Out-of-Distribution Generalization in Time Series

AAAI 2024 Tutorial

Songgaojun Deng, **Jindong Wang** and **Maarten de Rijke**

Tuesday, 20 February 2024

2:00 pm - 3:45 pm (PST)

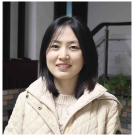
https://ood-timeseries.github.io/

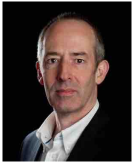
Acknowledgements

Tutorial based in part on materials in the Tutorial at IJCAI 2022: A Tutorial on Domain Generalization and a number of published papers.

Organizers

- Songgaojun Deng I am on the job market!
  - Postdoc Researcher, University of Amsterdam
  - s.deng@uva.nl
  - Machine learning and data mining in social, health informatics and e-commerce; domain generalization in time series
- Jindong Wang
  - Senior Researcher, Microsoft Research Asia
  - jindong.wang@microsoft.com
  - Robust machine learning, out-of-distribution generalization
- Maarten de Rijke
  - Distinguished Professor, University of Amsterdam
  - m.derijke@uva.nl
  - Information retrieval and machine learning





Outline

- Real-world scenarios and motivation
- Background
  - Preliminaries of time series
  - Preliminaries of out-of-distribution generalization
- Problems and challenges
- Methodology
- Datasets, benchmarks and evaluations
- Summary, future directions and discussion

Objectives

- Grasp the problem of out-of-distribution generalization in time series and its specific characteristics
- Understand the current landscape of methods
- Recognize the open challenges and opportunities for further exploration

Real-world scenarios and motivation

Real-world examples of time series predictive tasks facing out-of-distribution data challenges.

Stock market forecasting

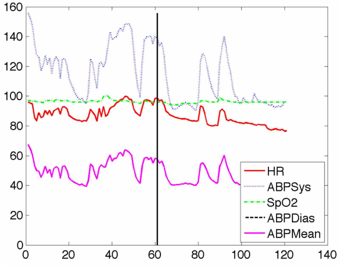
Stock market exhibits instability due to changing external conditions, e.g., different economic conditions, regulations, and trading behaviors.

Movement of the Dow Jones Industrial Average (DJIA) between 01/2017 and 12/2020, showing the pre-crash high on 12/02/2020, and the subsequent crash during the COVID-19 pandemic and recovery to new highs to close 2020.

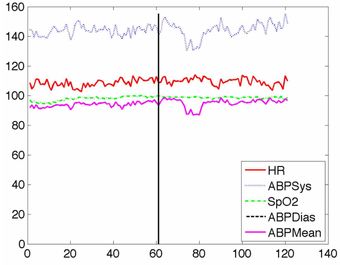
Image credits: Wikipedia

Physiological data analysis

Patient sensor data (e.g., heart rate, ambulatory blood pressure (ABP)) show different distributions due to varying physical conditions and events.



(a) Patient A



(b) Patient B

Multivariate time series data of patients. Patient A had experienced Arterial Hypotensive Episode (AHE) events, whereas Patient B did not.

Image credits: [Sun et al., 2010]

Product demand forecasting

In retail, product demand and sales patterns sometimes shift over time, resulting in distribution changes due to unexpected buying behaviors or economic fluctuations.

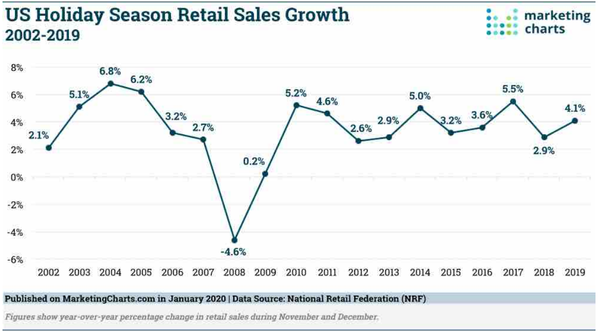


Image credits: marketingcharts.com

Incorporating new products or opening new stores in a retail chain also introduces new data distributions.

Vehicle intention prediction

Autonomous vehicles need to navigate in dynamically changing environments, e.g., unexpected road scenarios such as obstacles, emergency vehicles, and other vehicles breaking down.

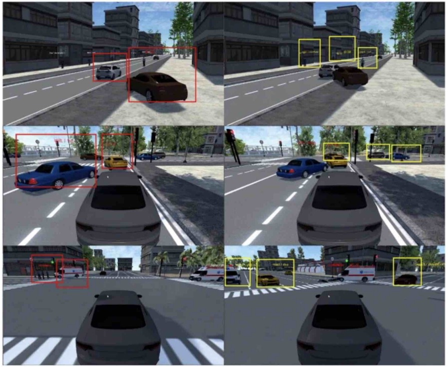


Image credits: [Wen et al., 2020a]

Systematic failure

A model trained on time series data fails when faced with new, unseen data, as

- the model's predictive accuracy can be compromised by data shifts, and
- the lack of abundant data on various real-world conditions for machine learning training.

To address this ...

Out-of-distribution generalization in time series

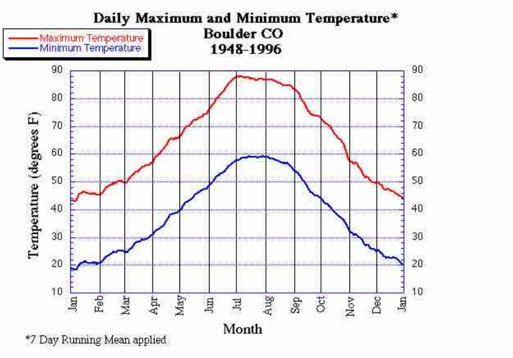
- Models are expected to generalize to unseen scenarios/domains in time series predictive tasks.

Background

Preliminaries of time series

Time series data

Time series is a sequence of data points indexed in time order.



Plot of daily average max and min temperature in Boulder CO.

Image credits: Physical Sciences Laboratory

Characteristics of time series data

Many time series exhibit one or more of the following characteristics:

- Trends, seasonal, cycle, irregular

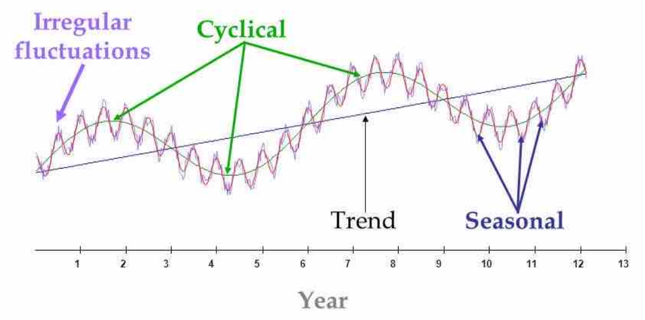
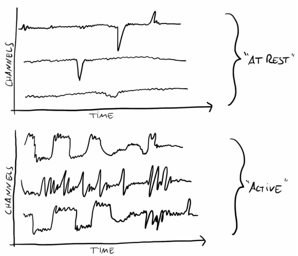


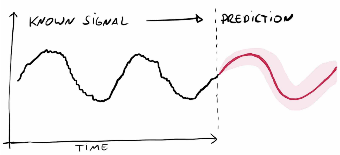
Image credits: Github of Ashish Patel.

Time series predictive tasks

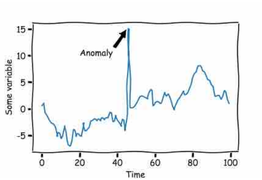
Some popular predictive tasks that model time series data.



(a) Time series classification (e.g., human activity recognition)



(b) Time series forecasting (e.g., stock price forecasting)



(c) Anomaly detection (e.g., fraud detection)

Image credits: Blog of Florian Huber and TSOD Github.

Traditional time series models

- Autoregressive (AR)

$$X_t = \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t,$$

  where p is the order, $\varphi_1, \ldots, \varphi_p$ are model parameters, and ε_t is white noise.

- Moving Average (MA)

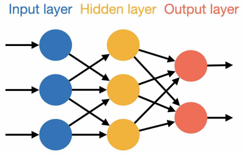
$$\text{Simple moving average (SMA)}_k = \frac{1}{k} \sum_{i=n-k+1}^{n} p_i,$$

  where k is the window size, and n is the total number of observed values.

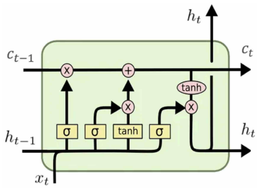
- Autoregressive Integrated Moving Average (ARIMA)

$$\text{AR} + \text{MA} + \text{I (preliminary differencing procedure)}$$

- Multilayer Perceptron (MLP)



- Long Short-Term Memory Networks (LSTMs) [Hochreiter and Schmidhuber, 1997]



- Transformer [Vaswani et al., 2017]

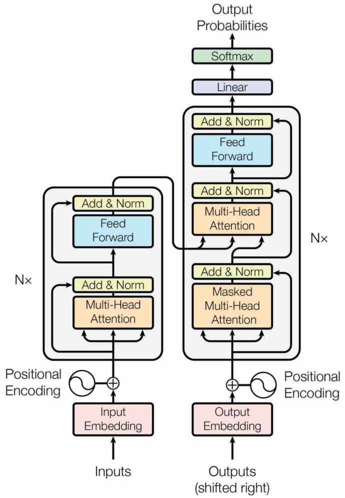
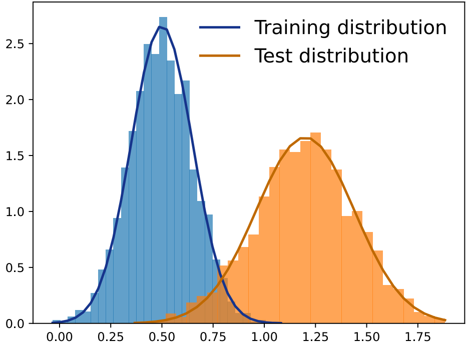


Image credits: A blog and [Vaswani et al., 2017]

Preliminaries of out-of-distribution (OOD) generalization

Distribution shifts in OOD generalization

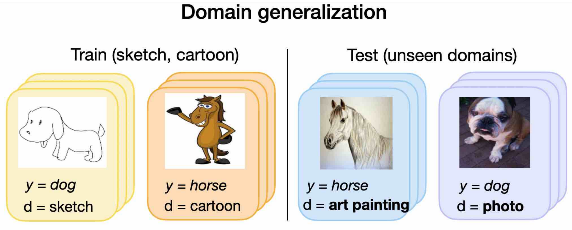
Distribution shifts denote the training distribution differs from the test distribution.



