AIM 2: Artificial Intelligence in Medicine II

Harvard - BMIF 203 and BMI 702, Spring 2025

Lecture 13: Digital biomarkers and disease progression tracking, Patient/disease progression modeling using transformers, Time series in healthcare, Longitudinal EHR modeling

HARVARD MEDICAL SCHOOL

Kempner INSTITUTE

For the Study of Natural & Artificial Intelligence at Harvard University

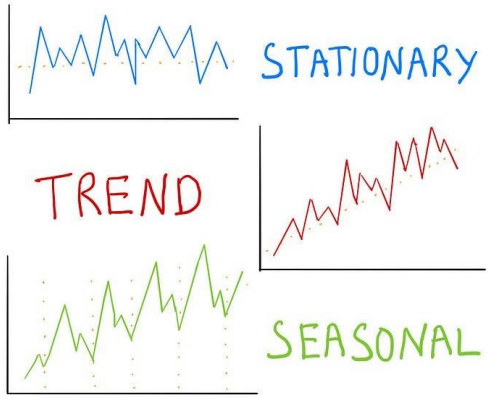
BROAD INSTITUTE

Marinka Zitnik
marinka@hms.harvard.edu

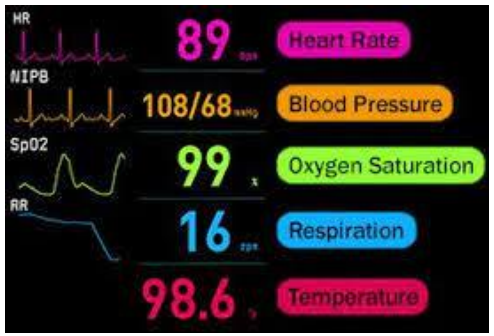
Time series are everywhere

Irregularly sampled with varying time intervals between successive readouts, complex dynamics and various sensors observed at different time points

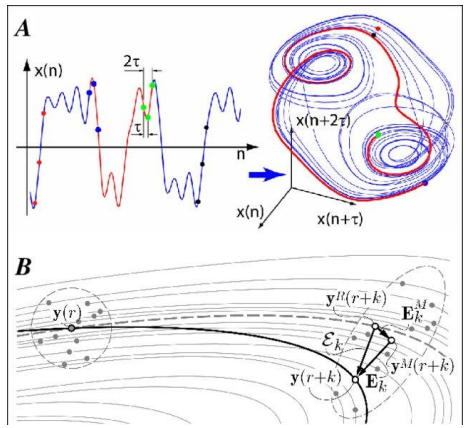
Climate



Healthcare

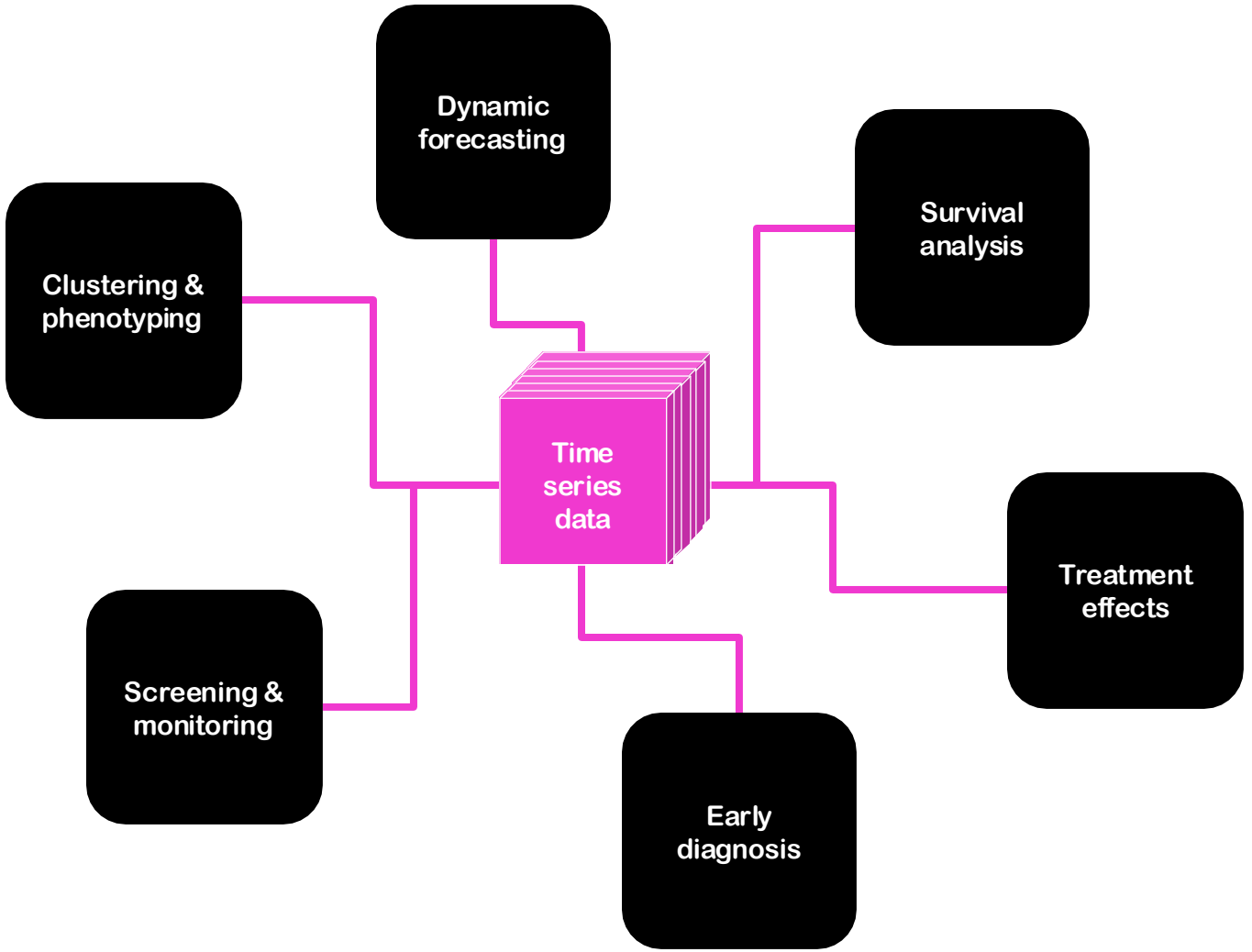


Space systems



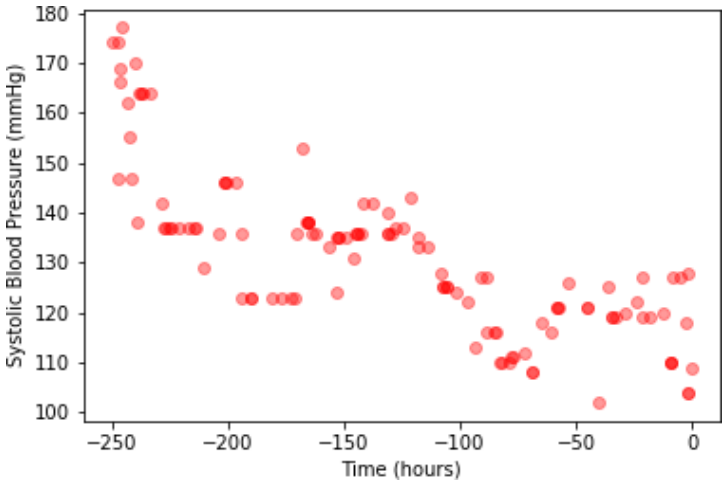
Domain Adaptation for Time Series Under Feature and Label Shifts, ICML 2023
Graph-Guided Network for Irregularly Sampled Multivariate Time Series, ICLR 2022
Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency, NeurIPS 2022
Encoding Time-Series Explanations through Self-Supervised Model Behavior Consistency, NeurIPS 2023

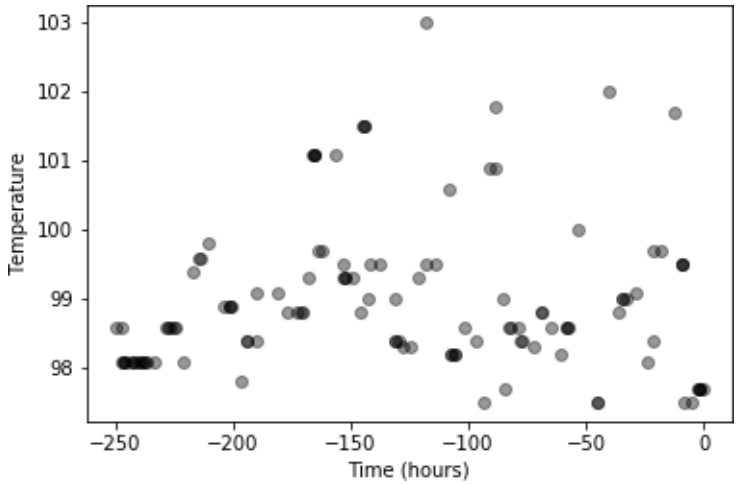
Time series in healthcare

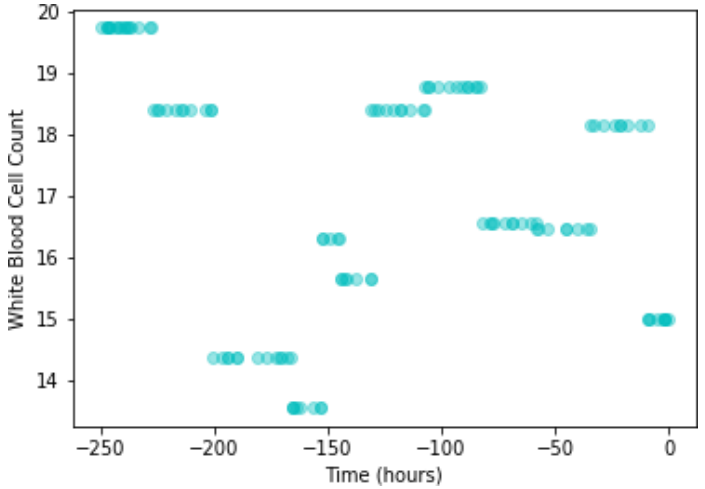


Time series in healthcare

- Multiple streams of measurements
- Measurements are sparse, irregularly and informatively sampled
- Multiple outcomes of interest (various events of interest, various morbidities)
- True clinical states are unobserved (e.g., onset of diseases)
- Many possible patterns (heterogeneous phenotypes, comorbidities)

