0.54 ± 0.09	8.6 ± 1.0	18.7 ± 1.4	19.9 ± 0.1
LastDomain	55.7 ± 0.5	74.2 ± 0.9	0.54 ± 0.12	11.3 ± 1.3	22.3 ± 0.7	20.6 ± 0.7
IncFinetune	51.9 ± 0.7	57.1 ± 1.4	0.52 ± 0.01	11.0 ± 0.8	19.9 ± 0.7	20.6 ± 0.2
IRM	15.6 ± 0.2	8.6 ± 0.4	0.53 ± 0.11	8.3 ± 0.5	18.0 ± 0.8	19.8 ± 0.2
V-REx	12.8 ± 0.2	8.6 ± 0.3	0.58 ± 0.05	8.9 ± 0.5	17.7 ± 0.5	20.2 ± 0.1
CIDA	18.7 ± 2.0	8.3 ± 0.7	0.63 ± 0.03	8.4 ± 0.8	17.0 ± 0.4	10.2 ± 1.0
TKNets	39.6 ± 1.2	37.7 ± 2.0	0.57 ± 0.04	8.4 ± 0.3	N/A	N/A
DRAIN	53.2 ± 0.9	59.1 ± 2.3	0.57 ± 0.04	10.5 ± 1.0	23.6 ± 0.5	9.8 ± 0.1
DRAIN-Δt	46.2 ± 0.8	57.2± 1.8	0.59 ± 0.02	11.0 ± 1.2	26.2 ± 4.6	9.9 ± 0.1
DeepODE	17.8 ± 5.6	48.6 ± 3.2	0.64 ± 0.02	13.0 ± 2.1	18.5 ± 3.3	10.7 ± 0.4
Koodos (Ours)	**2.8 ± 0.7**	**4.6 ± 0.1**	**0.71 ± 0.02**	**6.6 ± 1.3**	**16.4 ± 0.3**	**9.0 ± 0.2**

Paper Code Mail Wechat