Two types of distribution shifts

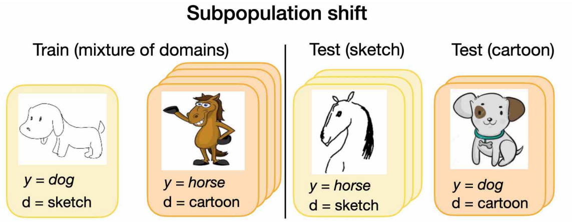
Domain generalization (our focus)

- Train and test on disjoint sets of domains.



Subpopulation shift

- Training and test domains overlap, but their relative proportions differ.



Formal definition of domain generalization

Domain: A domain is composed of data samples that are sampled from a distribution, denoted as $\mathcal{D}^d = \{(X^d, Y^d)\}^{n_d} \sim \mathbb{P}^d(X, Y)$. **Data samples** (X, Y) consists of the input observation X and the corresponding label Y.

Domain generalization (DG): Given M training (source) domains $\mathcal{D}_{\text{train}} = \{\mathcal{D}^i | i = 1, \dots, M\}$. The goal of DG is to learn a generalizable predictive function $h : \mathcal{X} \to \mathcal{Y}$ from the M training domains to achieve a minimum prediction error on unseen test domains $\mathcal{D}_{\text{test}}$ (i.e., $\mathbb{P}^i(X, Y) \neq \mathbb{P}^{\text{test}}(X, Y)$):

$$\min_h \mathbb{E}_{(X,Y)\in\mathcal{D}_{test}}[\ell(h(X), Y)],$$

where $\mathbb{E}$ is the expectation and $\ell(\cdot, \cdot)$ is the loss function.

Overview of DG methodology

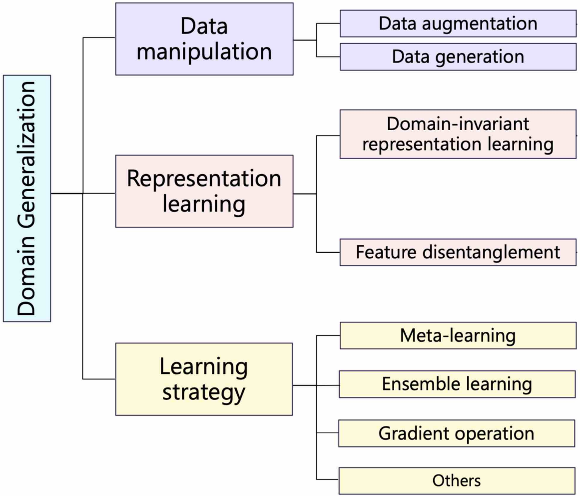


Image credits: A Tutorial on Domain Generalization

Wang et al. Generalizing to unseen domains: a survey on domain generalization. IEEE TKDE 2022.

Data manipulation

Manipulating the inputs to assist in learning general representations, by increasing data quality and quantity.

$$\min_h \mathbb{E}_{(X,Y)}[\ell(h(X,Y)] + \mathbb{E}_{(X',Y)}[\ell(h(X',Y)]$$

- Domain randomization (DR) [Yue et al., 2019]: Randomly draw K real-life categories from ImageNet for stylizing the source images.

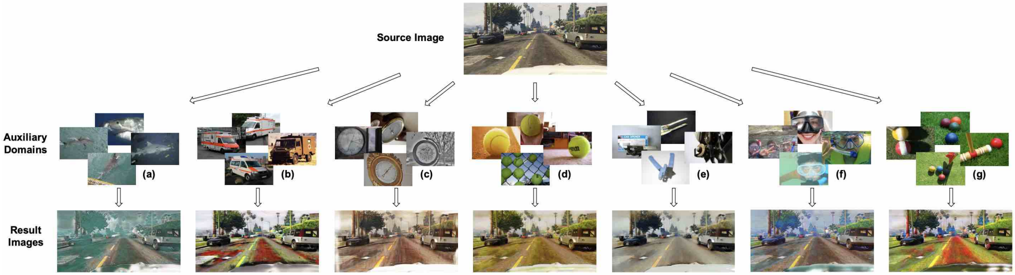


Image credits: [Yue et al., 2019]

Representation learning

Learning domain-invariant representations or disentangling the features into domain-shared or domain-specific parts.

- Domain adversarial neural network (DANN) [Ganin and Lempitsky, 2015]: Adopt a gradient reversal layer and update the feature extractor to fool the domain classifier by generating domain-invariant representations.

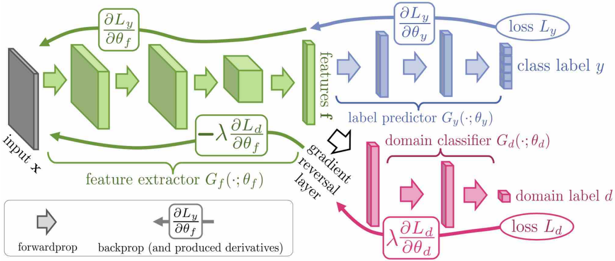
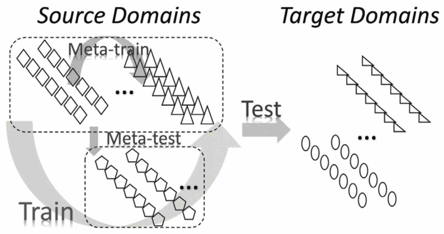


Image credits: [Ganin and Lempitsky, 2015]

Learning strategy

Exploiting learning strategies, such as meta-learning, ensemble learning, and gradient operation, to promote the generalization capability.

- Meta-learning Domain Generalization (MLDG) [Li et al., 2018]: Simulate train/test domain shift during training by synthesizing virtual testing domains within each mini-batch.



Algorithm 1 Meta-Learning Domain Generalization

procedure MLDG
  Input: Domains $\mathcal{S}$
  Init: Model parameters Θ. Hyperparameters α, β, γ.
  for ite **in** iterations **do**
    Split: $\bar{\mathcal{S}}$ and $\breve{\mathcal{S}} \leftarrow \mathcal{S}$
    Meta-train: Gradients $\nabla_\Theta = \mathcal{F}'_\Theta(\bar{\mathcal{S}};\Theta)$
    Updated parameters $\Theta' = \Theta - \alpha\nabla_\Theta$
    Meta-test: Loss is $\mathcal{G}(\breve{\mathcal{S}};\Theta')$.
    Meta-optimization: Update Θ

$$\Theta = \Theta - \gamma \frac{\partial(\mathcal{F}(\bar{\mathcal{S}};\Theta) + \beta\mathcal{G}(\breve{\mathcal{S}};\Theta - \alpha\nabla_\Theta))}{\partial\Theta}$$

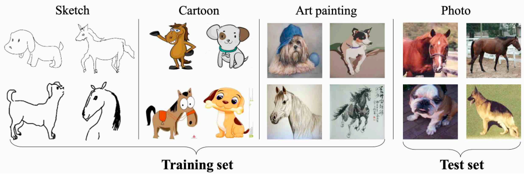
  end for
end procedure

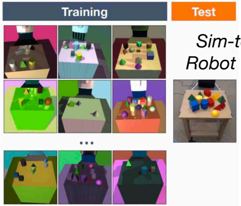
Image credits: [Li et al., 2018]

Applications for DG

Wide applications across CV, NLP, RL, and others.

Image classification

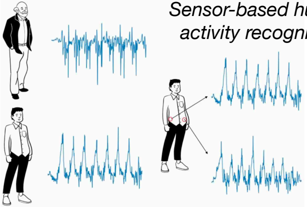




Sim-to-real Robot control

Semantic parsing

Sensor-based human activity recognition



Problems and challenges

Recap: Formal definition of domain generalization

Domain: A domain is composed of data samples that are sampled from a distribution, denoted as $\mathcal{D}^d = \{(X^d, Y^d)\}^{n_d} \sim \mathbb{P}^d(X, Y)$. **Data samples** (X, Y) consists of the input observation X and the corresponding label Y.

Domain generalization (DG): Given M training (source) domains $\mathcal{D}_{\text{train}} = \{\mathcal{D}^i | i = 1, \ldots, M\}$. The goal of DG is to learn a generalizable predictive function $h : \mathcal{X} \to \mathcal{Y}$ from the M training domains to achieve a minimum prediction error on unseen test domains $\mathcal{D}_{\text{test}}$ (i.e., $\mathbb{P}^i(X, Y) \neq \mathbb{P}^{\text{test}}(X, Y)$):

$$\min_h \mathbb{E}_{(X,Y)\in\mathcal{D}_{test}}[\ell(h(X), Y)],$$

where $\mathbb{E}$ is the expectation and $\ell(\cdot, \cdot)$ is the loss function.

DG in time series predictive tasks

A few distinctions from the standard setting:

Data samples: (X, Y) consist of the time series input $X = [x_t]_{t \in S_t}$, where S_t is the set of time steps, and the set of labels $Y = [y_t]_{t \in S_p}$, where $S_p \subseteq S_t$ is the set of labeled time steps.

Two types of domains:

- Source-domain: distribution shifts across data sources.
- Time-domain: distribution shifts over time.

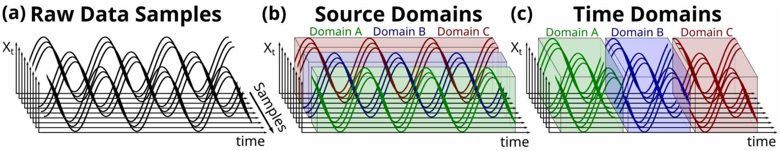
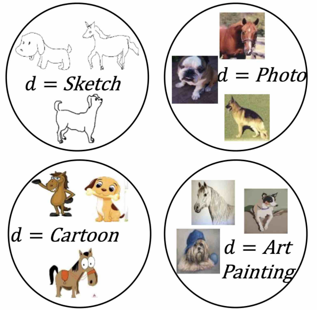


Image credits: [Gagnon-Audet et al., 2022]

Defining domains

- Invariant characteristics should exist across domains for effective generalization.
- Distribution within a time series may shift over time; subdomains may exist.