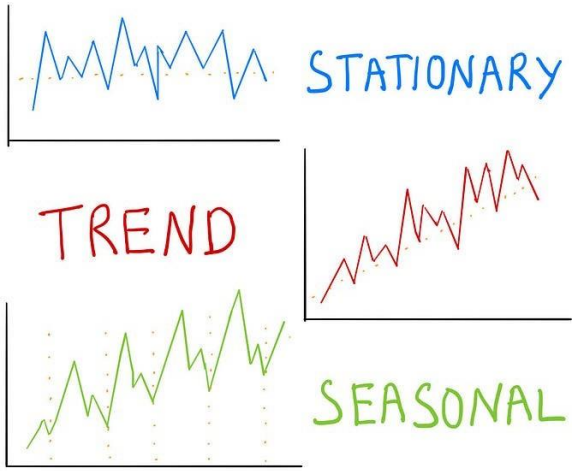






Time series data come in different forms

- Time series consist of
  - Data points ordered in time
  - Preferably in regular intervals
  - Typically, 1 or more components/channels/dimensions



- Time axis not strictly needed but helpful to
  - Order data
  - Perform analysis and forecast future readouts
  - Temporal resolution of the time series depends on the use-case
    - Often in minutes, hours, days, weeks, month, ...

Today's lecture

1. **Time series tasks in healthcare**
2. **Self-supervised pre-training for time series**
3. **Learning representations** of regular and irregular time series
4. **Understanding time series models**

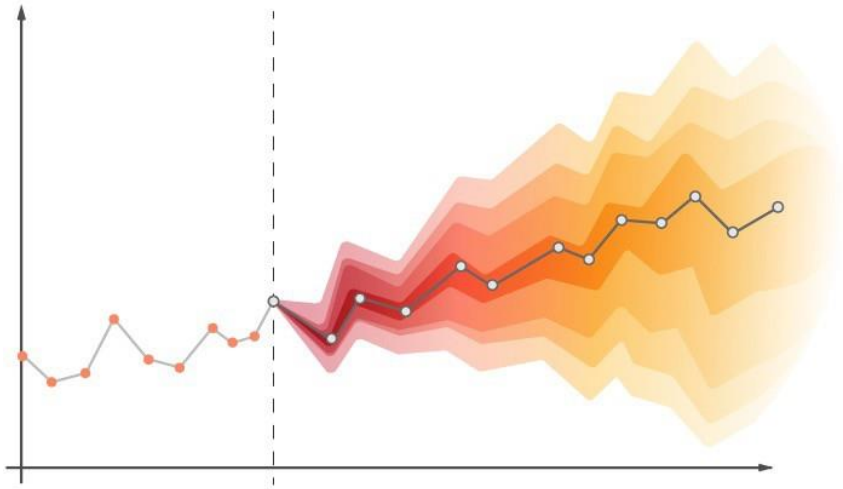
Time series tasks in healthcare

- Dynamic forecasting
- Time-to-event and survival analysis
- Clustering and phenotyping
- Screening and monitoring
- Early prediction of diagnosis
- Treatment effect prediction

Time series forecasting: Predicting the future

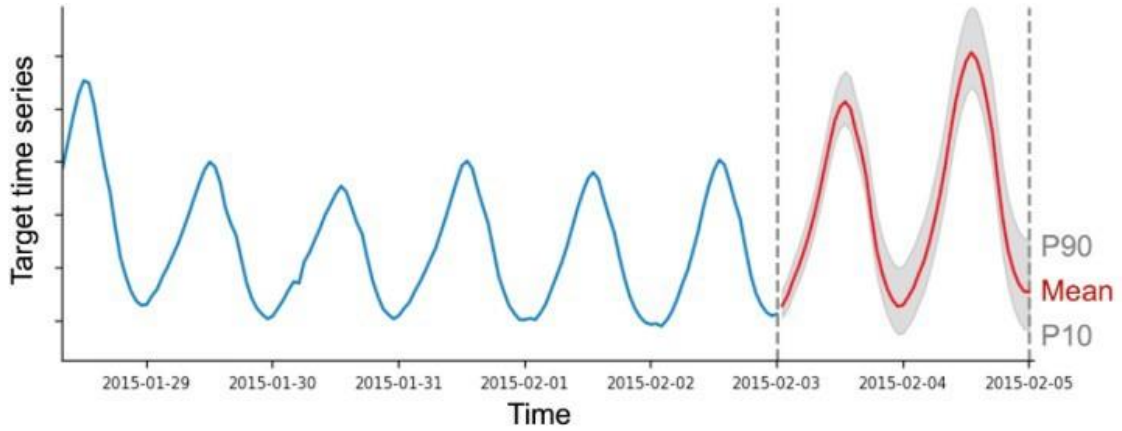
Univariate forecasting

- Single time series forecasting (with time)
- The future may be forecasted just by looking at the past
- Analysis and ML forecasting methods for this are widely researched
- Simple methods like moving average, regression, and general neural networks can often be sufficient, depending on the accuracy required



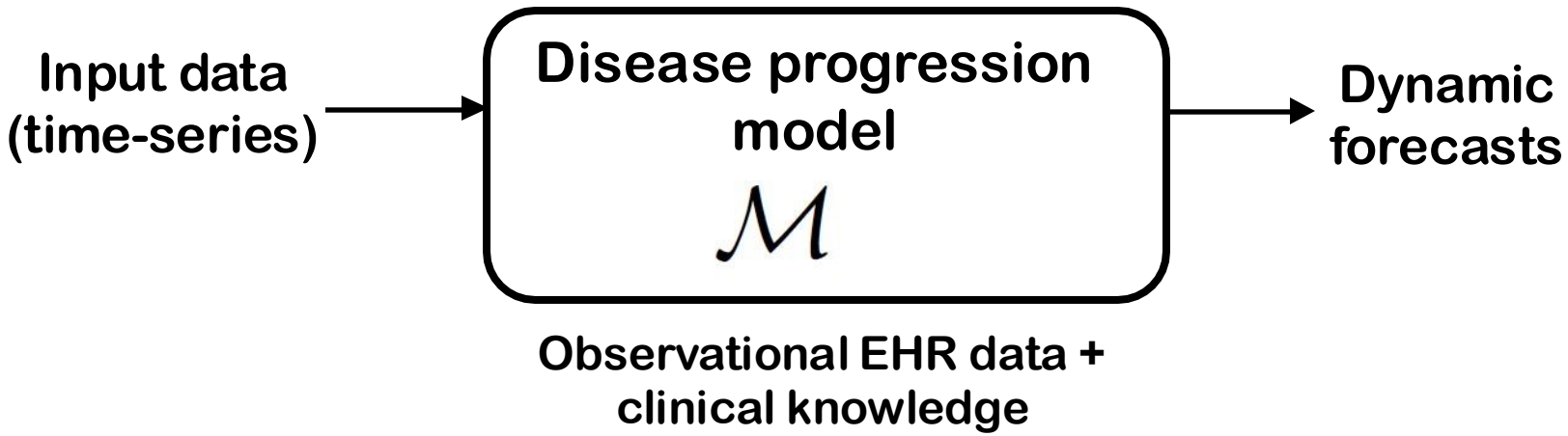
Multivariate forecasting

- Forecasting with multiple time series
- Access to conditional past and future data, both continuous and categorical
- Accurate forecasting is a difficult task
  - Dependency on multiple covariates
  - Long-range dependencies
  - Inherent uncertainty in input/target data



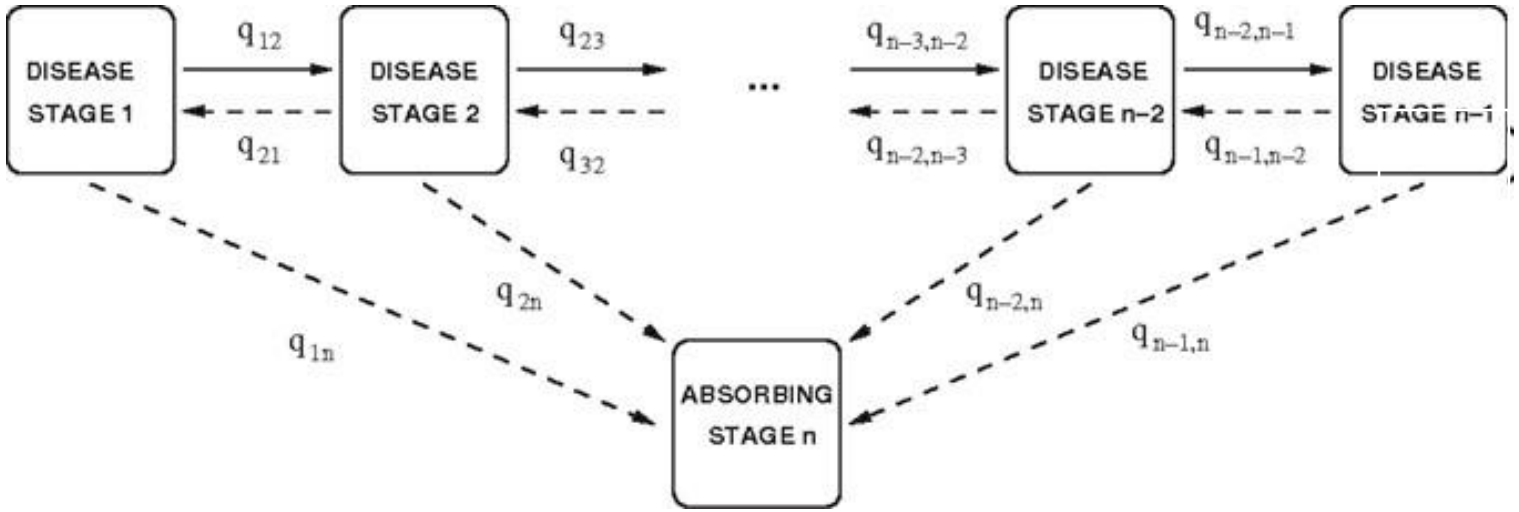
Time series forecasting: Predicting the future

- Build disease progression models
- Understand and model carefully the available data!
- Learn the model parameters from available EHR data (Training time)
- Issue dynamic forecasts for the patient at hand (Test time/Run-time)
- Unravel new understanding of disease progression
  - Population
  - Sub-groups of patients
  - Personalized



Disease progression models: Formalism

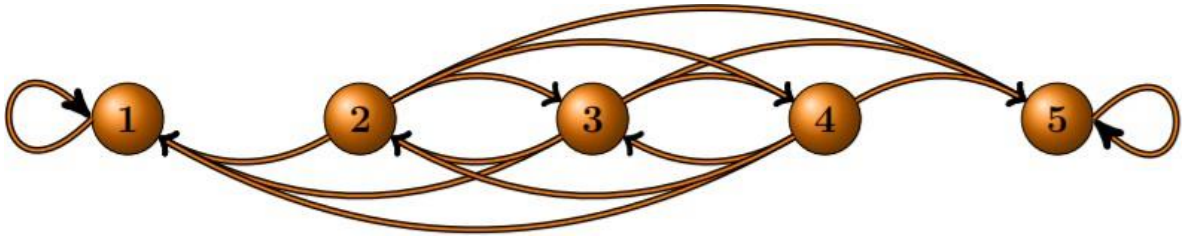
Markov models $P(\boldsymbol{Z}_{n+1} \mid \mathcal{H}_{t_n}) = P(\boldsymbol{Z}_{n+1} \mid \boldsymbol{Z}_n)$



Disadvantages:

- Observable models
- One disease at a time
- “Average” patient

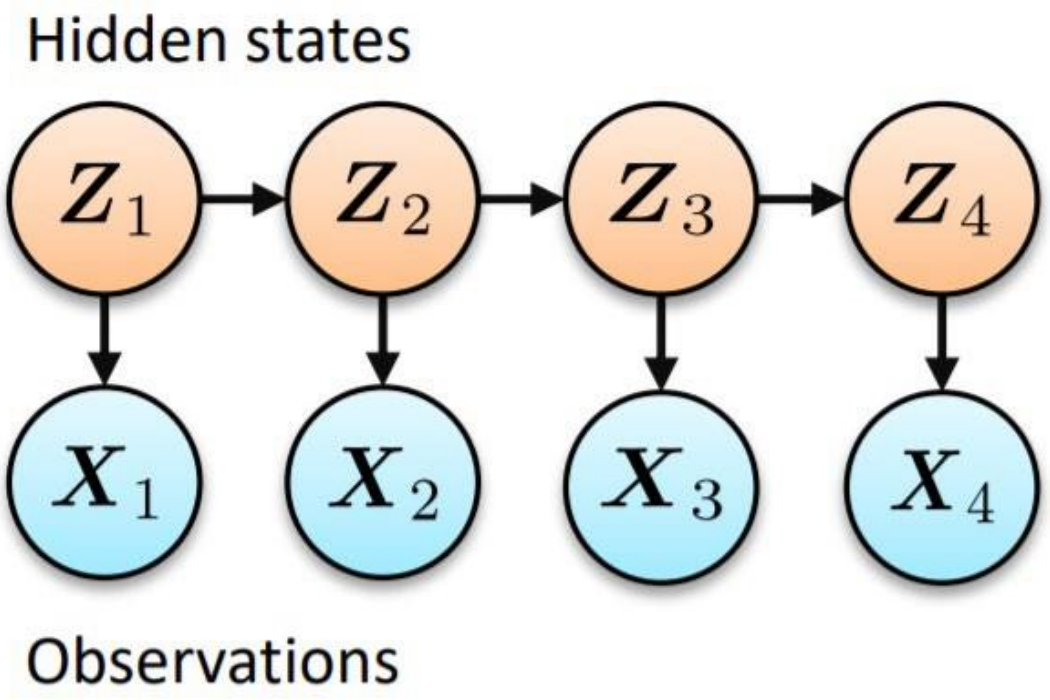
Population-level representation of disease states



Disease progression models: Formalism

Hidden Markov Models (HMMs)

Introducing latent (hidden/unobservable) disease states



Disease Stages

Clinical findings
Lab measurements
Vital signs
Treatments
Events of interest
Observation times

Disease progression models: Goals

History matters!

Ignore history
- Previous states
- Order of states
- Duration in a state

One size fits all!

Only capture population-level transitions across progression stages
Ignores individual clinical trajectories

Goal A: Accurately forecast individual-level disease trajectories

What are the risks of mortality, relapse, comorbidities, complications, etc. in the future?

Goal B: Understand disease progression mechanisms

Underlying latent structure of disease evolution

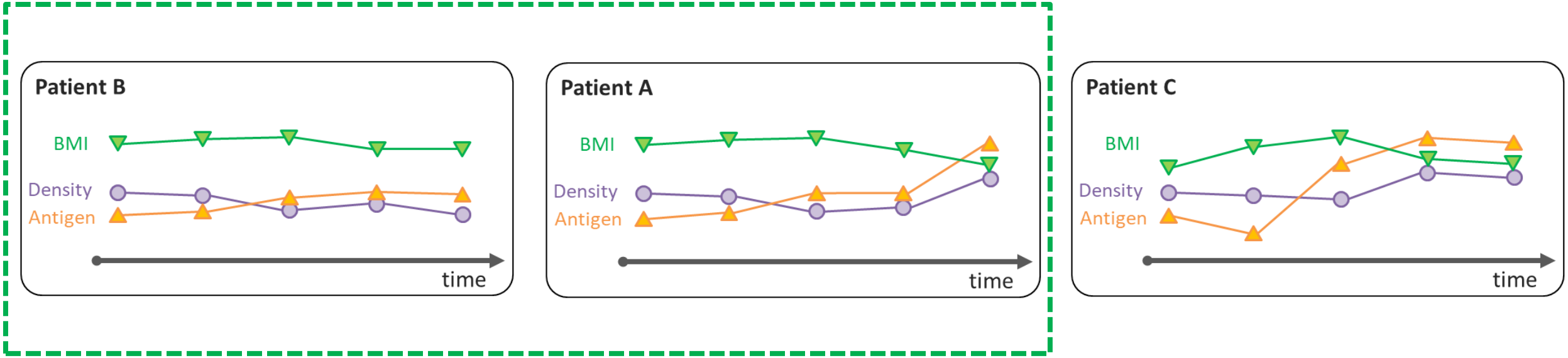
Patients' subgroup analysis

Refined phenotypes

Clustering & phenotyping: How should we group patients?

Example of 3 patients diagnosed with breast cancer (BC)

Should we group patients based on similarity in the time-series observations?



conventional notion of clustering

Key idea: similarity in time-series observations

(e.g. dynamic time warping, auto-encoders)

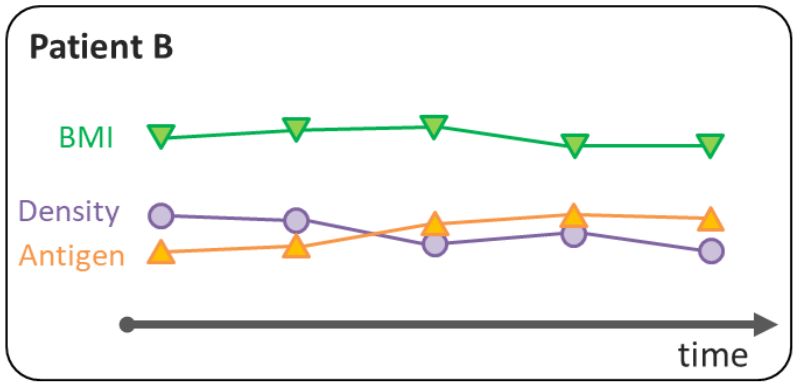
Autoencoder-based approaches

Clustering & phenotyping: How should we group patients?