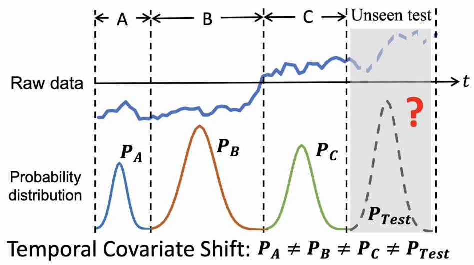
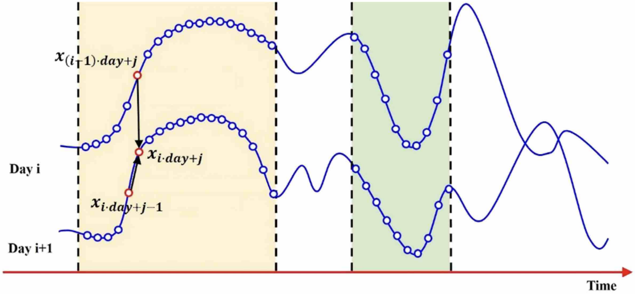


Image credits: [Lu et al., 2022, Du et al., 2021]

DG challenges in time series predictive tasks

Temporal dependencies

- Modeling temporal dependencies while capturing domains' invariant characteristics.

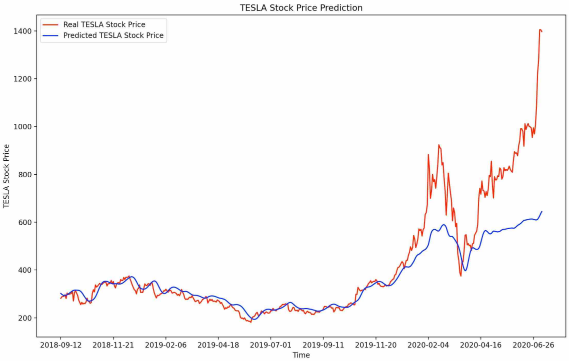


An illustration of daily dependencies of traffic flow time series.

Image credits: [Feng et al., 2018]

Continuous output space

- Dealing with unbounded and potentially infinite output values in forecasting tasks.

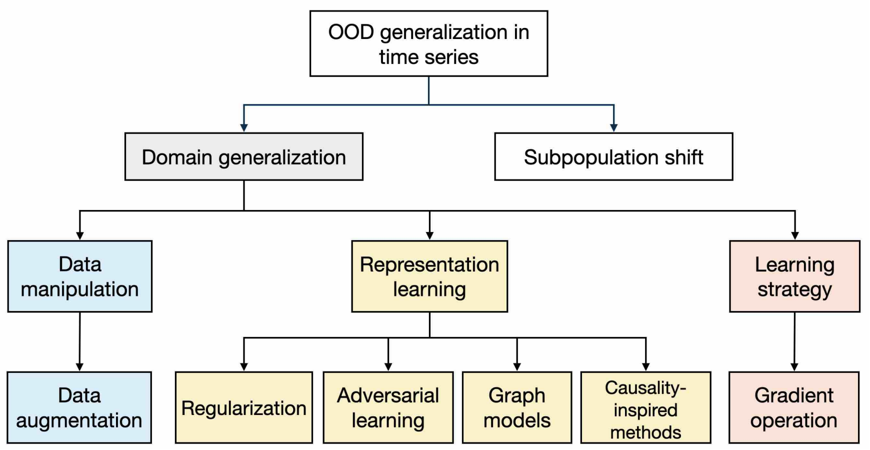


Real and predicted TESLA stock price.

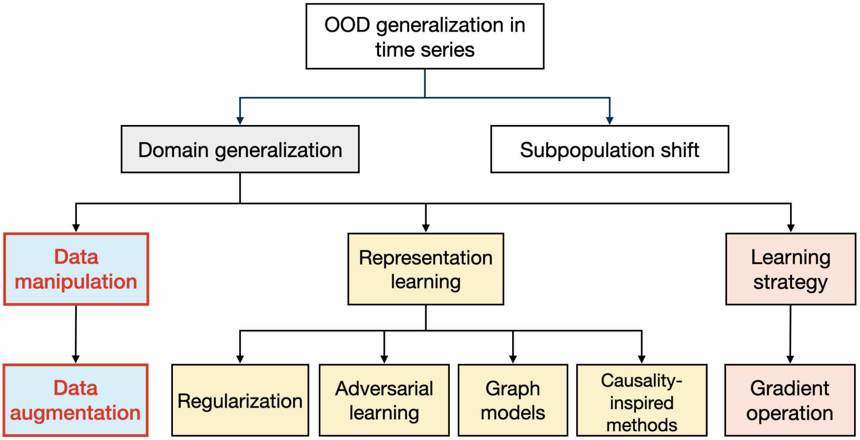
Image credits: Blog of Serafeim Loukas

Methodology

Overview of OOD generalization methodology in time series



Data augmentation



Data augmentation for time series classification

Paper: Data Augmentation for Time Series Classification using Convolutional Neural Networks [Le Guennec et al., 2016]

Two data augmentations are used:

- Window slicing (WS): Divide the time series into slices, each of which is assigned to the same class.
- Window warping (WW): Warp a randomly selected slice of a time series by speeding it up or down.

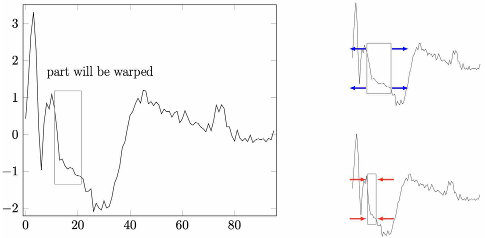
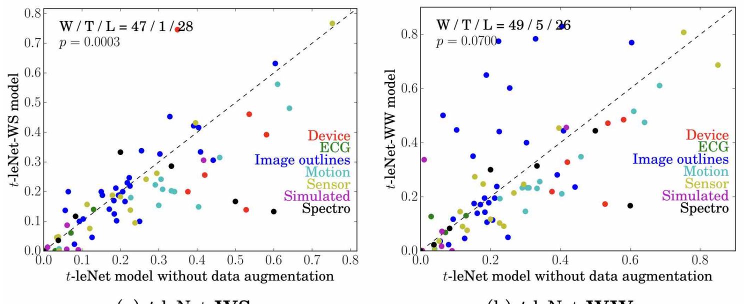


Image credits: [Le Guennec et al., 2016]

Impact of data augmentation on time series classification

Both WS and WW methods help improve classification performance on UCR Archive [Chen et al., 2015].



(a) t-leNet-**WS** (b) t-leNet-**WW**

Both axes correspond to error rates. "W(Win)" means that the y-axis method has lower error rates. T for Tie and L for Lose.

Image credits: [Le Guennec et al., 2016]

Domain-wise time series augmentation

Paper: Domain Generalization via Selective Consistency Regularization for Time Series Classification [Zhang et al., 2022]

- For each source domain, sample an augmentation function from a pre-defined distribution at each iteration. The domain-wise augmentation simulates potential test-time domain shifts.

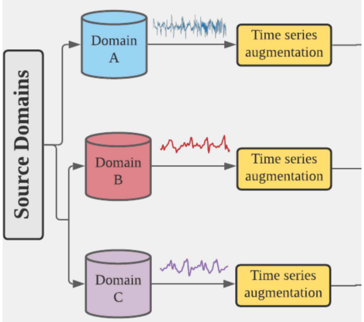


Image credits: [Zhang et al., 2022]

Time series augmentation methods

Three time series augmentation methods are considered in this work.

Augmentation	General Expression
mean shift	$a_{mean}(x) = x - \mu + \mu_{new}$
scaling	$a_{scale}(x) = \left(\frac{x-\mu}{\sigma}\right) * \sigma_{new} + \mu$
masking	$a_{mask}(x[i]) = \begin{cases} x[i] & \text{w.p. } 0.9 \\ \mu & \text{w.p. } 0.1 \end{cases}$

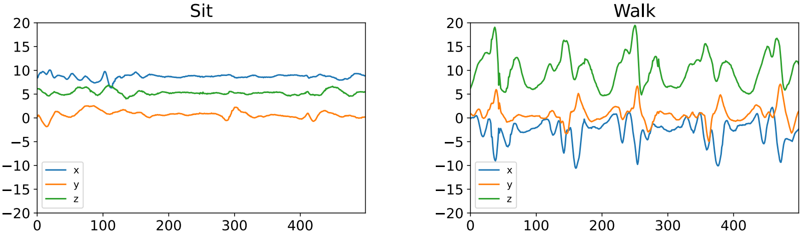
Applying data augmentations improves model performance on the Bearings (Detect bearings faults in rotating machines) dataset.

Aug	Avg Acc (%)
None	82.2
Mean shift	83.0
Scale	82.4
Mask	82.4
All	86.5

The choice of augmentations methods

The choice of augmentations depends on the dataset to avoid perturbing characteristics known to be important for classification.

On HHAR (Heterogeneity human activity recognition) dataset, limited augmentation, i.e., scaling with $\mu = 0, \sigma = 1$ and $\sigma_{new} \sim Unif(0.8, 1.2)$, is applied since mean and standard deviation are key classification features.



Accelerometer time series plots (for each axis) of a static activity "Sit" and a dynamic activity "Walk".

Summary of data augmentation

Limited data augmentation research in DG for time series tasks.