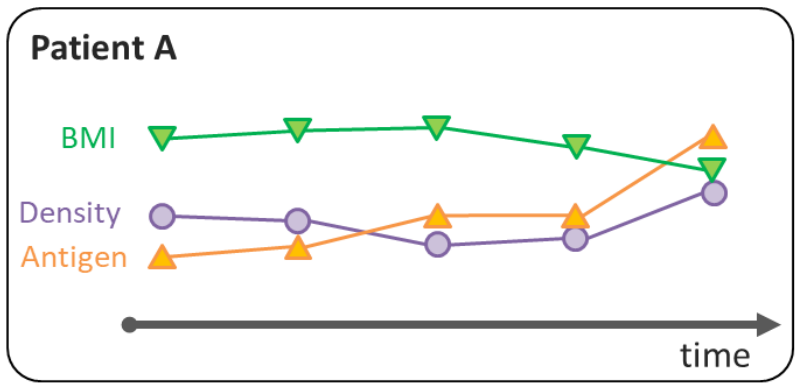
Example of 3 patients diagnosed with breast cancer (BC)

What if both Patient A and C will have an adverse event (e.g., death) that can be expected by increases in cancer antigen and mammographic density

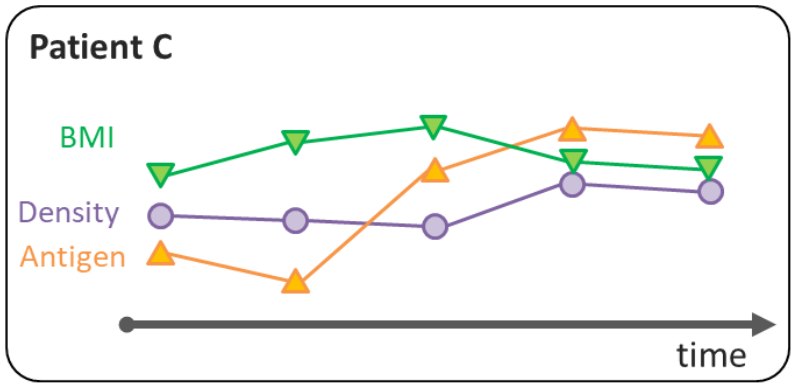
no adverse outcomes



BC-related Death



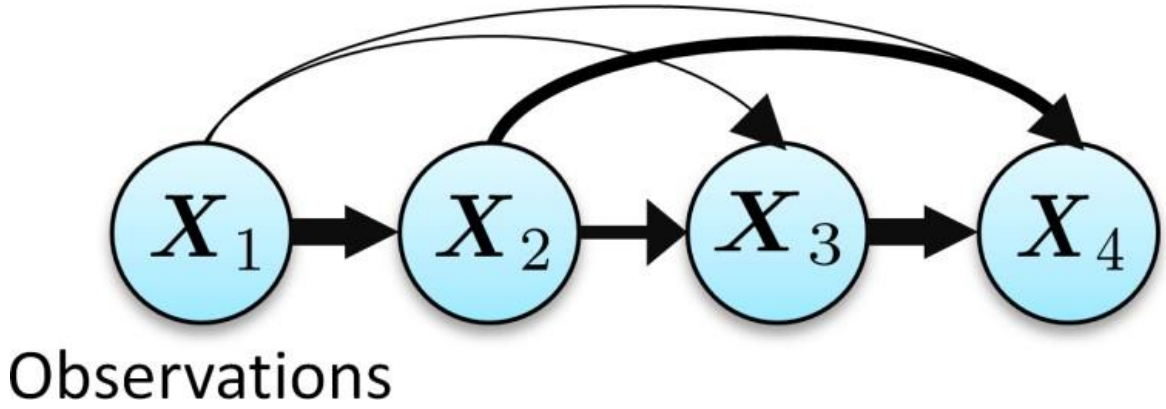
BC-related Death



Temporal notion of clustering: similarity in future outcomes

- **Predictive of similar future outcomes**
- **Doctors and patients can actively plan**
- **Learn representations of past observations (time-series data) that best describe future events and outcomes**

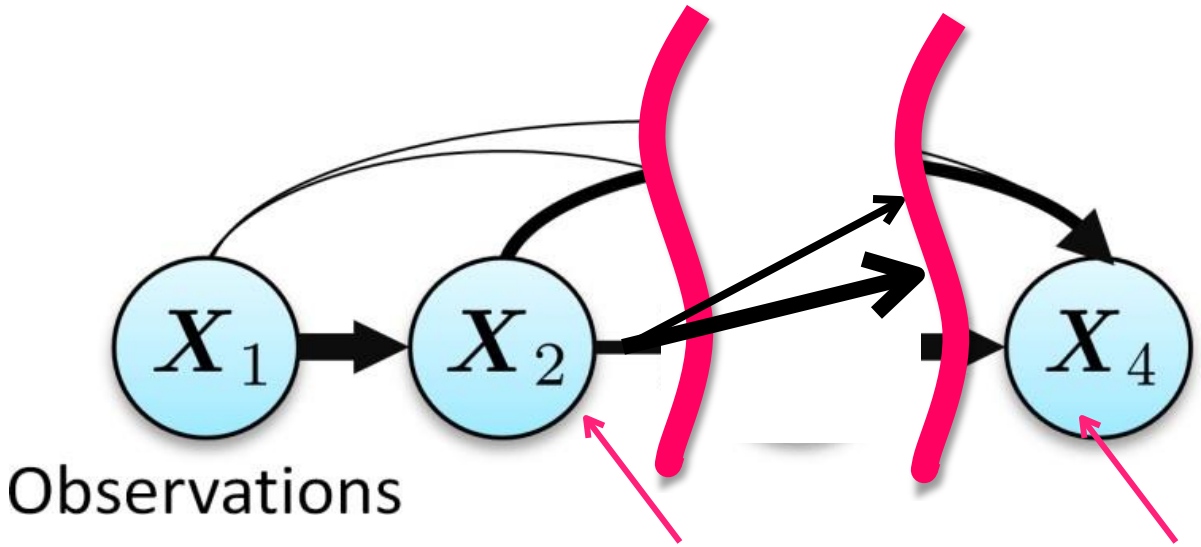
Personalized screening and monitoring



Who to screen?
When to screen?
What to screen?

- **What is the value of various information over time for this event for this individual?**

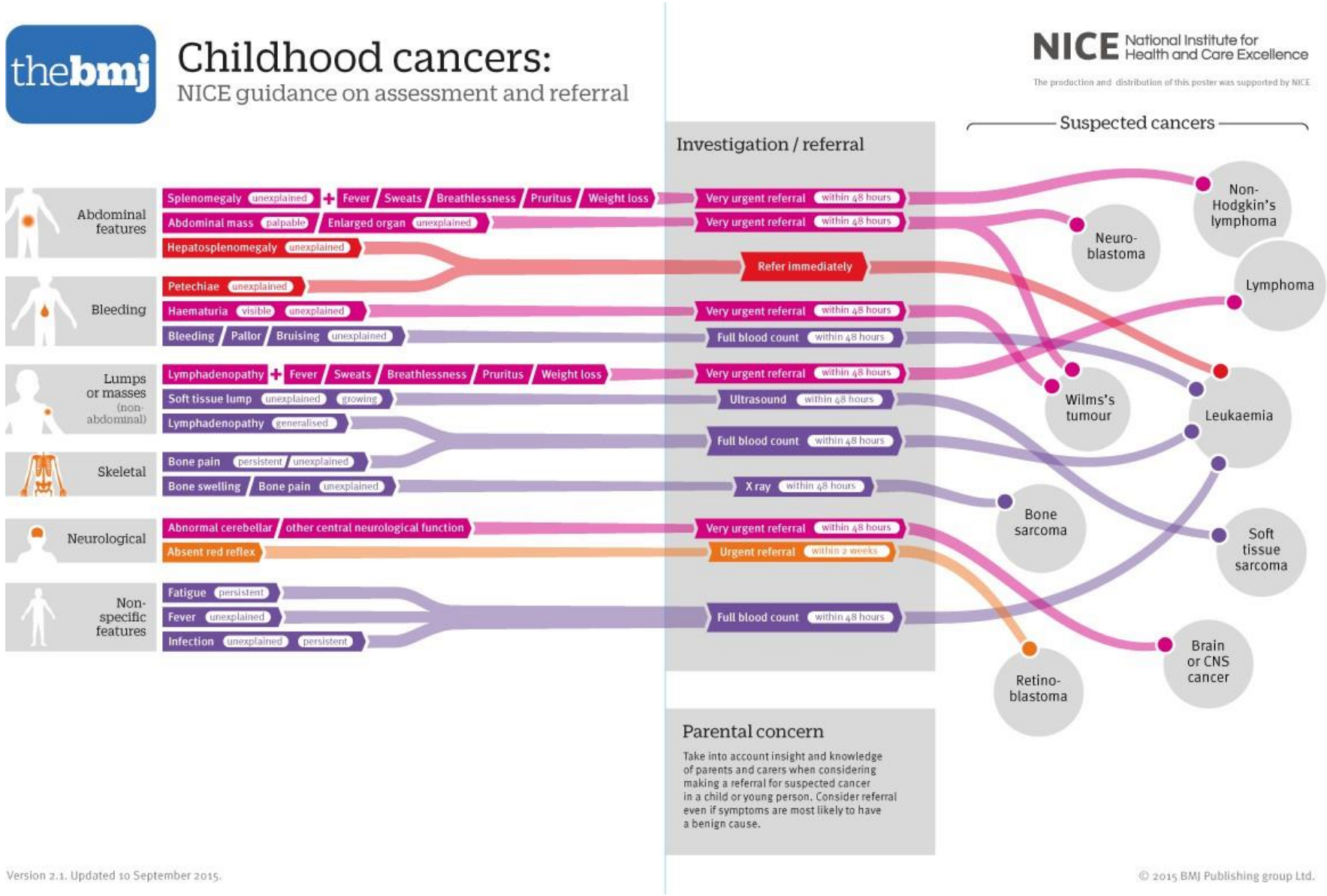
Personalized screening: Formalism



Who to screen?
When to screen?
What to screen?

- **Deep Sensing [Yoon, Jordon, vdS, 2018]**
- **Disease Atlas [Yoon, Jordon, vdS, 2019]**
- **Clairvoyance [Jarrett et al, 2021]**

Early prediction of diagnosis



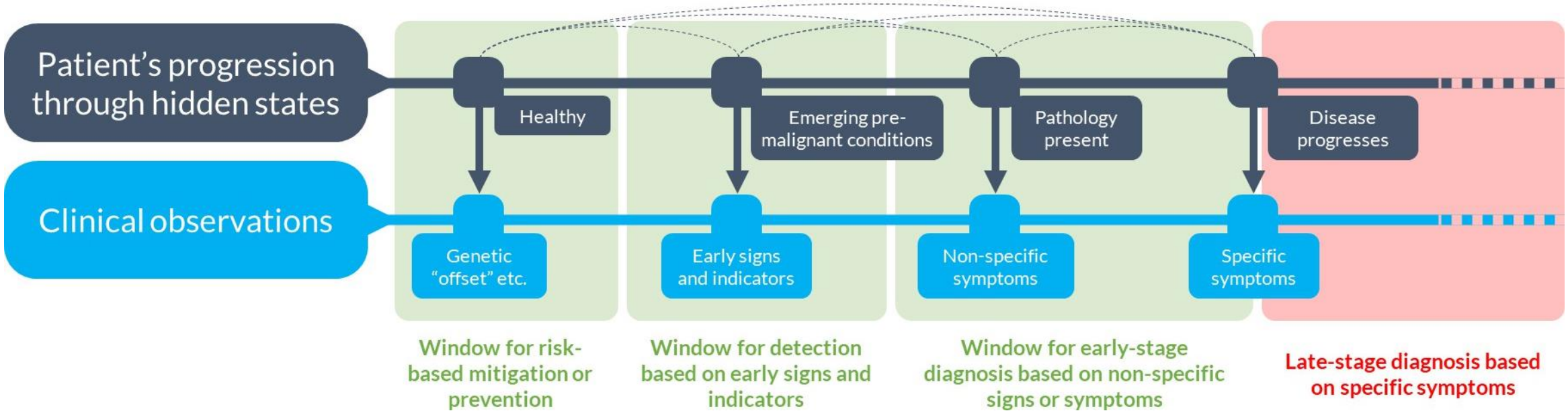
Early prediction of diagnosis

- Risk prediction
  - Segments individuals using population-based risks, usually based on few variables
  - Rarely uses longitudinal data usually only calculated once

- Risk scores then lead to guideline-driven management of patients often rigid
  - Many diseases lack guidelines and protocols
  - This is all predicated upon a quantitative understanding of disease progression

How can we detect disease early?

- Early diagnosis is more than just event prediction/forecasting
- It involves unravelling and dissecting the underlying states of disease progression towards the event of interest

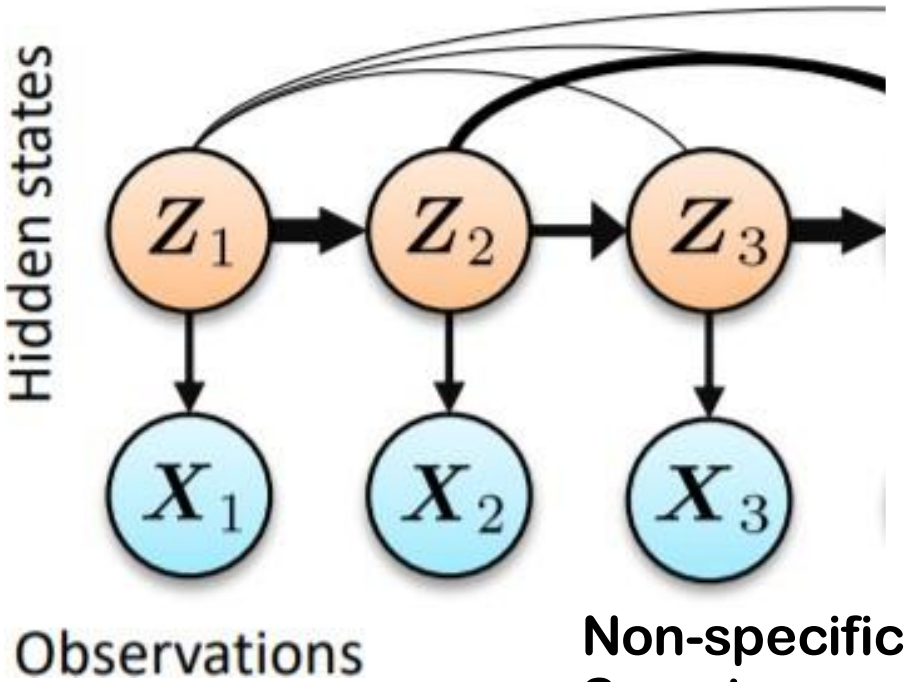


A quantitative understanding of disease progression is needed!

Early diagnosis: How?

Healthy

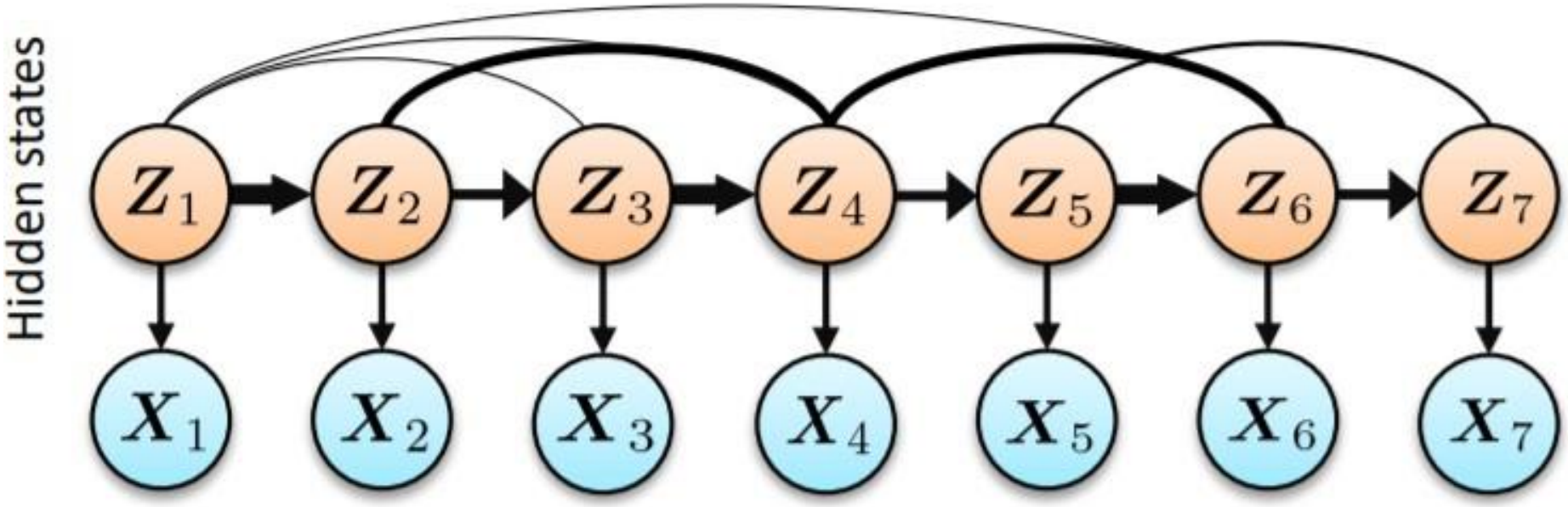
Emerging Pre-malignant Conditions (Dysplasia)



Non-specific Symptoms

Genetic "Offset"

Early diagnosis: How?