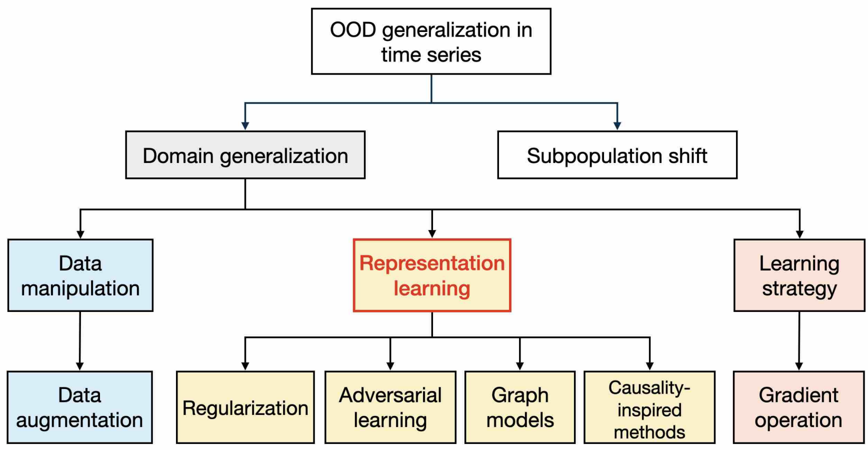
Advantages

- Increase data quantity
- Easy to understand and simple to implement

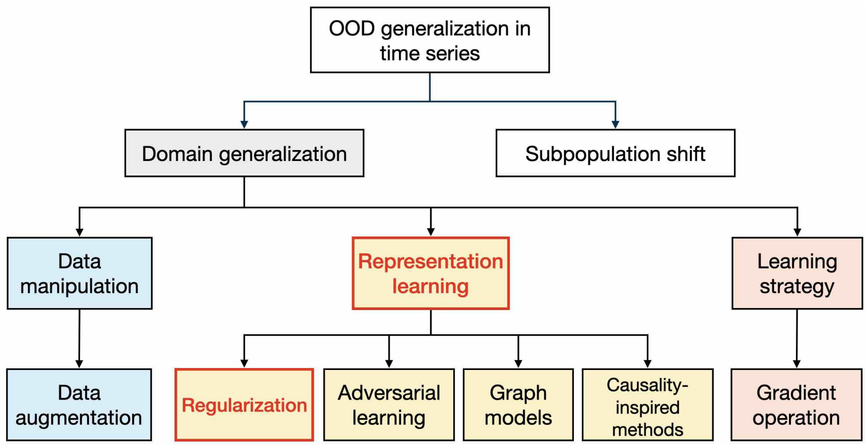
Disadvantages

- Lack of theoretical guarantee

Representation Learning



Regularization



Regularization

These methods introduce regularization terms into the model's objective function to enhance domain generalization by learning better representations, e.g., domain-invariant representations.

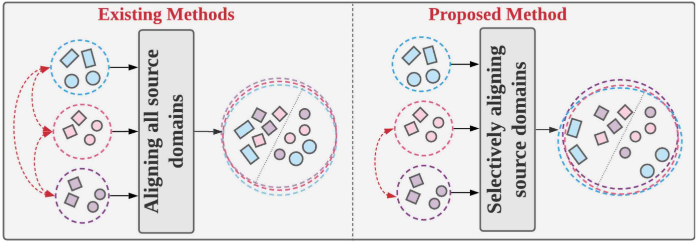
The overall objective can be expressed as:

$$L_{obj} = L_{model} + \lambda L_{reg}$$

Note that L_{reg} does not mean L1/L2 norm that prevent overfitting in general.

Selective cross-domain consistency regularization

Paper: Domain Generalization via Selective Consistency Regularization for Time Series Classification [Zhang et al., 2022]



Learn model parameters such that the class conditional distribution is invariant for closely related domains according to latent inter-domain relationships.

Selective cross-domain consistency regularization

Impose greater regularization on more similar domains:

$$L_{sel} = \sum_{i,j}^{M} \underbrace{w(D^i, D^j)}_{\text{Domain similarity}} \sum_{l=1}^{L} \underbrace{||\bar{\mathbf{g}}^{D^i,l} - \bar{\mathbf{g}}^{D^j,l}||_2^2}_{\text{Difference of mean logit vectors of domains for class } l}$$

where $\bar{\mathbf{g}}^{D^i,l}$ is the mean logit vector for domain D^i class l, referred to as the class-conditional domain centroid.

Defining domain similarity – metadata based similarity

Domain metadata (i.e., descriptions of source domain data) is available:

- Use metadata to infer relationships by grouping the domains into clusters.
- Only domains within a cluster are assumed to share class relationships and are subject to regularization.

$$L_{sel}^{\text{meta}} = \sum_{c=1}^{K} \sum_{D^i \in \mathcal{S}_c} \sum_{l}^{L} ||\bar{\mathbf{g}}^{D^i,l} - \bar{\gamma}^{c,l}||_2^2$$

where $\mathcal{S}_c$ is the set of domains in cluster c. $\gamma^{c,l}$ is the mean logit vector for domain cluster c class l, denoted class-conditional cluster centroid.

Defining domain similarity – learned similarity

Domain metadata is not available:

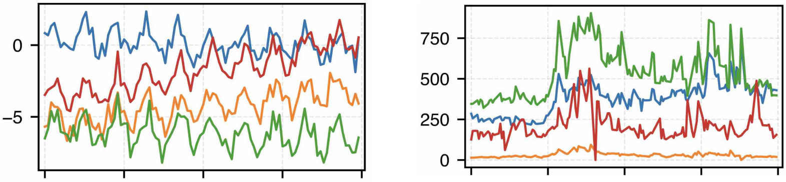
- Measure domain distance using the squared L2 distance of their class-conditional domain centroids.
- Regularization applies to each domain and its nearest neighbor domain.
  - RBF kernel is applied on the inter-domain distance with hyperparameter ξ.

$$w_{\text{learned}}(D^i, D^j) = \begin{cases} \frac{1}{L}\sum_{l=1}^{L} \exp\left(\frac{-\|\bar{\mathbf{g}}^{D^i,l} - \bar{\mathbf{g}}^{D^j,l}\|_2^2}{2\xi^2}\right), & d^j \text{ is nearest to } d^i \text{ for most classes} \\ 0, & \textit{Otherwise} \end{cases}$$

Cross-domain regularizations with difficulty awareness

Paper: Domain Generalization in Time Series Forecasting [Deng et al., 2024]

This work focuses on the scenario where time series domains share certain common attributes (e.g., same seasonality and trend) and exhibit no abrupt distribution shifts within a single domain.



Propose the domain discrepancy regularization, and an extended version by incorporating a notion of domain difficulty awareness (named CEDAR).

Image credits: [Deng et al., 2024]

Domain discrepancy regularization

Dissimilar training domains should not exhibit significant variations in forecasting performance:

$$L_{DD} = \sum_{i,j}^{M} \underbrace{d_{\mathcal{H}}\left(D^i, D^j\right)}_{\text{Distribution divergence}} \cdot \underbrace{d_{\mathcal{L}_{\text{fcst}}}\left(D^i, D^j\right)}_{\text{Difference in mean forecasting performance}}$$

where $d_{\mathcal{H}}(,)$ calculates the discrepancy of high-level representation of two domains (e.g., RNN hidden states). $d_{\mathcal{L}_{\text{fcst}}}(,)$ computes the Euclidean distance between two domain-averaged losses.

The regularization term aims to prevent severe overfitting in all source domains.

Domain discrepancy regularization with domain difficulty awareness

A scaling factor is introduced to adjust the penalty to account for the difficulty of the domains:

$$L_{DDD} = \sum_{i,j}^{M} d_{\mathcal{H}}\left(D^i, D^j\right) \cdot d_{\mathcal{L}_{\text{fcst}}}\left(D^i, D^j\right) \cdot \underbrace{\omega(D^i, D^j)}_{\text{A scaling factor that modulates the penalty}}$$

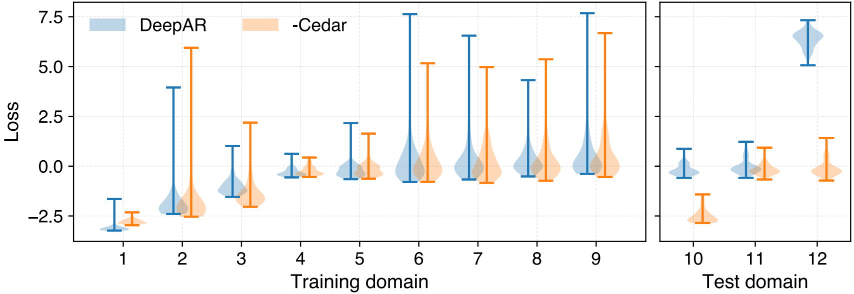
The scaling factor is based on standard deviations of losses:

$$\omega(D^i, D^j) = \frac{1}{Std\left(\mathcal{L}_{\text{fcst}}(D^i)\right) + Std\left(\mathcal{L}_{\text{fcst}}(D^j)\right) + \varepsilon}$$

Higher loss variance implies greater challenges in training. A smaller penalty is applied to that domain, allowing the model more flexibility to learn from its data.

Domain performance analysis

CEDAR achieves more even loss distributions for some training domains (e.g., 6–9), which denotes less underfitting and overfitting. CEDAR also shows notable performance improvements across all test domains.



Forecasting performance by domains of the base model and CEDAR on Stock-volume.

Image credits: [Deng et al., 2024]

Such DG methods might not be useful for non-stationary time series, because:

- Complex distributions exist within a time series, i.e., it contains many unknown sub-distributions.

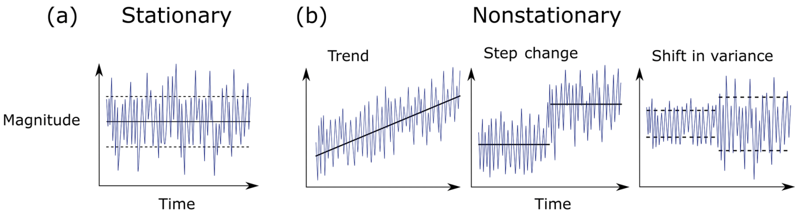
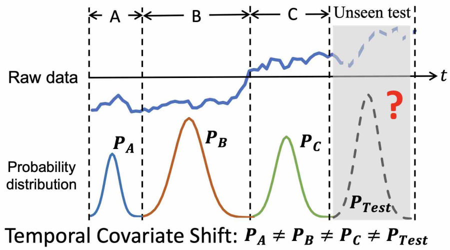


Image credits: Slater et al. [2021]

Adaptive learning and forecasting for time series

Paper: *AdaRNN: Adaptive Learning and Forecasting for Time Series* [Du et al., 2021]

A two-stage approach AdaRNN is proposed to generalize non-stationary time series.



1. Temporal distribution characterization segments time series into multiple domains.
2. Temporal distribution matching matches distribution gaps of domains.

Image credits: Du et al. [2021]

Temporal distribution characterization (TDC)

Identify the most distinct periods/domains within a time series, which represents the worst case of temporal covariate shift since the cross-domain distributions are the most diverse.

Solve an optimization problem:

$$\max \frac{1}{K} \sum_{i,j}^{K} d(D^i, D^j)$$

where $d(,)$ can be any distance function.

A greedy algorithm is used.

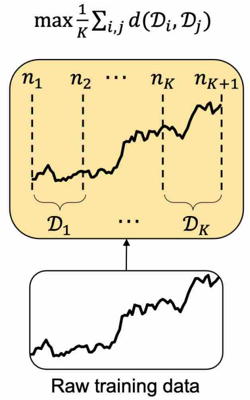


Image credits: [Du et al., 2021]

Temporal distribution matching (TDM)

Once the time domains are obtained, learn common knowledge shared by different domains via matching their distributions.