Healthy

Emerging Pre-malignant Conditions (Dysplasia)

Pathology Present

Disease Progresses

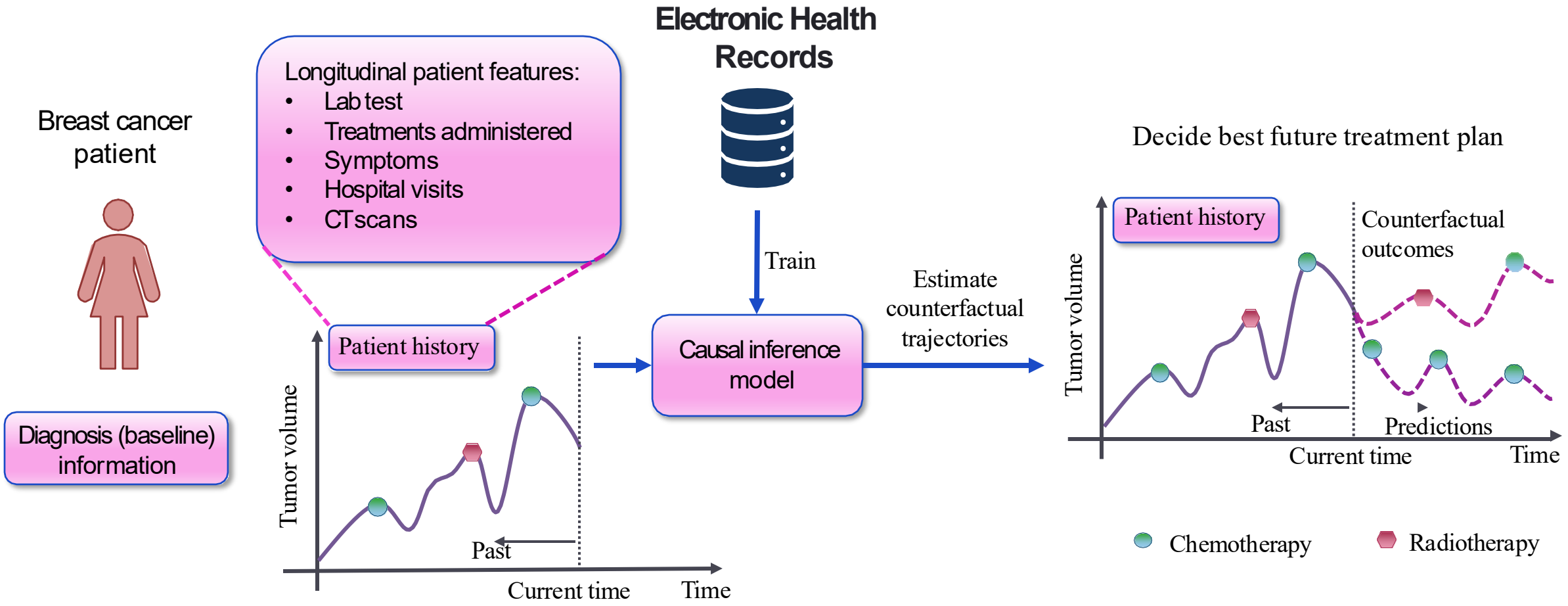
Genetic "Offset"

Non-specific Symptoms

Early diagnosis Made (with ML support)

Diagnosis Made

Treatment effect prediction

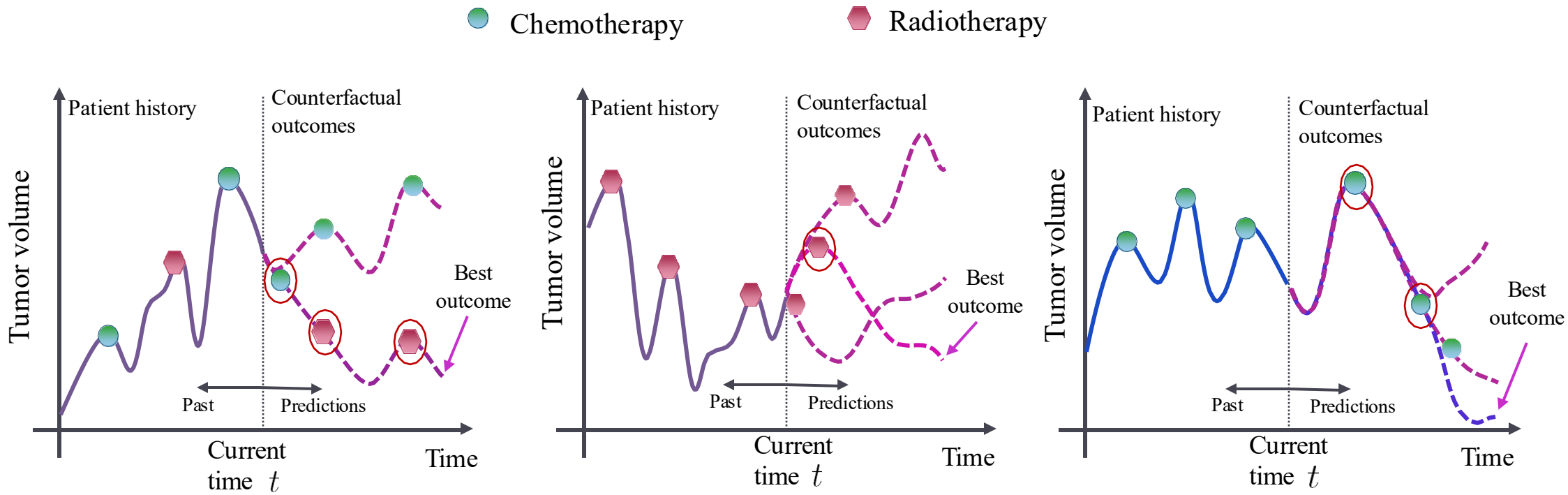


Treatment effect prediction

How to treat?

When to give treatment?

When to stop treatment?

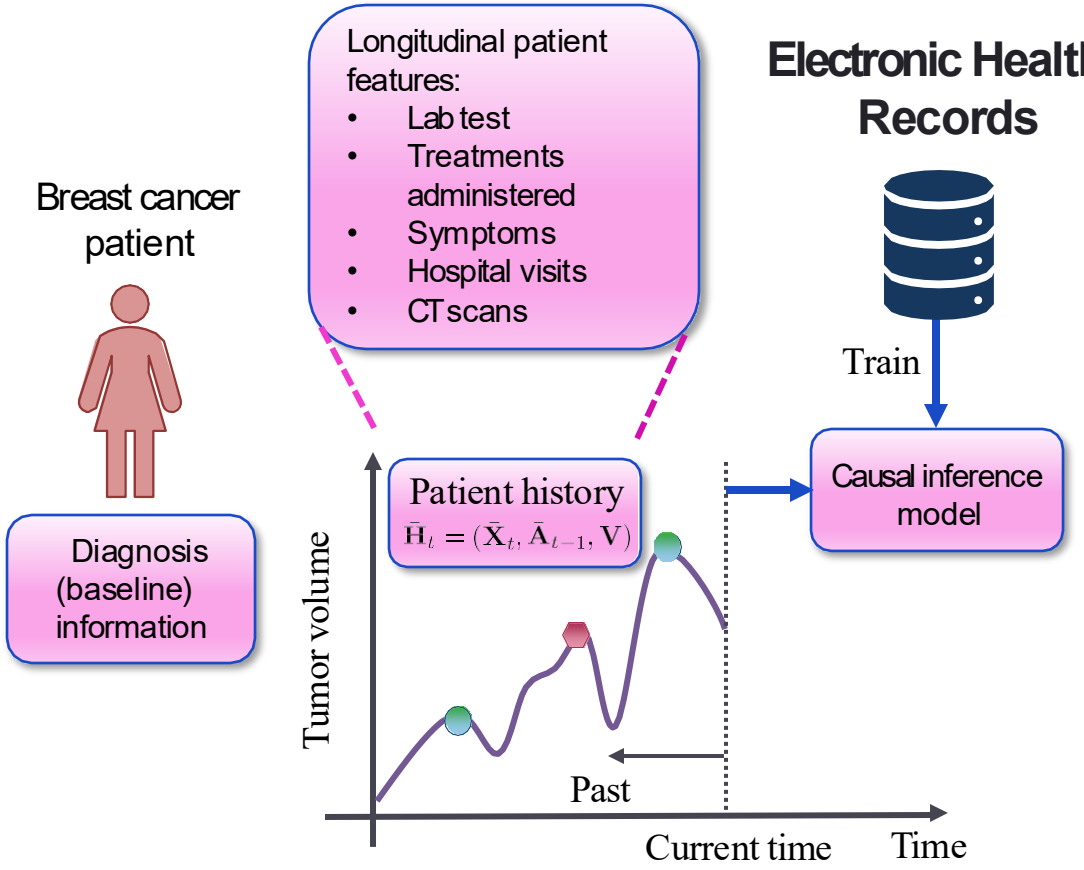


(a) Decide treatment plan

(b) Decide optimal time of treatment

(c) Decide when to stop treatment

Causal effect inference from longitudinal patient observational data



Longitudinal patient observational data

- Time-dependent patient features: $\bar{\mathbf{X}}_t = (\mathbf{X}_1, \ldots \mathbf{X}_t)$
- Time-dependent treatments: $\bar{\mathbf{A}}_t = (\mathbf{A}_1, \ldots \mathbf{A}_t)$ where

$$\mathbf{A}_t \in \{A_1, \ldots, A_K\}$$

- Static patient features: $\mathbf{V}$

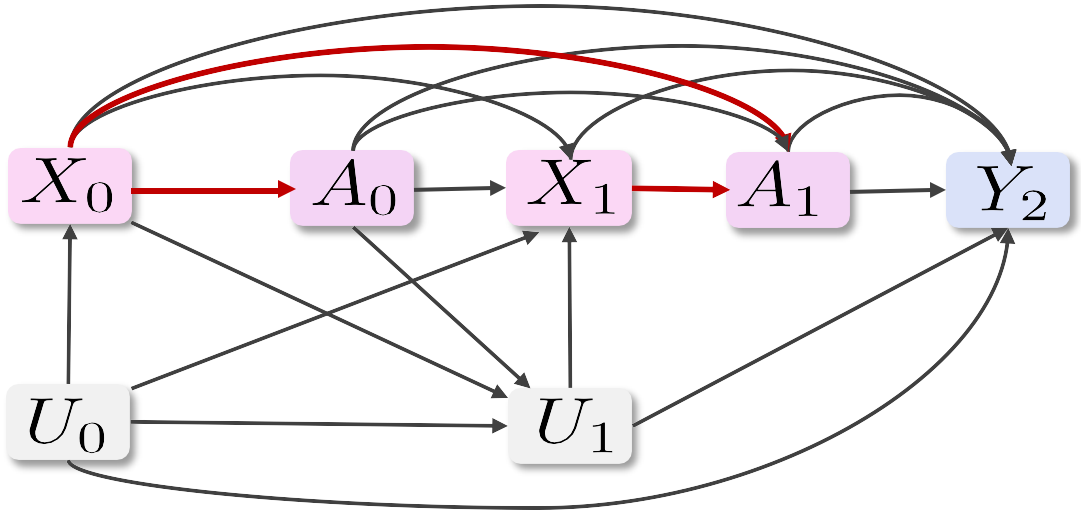
Patient history: $\bar{\mathbf{H}}_t = (\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, \mathbf{V})$

Observed (factual) outcome for treatment $\mathbf{A}_t$ given patient history $\bar{\mathbf{H}}_t$: $\mathbf{Y}_{t+1}$

Challenges in using longitudinal observational data for estimating treatment outcomes

The **patient history** $\bar{\mathbf{H}}_t = (\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, \mathbf{V})$ **contains time-dependent confounders which bias the treatment assignment** $\mathbf{A}_t$ **in the observational dataset.**

Patient covariates - affected by past treatments which then influence future treatments and outcomes



Bias from time-dependent confounders.

Time series tasks in healthcare

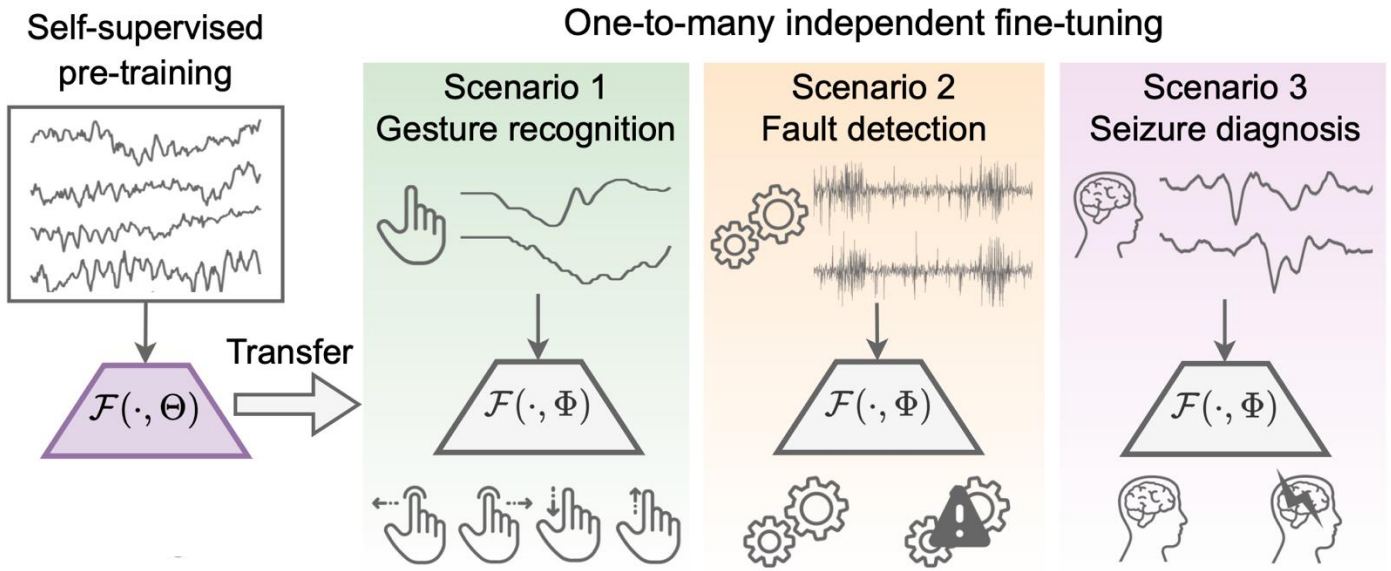
- Dynamic forecasting
- Time-to-event and survival analysis
- Clustering and phenotyping
- Screening and monitoring
- Early prediction of diagnosis
- Treatment effect prediction

Today's lecture

1. **Time series tasks in healthcare**
2. **Self-supervised pre-training for time series**
3. **Learning representations** of regular and irregular time series
4. **Understanding time series models**

Pre-training on time series

- **Question:** How to process a time series dataset so as to greatly improve generalization to new time series coming from a different dataset

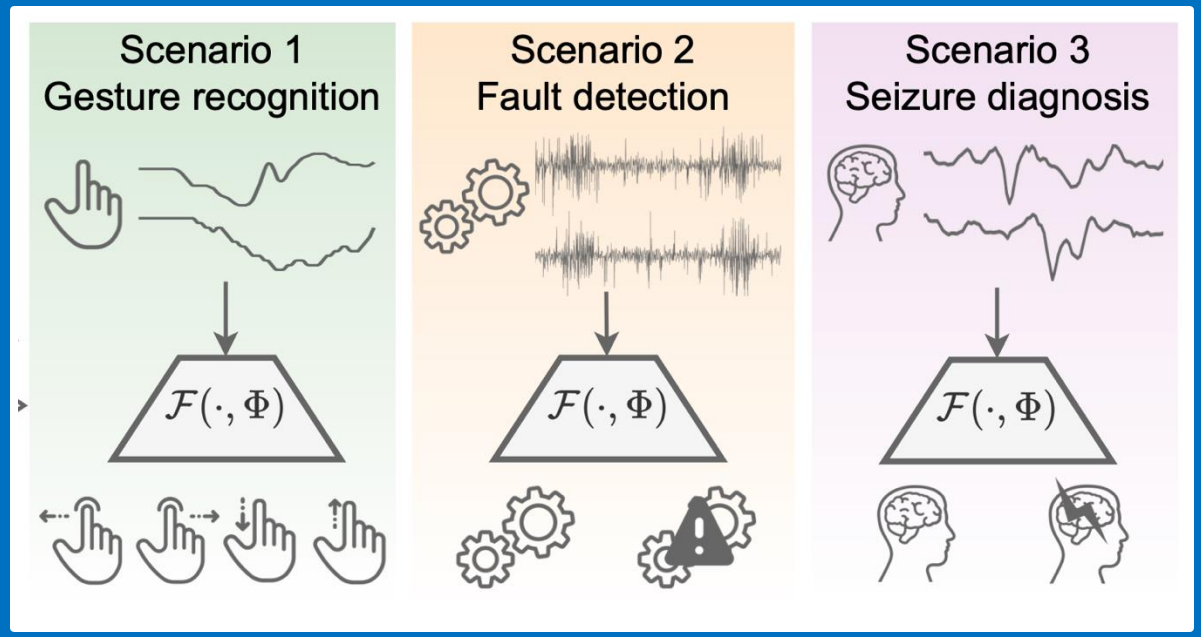


- **Goal:**
  - Transfer a model to new datasets without explicit retraining or minimal adaptation
  - Resulting performance $\geq$ SoTA model on target dataset alone

Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency, NeurIPS 2022

Why is this challenging?

- Expected performance gains are often not realized:
  - Dist
  - Pro
  - Larg
  - Vary
    dev
    different
- For ex
  comp
  tasks involving high-frequency components (transient events)
  equency
  er on

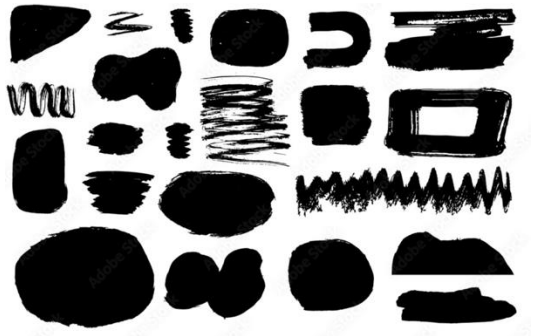


What inductive biases could facilitate such generalizable representations of time series?

Do time series datasets represent unique challenges?