Given a domain-pair (D^i, D^j), the loss of TDM is formulated as:

$$L_{\text{tdm}}(D^i, D^j; \theta, \boldsymbol{\alpha}) = \sum_{t=1}^{T} \alpha_{i,j}^t d(\mathbf{h}_i^t, \mathbf{h}_j^t; \theta)$$

where $\alpha_{i,j}^t$ denotes the distribution importance between D^i and D^j at t.

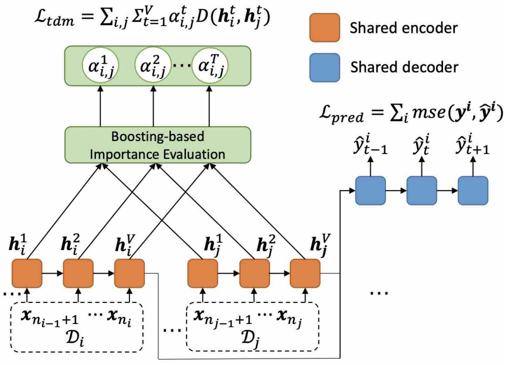


Image credits: [Du et al., 2021]

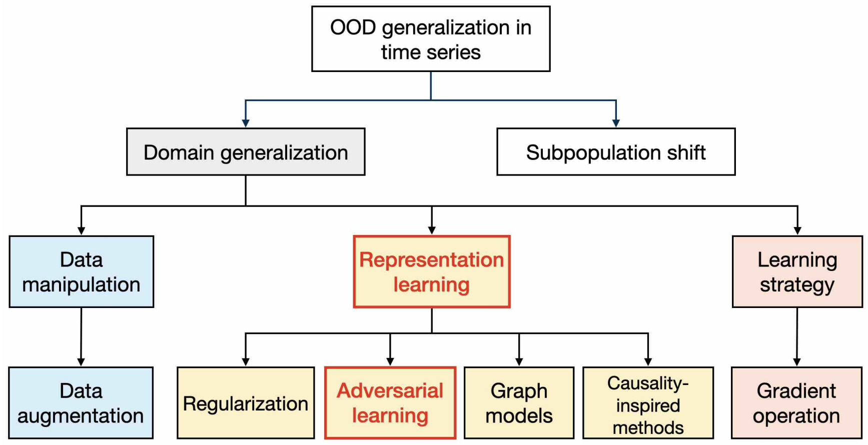
Temporal distribution matching (TDM)

The final objective (one RNN layer) is:

$$L(\theta, \boldsymbol{\alpha}) = L_{\text{pred}}(\theta) + \lambda \frac{2}{K(K-1)} \sum_{i,j}^{K} L_{\text{tdm}}(D^i, D^j; \theta, \boldsymbol{\alpha})$$

where $\boldsymbol{\alpha}$ is leaned through a boosting-based importance evaluation algorithm.

Adversarial learning



Adversarial learning

Adversarial learning is a technique used in machine learning to fool a model with malicious input.

In DG, adversarial learning is designed to learn representations that are invariant to domain variations.

- E.g., a discriminator is trained to identify different domains, while a generator is simultaneously trained to fool the discriminator, leading to domain-agnostic features [Ganin and Lempitsky, 2015].

Mostly studied in classification tasks.

Out-of-distribution representation learning for time series classification

Paper: Out-of-Distribution Representation Learning for Time Series Classification [Lu et al., 2022]

Propose an end-to-end approach, DIVERSIFY incorporating adversarial learning for out-of-distribution representation learning on non-stationary times series.

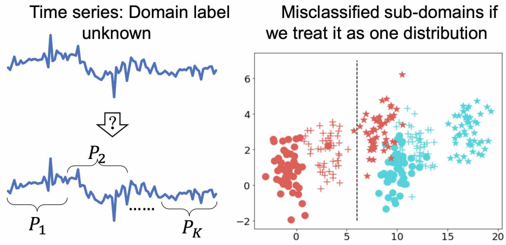


Image credits: [Lu et al., 2022]

The framework of DIVERSIFY

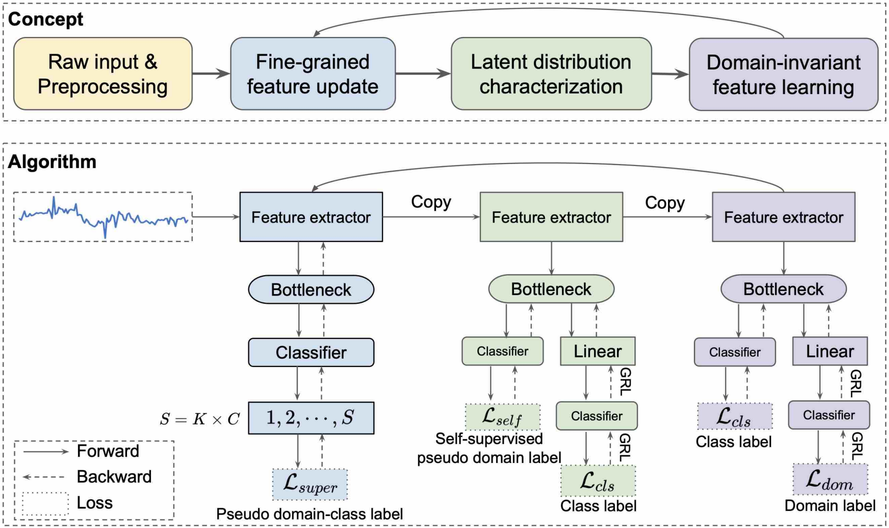


Image credits: [Lu et al., 2022]

Fine-grained feature update

Propose pseudo domain-class label to supervise a feature extractor. Features are more fine-grained w.r.t. domains and labels.

The supervised loss is:

$$L_{super} = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}^{tr}}(h_c(h_{bf}(\mathbf{x})), s)$$

Treat per category per domain as a *new class* with label $s \in \{1, 2, \ldots, S = K \times C\}$. K is the number of latent distributions/domains and C is the number of labels. $s = d' \times C + y$ where d' is the domain label initialized to 0.

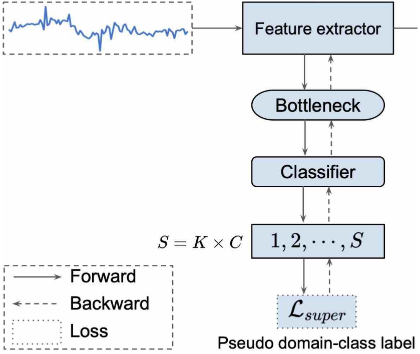


Image credits: [Lu et al., 2022]

Latent distribution characterization

Employ a self-supervised pseudo-labeling strategy to obtain domain labels.

1. Obtain the initial centroid for each (latent) domain:

$$\tilde{\mu}_k = \frac{\sum_{\mathbf{x}_i \in X^{tr}} \delta_k(h_c(h_{bf}(\mathbf{x}_i))) \cdot h_{bf}(\mathbf{x}_i)}{\sum_{\mathbf{x}_i \in X^{tr}} \delta_k(h_c(h_{bf}(\mathbf{x}_i)))}$$

   where δ_k is the k^{th} element of the logit softmax output.

2. Obtain the pseudo domain labels according to the nearest centroid:

$$\tilde{d}'_i = \text{argmin}_k Dis(h_{bf}(\mathbf{x}_i), \tilde{\mu}_k)$$

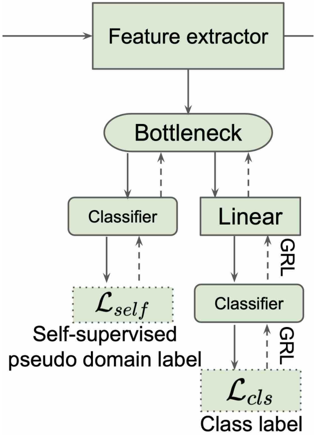


Image credits: [Lu et al., 2022]

3. Compute the centroids again and obtain the updated pseudo domain labels.

$$\mu_k = \frac{\sum_{\mathbf{x}_i \in X^{tr}} \mathbb{1}(\tilde{d'}_i = k) \cdot h_{bf}(\mathbf{x}_i)}{\sum_{\mathbf{x}_i \in X^{tr}} \mathbb{1}(\tilde{d'}_i = k)}$$

$$d'_i = \mathrm{argmin}_k Dis(h_{bf}(\mathbf{x}_i), \mu_k)$$

4. Compute the self-supervised pseudo domain loss L_{self} and the classification loss L_{cls}.

   Use adversarial training, i.e., gradient reversal layer (GRL) [Ganin and Lempitsky, 2015] to learn features for classifying domains that disregard class information.

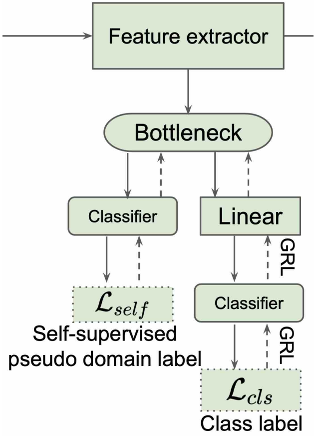
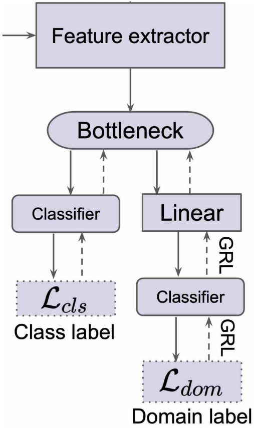


Image credits: [Lu et al., 2022]

Domain-invariant representation learning

Given the learned domain labels, update the classification loss L_{cls} and domain classifier loss L_{dom} using adversarial training.

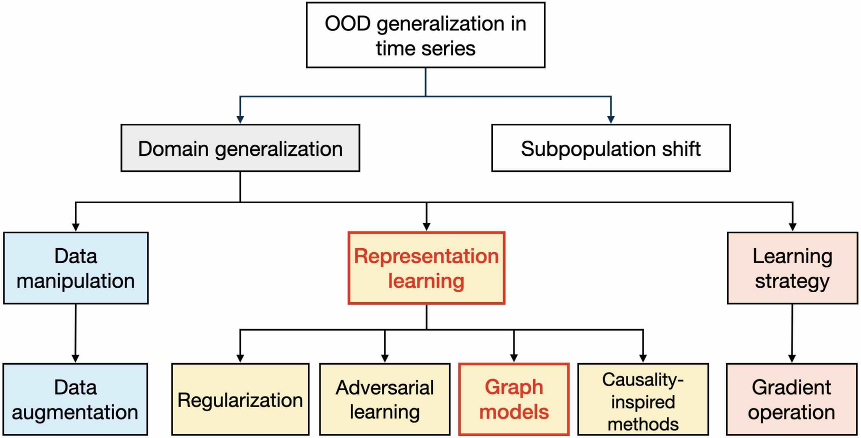
The gradient reversal layers help learn key features for classification while eliminating domain-specific information.



Repeat these steps until convergence or max epochs.

Image credits: [Lu et al., 2022]

Graph model

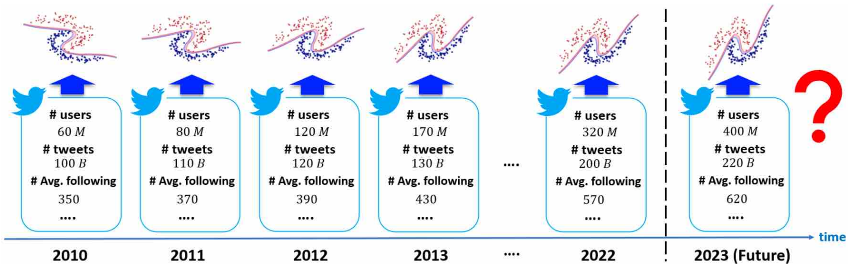


OOD generalization in time series
Domain generalization
Subpopulation shift
Data manipulation
Representation learning
Learning strategy
Data augmentation
Regularization
Adversarial learning
Graph models
Causality-inspired methods
Gradient operation

Temporal domain generalization with drift-aware dynamic neural network

Paper: Temporal Domain Generalization with Drift-Aware Dynamic Neural Networks [Bai et al., 2022]

Build a time-sensitive model, DRAIN, using dynamic neural networks to achieve temporal domain generalization.

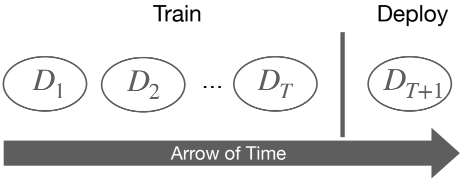


An illustrative example of *temporal domain generalization*.

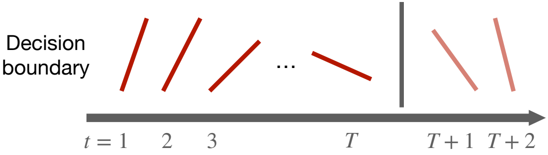
Image credits: [Bai et al., 2022]

Problem formulation – temporal domain generalization

Given source/training domains $D_1, D_2, \ldots D_T$ where we assume the distribution of $D_t, t = 1, 2, \ldots T$ evolves over time and temporal drift across time is not too high.



The goal is to infer the shifting decision boundary and extrapolate it to target domain D_{T+1} in the immediate future.



A probabilistic view of concept drift in temporal domain generalization

Propose a Bayesian framework to characterize the temporal data distribution drift and its influence on models, namely $P(\mathbf{w}_t|D_t)$.

Predict $\mathbf{w}_{T+1}$ given all training data $D_{1:T}$:

$$P(\mathbf{w}_{T+1}|D_{1:T}) = \int_{\Omega} \underbrace{P(\mathbf{w}_{T+1}|\mathbf{w}_{1:T}, D_{1:T})}_{\text{inference}} \cdot \underbrace{P(\mathbf{w}_{1:T}|D_{1:T})}_{\text{training}} d_{\mathbf{w}_{1:T}}$$

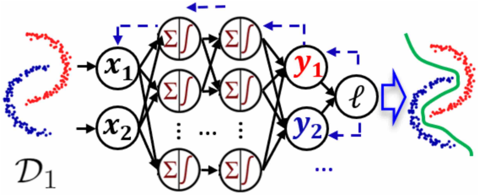
Decompose the training phase:

$$\begin{aligned} P(\mathbf{w}_{1:T}|D_{1:T}) &= \prod_{s=1}^{T} P(\mathbf{w}_s|\mathbf{w}_{s-1}, D_{1:T}) \\ &= P(\mathbf{w}_1|D_1) \cdot P(\mathbf{w}_2|\mathbf{w}_1, D_{1:2}) \cdots P(\mathbf{w}_T|\mathbf{w}_{1:T-1}, D_{1:T}) \end{aligned}$$

Neural network with dynamic parameters

Treat the time-evolving model parameters $\mathbf{w}_t$ as a dynamic graph to achieve a fully time-sensitive model.

Use an edge-weighted graph $G = (V, E, \mathbf{w})$ to represent a neural network. $\mathbf{w}$ represents the entire set of parameters for the neural network.



Assume the topology of the neural network is given, i.e., V, E are fixed and $\mathbf{w}$ is changing w.r.t time.

Image credits: [Bai et al., 2022]

The framework of DRAIN

Leverage the sequential model to learn the temporal drift adaptively and to predict the model parameters on the future domain.

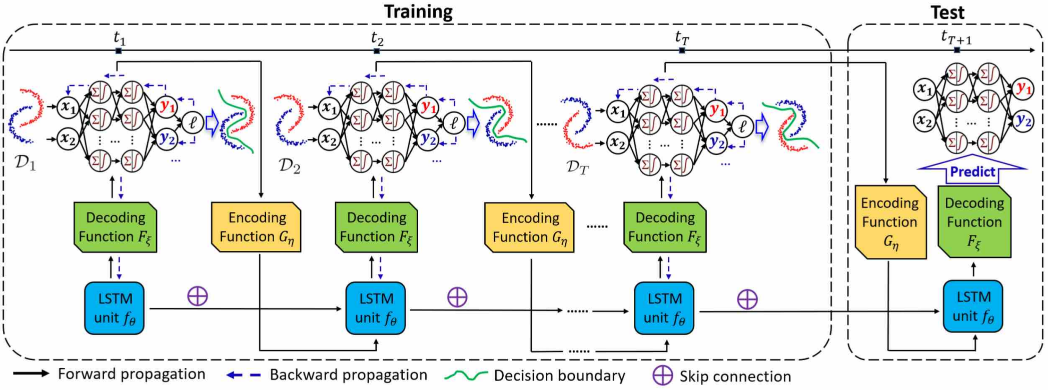
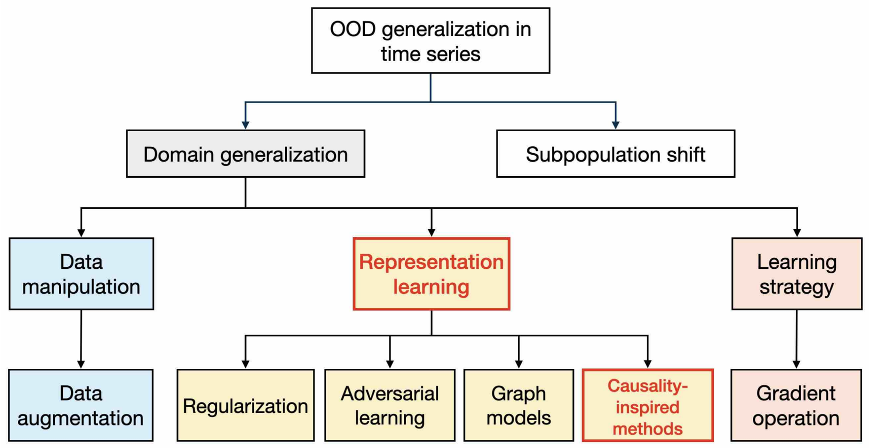


Image credits: [Bai et al., 2022]

Causality-inspired method



Causality

Causality is a relationship between two events, in which one event causes an effect on the other event.

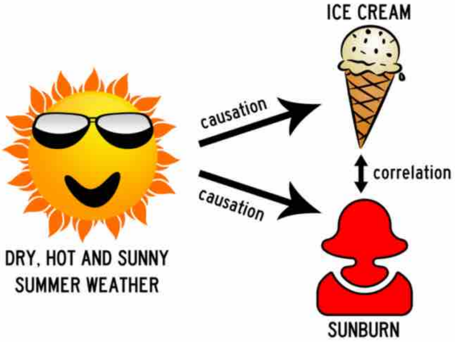


Image credits: Blog of Luke Worthington

Paper: Causal-based Time Series Domain Generalization for Vehicle Intention Prediction [Hu et al., 2022]

Propose the Causal-based Time Series Domain Generalization (CTSDG) model, which constructs a structural causal model for vehicle intention prediction (i.e., predict interaction outcomes such as pass/yield).

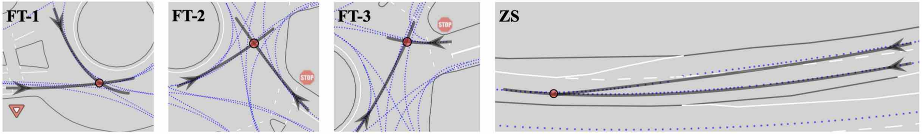
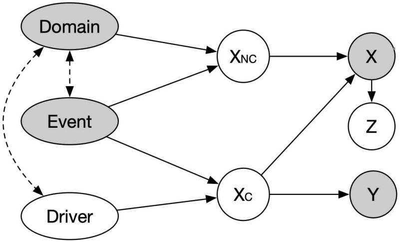


Illustration of selected domains for driving scenarios. Black arrow line ($\rightarrow$) represents a reference path and red circles ($\bullet$) are intersecting points.

Image credits: [Hu et al., 2022]

The framework of CTSDG

A causal view of data generating process under vehicle interaction settings.

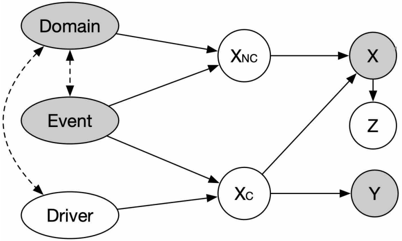


Shaded/transparent nodes are observed/latent variables. Directed edge denotes a causal relationship. Dashed edges denote correlation.

Image credits: [Hu et al., 2022]

- Domain (D): map properties, e.g., road topology, speed limit, and traffic rules.
- Event (E): two-vehicle interactions, e.g., initial states and the length of interaction
- Driver (O): driver's driving preferences
- X: vehicle interactive trajectories; multivariate time series
- Z: latent representations
- Y: vehicle intention label
- X_C/X_{NC}: causal/non-causal features

Invariance condition



Shaded/transparent nodes are observed/latent variables. Directed edge denotes a causal relationship. Dashed edges denote correlation.

According to the causal framework, X_C causes Y, and by d-separation, we have $Y \perp\!\!\!\perp D | X_C$.

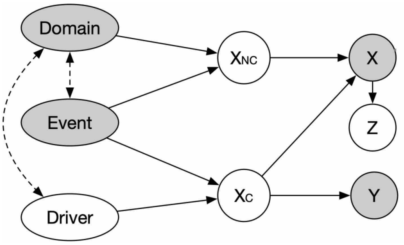
Learn a $q(\cdot)$ maps X to Z, $\phi(\cdot)$ maps Z to X_C and a classifier $h(\cdot)$ maps X_C to Y.

Minimize the prediction loss:

$$L_{clf} = Loss(h(\phi(q(X))), Y)$$

Image credits: [Hu et al., 2022]

Invariance condition



Shaded/transparent nodes are observed/latent variables. Directed edge denotes a causal relationship. Dashed edges denote correlation.

By d-separation, X_C also needs to satisfy the invariance condition $X_C \perp\!\!\!\perp D|\{E, O\}$.

However, O is unobservable and there may not be a same E across domains.

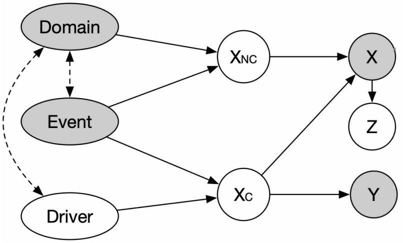
Instead, assume that the distance over X_C between same-class inputs from different domains is bounded. Minimize the distance:

$$L_{dis} = \sum_{\Omega(\mathbf{x}_i,\mathbf{x}_j)=1, i\neq j} Dis(\phi(q(\mathbf{x}_i)), \phi(q(\mathbf{x}_j)))$$

where $\Omega : X \times X \to \{0, 1\}$ is a match function. $\Omega(\mathbf{x}_i, \mathbf{x}_j) = 1$ denotes same-class inputs from different domains.

Capturing temporal latent dependencies

Remember that $q(\cdot)$ is a function maps X to Z. Given $Z \perp\!\!\!\perp D|X$, by learning $q(\cdot)$, we can extract a domain-invariant latent variable that represents the input space.



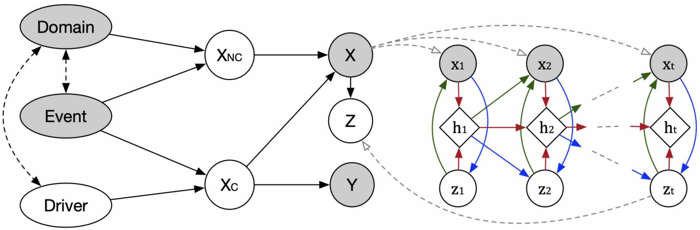
Since X is time series data, the learned Z should capture temporal latent information.

Image credits: [Hu et al., 2022]

Capturing temporal latent dependencies

Variational Recurrent Neural Networks (VRNN) [Chung et al., 2015] is used to model the dependencies between latent random variables across time steps, and $q(\cdot)$.

The VRNN contains a Variational Autoencoder (VAE) [Kingma and Welling, 2013] at every time step and these VAEs are conditioned on previous auto-encoders via the hidden states of an RNN.



Green lines: generation process; blue lines: inference process; red lines: recurrence process

Image credits: [Hu et al., 2022]

Overall algorithm

The complete objective function to minimize:

$$L_{clf} + \gamma L_{dis} + \lambda L_{temp}$$

where L_{dis} denotes the distance over X_C between same-class inputs from different domains. L_{temp} is the the objective function for the VRNN.

Summary of representation learning

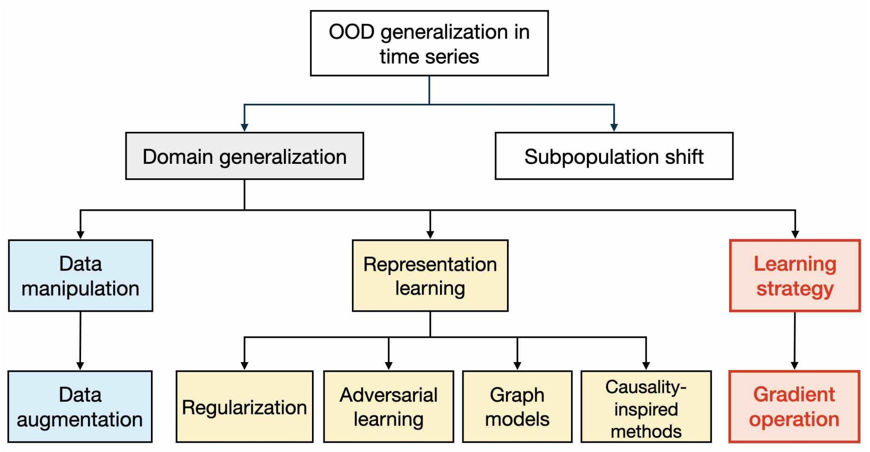
Advantages

- General and popular
- Better performance
- Some theoretical guarantee

Disadvantages

- Still difficult to remove spurious features
- Data-driven

Gradient operation



Gradient operation

Gradient operation approaches optimize machine learning models by adjusting their parameters to minimize the loss function.

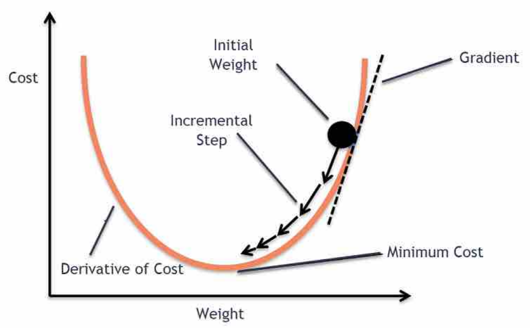
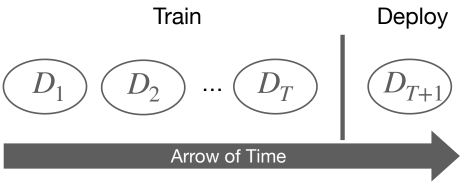


Image credits: Clairvoyant Blog

Gradient interpolation loss to generalize along time

Paper: Training for the Future: A Simple Gradient Interpolation Loss to Generalize Along Time [Nasery et al., 2021]

Introduce a Gradient Interpolation (GI) approach for temporal domain generalization.



The approach includes a time sensitive network and imposes a special loss to encourage the network to generalize to the near future.

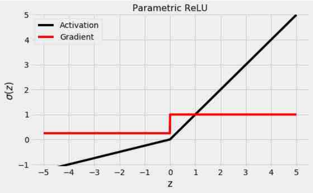
Time sensitive network $F_\theta(\mathbf{x}, t)$

Use Time2Vec (t2v) [Kazemi et al., 2019] to capture complex dependencies such as periodicity.

$$\tau_t[i] = \begin{cases} w_i t + b_i, & 1 \leq i \leq m_p \\ \sin(w_i t + b_i), & m_p \leq i \leq m \end{cases}$$

Introduce a novel time dependent leaky ReLU (TReLU)

whose threshold and slop are affected by time.



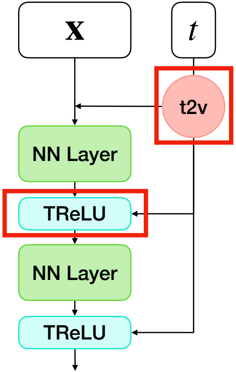


Image credits: slideslive

Gradient interpolation

Despite using a time-sensitive artitecture, ERM may overfit on $D_1, \dots D_T$, since there is no relation or constraint between the prediction of the network on different timestamps.

GI loss:

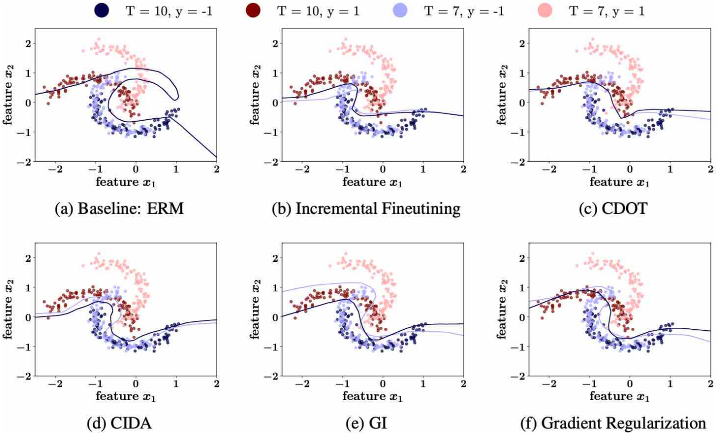
$$\underbrace{L(y; F_\theta(\mathbf{x}, t))}_{\text{Pred loss}} + \lambda \max_{\delta \in (-\Delta, \Delta)} \underbrace{L(y; F_\theta(\mathbf{x}, t-\delta) + \delta \frac{\partial F_\theta(\mathbf{x}, t-\delta)}{\partial t})}_{\text{Pred loss on interpolated logits}}$$

The second term is the loss on a regularized approximation of $F_\theta(\mathbf{x}, t)$ using the first-order Taylor Expansion at $t - \delta$. It provides "supervision" on nearby time steps and encourages smoother functions.

δ is adversarially chosen by gradient ascent within a user-provided window Δ.

A negative δ encourages extrapolation from the future.

Qualitative analysis on 2-moons



(a) Baseline: ERM (b) Incremental Fineutining (c) CDOT

(d) CIDA (e) GI (f) Gradient Regularization

Image credits: [Nasery et al., 2021]

GI learns a more accurate decision boundary, which rotates correctly along time.

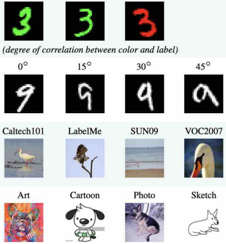
Datasets, benchmarks and evaluation

Benchmarks for OOD generalization

Two popular benchmarks for OOD generalization:

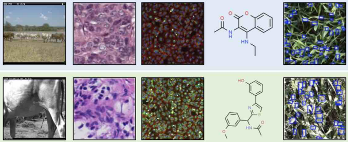
Dataset	Domains			
Colored MNIST	+90%	+80%	-90%	
Rotated MNIST	0°	15°	30°	45°
VLCS	Caltech101	LabelMe	SUN09	VOC2007
PACS	Art	Cartoon	Photo	Sketch

(degree of correlation between color and label)



(a) DomainBed [Gulrajani and Lopez-Paz, 2020]

Dataset	iWildCam	Camelyon17	RxRx1	OGB-MolPCBA	GlobalWheat
Input (x)	camera trap photo	tissue slide	cell image	molecular graph	wheat image
Prediction (y)	animal species	tumor	perturbed gene	bioassays	wheat head bbo
Domain (d)	camera	hospital	batch	scaffold	location, time
# domains	323	5	51	120,084	47
# examples	203,029	455,954	125,510	437,929	6,515
Train example					
Test example					
Adapted from	Beery et al. 2020	Bandi et al. 2018	Taylor et al. 2019	Hu et al. 2020	David et al. 2021



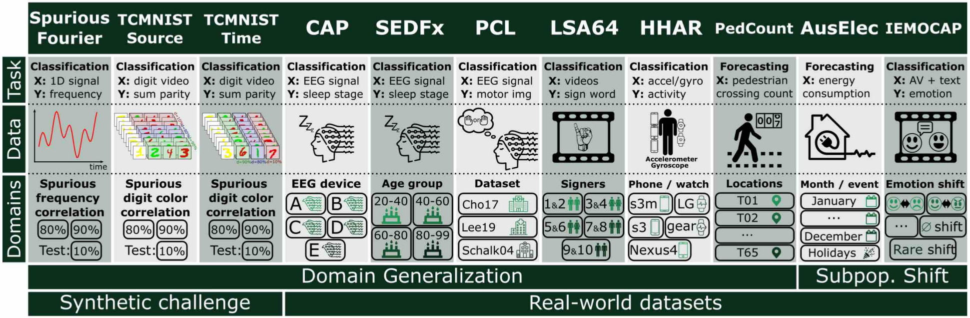
(b) WILDS [Koh et al., 2021]

They focus on image datasets.

Image credits: [Gulrajani and Lopez-Paz, 2020, Koh et al., 2021]

A benchmark for OOD generalization in time series

WOODS [Gagnon-Audet et al., 2022] is a benchmark of 3 synthetic and 8 real-world time series datasets spanning a wide array of critical problems and data modalities, such as videos, brain recordings, etc.



https://woods-benchmarks.github.io/auselec.html

Image credits: [Gagnon-Audet et al., 2022]

OOD generalization algorithms for time series

The framework includes adaptation of existing OOD generalization algorithms for time series datasets.

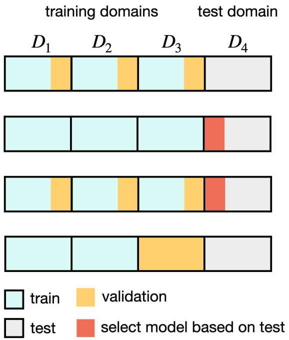
- Empirical Risk Minimization (ERM)
- Invariant Risk Minimization (IRM)
- Group Distributionally Robust Optimization (GroupDRO)
- ...
- DIVERSIFY [Lu et al., 2022]

Some methods are agnostic to data and tasks, and some are only applicable for classification tasks.

Model selection

For DG in time series

- **Train-domain validation**: Choose the model that gets the best average validation performance across training domains.
- **Test-domain validation**: Choose the model with the best performance on the test domain. No early stopping.
- **Oracle train-domain validation**: Choose the model with the best performance on the test domain. During training, the validation is done on training domains.
- **Leave-one-domain-out cross-validation** [Gulrajani and Lopez-Paz, 2020]: Train each model while holding one of the training domains as validation set. Choose the model maximizing this average accuracy, retrained on all training domains.



Experimental findings of WOODS

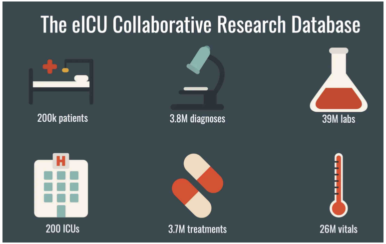
- WOODS datasets have a significant generalization gap

Dataset (Perf. is accuracy unless specified)	Performance ID	Performance OOD	**Gap**
Spur.-Fourier	74.5 (0.1)	9.8 (0.2)	64.7
TCM.-Source	68.4 (0.1)	10.2 (0.1)	58.2
	...		
AusElec (rmse)	232.0 (2.6)	397.2 (8.4)	165.2
IEMOCAP	69.1 (0.4)	57.7 (1.9)	11.4

- Marginal improvement over ERM on WOODS real-world datasets on average
- Algorithms fail on synthetic datasets

More datasets

Healthcare: eICU collaborative research database [Pollard et al., 2018] is a freely available multi-center database for critical care research.



Retail: Favorita [Mendoza Calero, 2018] comprises grocery sales data from Corporación Favorita.

Environmental monitoring: Air-quality dataset [Zhang et al., 2017] contains hourly air quality information collected from 12 stations in Beijing.

Image credits: Blog of Toby Manders

Summary, future directions and discussion

Conclusion

Motivation, background, problems and challenges of OOD generalization in time series

Methodology:

- Data manipulation: Data augmentation
- Representation learning
  - Regularization, adversarial learning, graph models, causality-inspired method
- Learning strategy: Gradient operation

Datasets, benchmarks and evaluation

Future directions

Interpretable OOD generalization in time series

- Learning to interpret: why it can generalize?

Ethical and fair AI

- Ensure models are fair and unbiased, especially in critical applications like healthcare.
- Develop fairer evaluation standards.

Sustainability and scalability

- Computational efficiency in model training and execution for large-scale time series data.

Q&A

Thank You!

Questions, comments, . . .

References i

G. Bai, C. Ling, and L. Zhao. Temporal domain generalization with drift-aware dynamic neural networks. *arXiv preprint arXiv:2205.10664*, 2022.

Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. The ucr time series classification archive, July 2015. www.cs.ucr.edu/~eamonn/time_series_data/.

J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28, 2015.

S. Deng, O. Sprangers, M. Li, S. Schelter, and M. de Rijke. Domain generalization in time series forecasting. *ACM Trans. Knowl. Discov. Data*, jan 2024. ISSN 1556-4681. doi: 10.1145/3643035. URL https://doi.org/10.1145/3643035.

Y. Du, J. Wang, W. Feng, S. Pan, T. Qin, R. Xu, and C. Wang. Adarnn: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 402–411, 2021.

S. Feng, X. Wang, H. Sun, Y. Zhang, and L. Li. A better understanding of long-range temporal dependence of traffic flow time series. *Physica A: Statistical Mechanics and its Applications*, 492: 639–650, 2018.

J.-C. Gagnon-Audet, K. Ahuja, M.-J. Darvishi-Bayazi, P. Mousavi, G. Dumas, and I. Rish. Woods: Benchmarks for out-of-distribution generalization in time series. *arXiv preprint arXiv:2203.09978*, 2022.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Y. Hu, X. Jia, M. Tomizuka, and W. Zhan. Causal-based time series domain generalization for vehicle intention prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7806–7813. IEEE, 2022.

S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. Brubaker. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*, 2019.

J. Kim and J.-M. Kim. Bearing fault diagnosis using grad-cam and acoustic emission signals. *Applied Sciences*, 10(6):2050, 2020.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

A. Le Guennec, S. Malinowski, and R. Tavenard. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*, 2016.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

W. Lu, J. Wang, X. Sun, Y. Chen, and X. Xie. Out-of-distribution representation learning for time series classification. In *The Eleventh International Conference on Learning Representations*, 2022.

A. Maurya, R. K. Yadav, M. Kumar, and Saumya. Comparative study of human activity recognition on sensory data using machine learning and deep learning. In *Proceedings of Integrated Intelligence Enable Networks and Computing: IIENC 2020*, pages 63–71. Springer, 2021.

A. S. Mendoza Calero. Corporación favorita grocery sales forecasting kaggle competition. Master's thesis, Universidad Internacional de Andalucía, 2018.

A. Nasery, S. Thakur, V. Piratla, A. De, and S. Sarawagi. Training for the future: A simple gradient interpolation loss to generalize along time. *Advances in Neural Information Processing Systems*, 34: 19198–19209, 2021.

T. J. Pollard, A. E. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):1–13, 2018.

H. Qian, S. J. Pan, and C. Miao. Latent independent excitation for generalizable sensor-based cross-person activity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11921–11929, 2021.

Y. Shi, J. Seely, P. H. Torr, N. Siddharth, A. Hannun, N. Usunier, and G. Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

L. J. Slater, B. Anderson, M. Buechel, S. Dadson, S. Han, S. Harrigan, T. Kelder, K. Kowal, T. Lees, T. Matthews, et al. Nonstationary weather and water extremes: a review of methods for their detection, attribution, and management. *Hydrology and Earth System Sciences*, 25(7):3897–3935, 2021.

J. Sun, D. Sow, J. Hu, and S. Ebadollahi. Localized supervised metric learning on temporal physiological data. In *2010 20th International Conference on Pattern Recognition*, pages 4149–4152. IEEE, 2010.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

W. Wah, S. Das, A. Earnest, L. K. Y. Lim, C. B. E. Chee, A. R. Cook, Y. T. Wang, K. M. K. Win, M. E. H. Ong, and L. Y. Hsu. Time series analysis of demographic and temporal trends of tuberculosis in singapore. *BMC Public Health*, 14(1):1–10, 2014.

J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

M. Wen, J. Park, and K. Cho. A scenario generation pipeline for autonomous vehicle simulators. *Human-centric Computing and Information Sciences*, 10(1):1–15, 2020a.

Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020b.

X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2100–2110, 2019.

H. Zhang, N. Dullerud, L. Seyyed-Kalantari, Q. Morris, S. Joshi, and M. Ghassemi. An empirical framework for domain generalization in clinical settings. In *Proceedings of the conference on health, inference, and learning*, pages 279–290, 2021.

S. Zhang, B. Guo, A. Dong, J. He, Z. Xu, and S. X. Chen. Cautionary tales on air-quality improvement in beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205):20170457, 2017.

W. Zhang, M. Ragab, and C.-S. Foo. Domain generalization via selective consistency regularization for time series classification. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 2149–2156. IEEE, 2022.