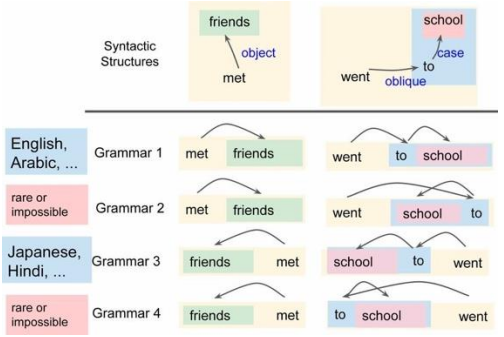
- Target datasets are not available in pre-training
  - Pre-trained model must capture a latent shared property that can apply to an unseen dataset
- Need to identify a shared property universal to different time series datasets to enable transfer from pre-training to target datasets

Vision



Shapes, edges, texture

Language



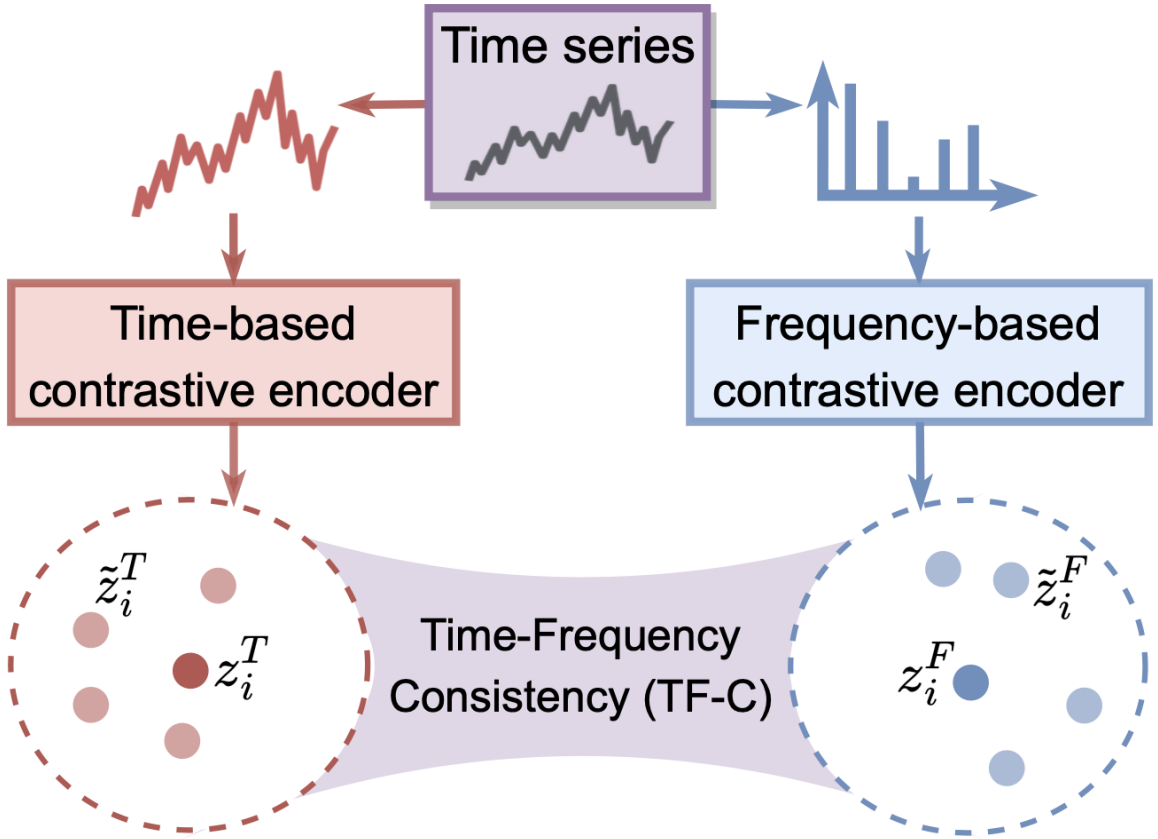
Word order, grammar

Time series

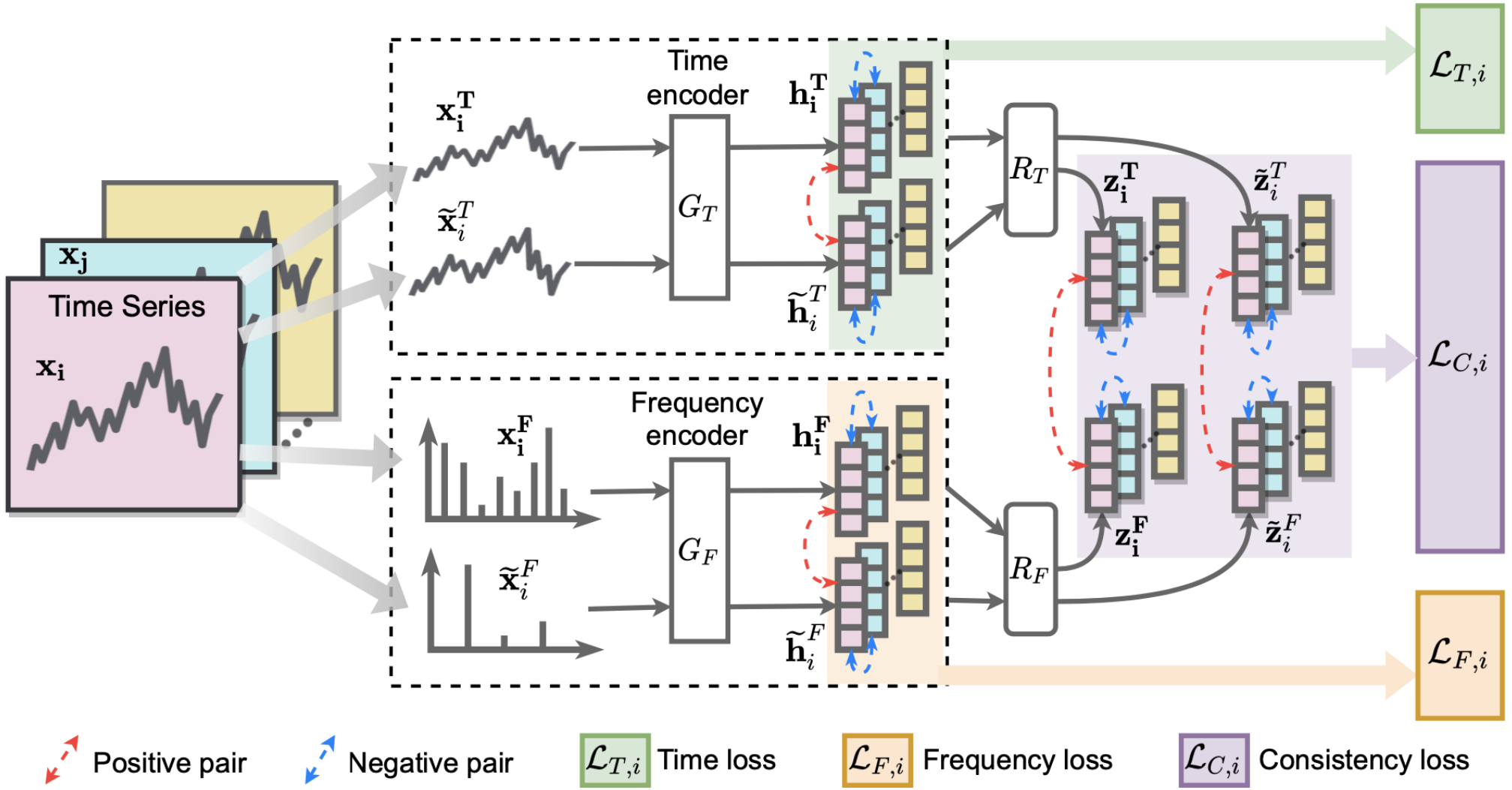
?

Time-frequency consistency (TF-C)

Representational Time-Frequency Consistency (TF-C). *Let be given a time series sample* $\boldsymbol{x}_i$. *Then in a model* $\mathcal{F}$ *satisfying TF-C, time-based representation* $\boldsymbol{z}_i^{\mathrm{T}}$ *and frequency-based representation* $\boldsymbol{z}_i^{\mathrm{F}}$ *learned from* $\boldsymbol{x}_i$ *by* $\mathcal{F}$, *and representations learned from local augmentations of* $\boldsymbol{x}_i$ *are close together in the latent time-frequency space.*

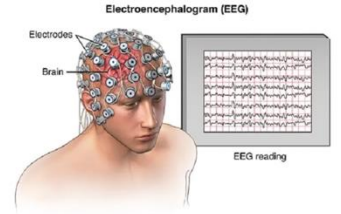


Overview of TF-C approach



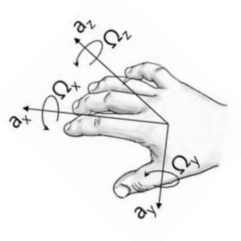
Datasets

- **SleepEEG:** 371,055 univariate brainwaves (100 Hz) collected from 197 individuals. Samples are labeled by 5 sleeping stages
- **Epilepsy:** Brain activity of 500 subjects recorded by single-channel EEG (174 Hz). Samples are labeled by epilepsy seizures

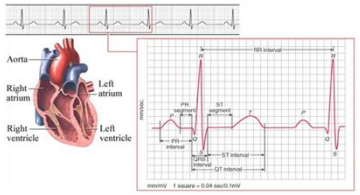


- **FD-A:** Vibration from rolling bearing of a mechanical system aiming at fault detection. Every sample has 5,120 timestamps and an indicator for one out of three mechanical device states
- **FD-B:** Same as FD-A but the rolling bearings are performed in different working conditions (*e.g.*, varying rotational speed)

- **HAR:** 10,299 9-dimensional samples recording 6 daily activities
- **GESTURE:** 440 accelerometer samples on 8 hand gestures



- **ECG:** 8,528 single-sensor ECG recordings sorted into four classes based on human physiology
- **EMG:** 163 EMG samples with 3-class labels implying muscular diseases



Baselines and experimental setup

- Six pre-training methods:
  - TS-SD, TS2vec, CLOCS, Mixing-up, TS-TCC, SimCLR
- To examine utility of pre-training:
  - Non-deep learning KNN model applied directly to fine-tuning datasets
  - Random initialization approach to randomly initialize fine-tuning model
- Performance metrics: accuracy, precision (macro-averaged), recall, F1 score, AUROC, and AUPRC

Results: One-to-one transfer learning

- Pre-train a model on a **pre-training dataset** and fine-tune the model on **one target dataset only**
  - HAR → GESTURE: 6 types of human daily activities measured by an 8-channel time series → 8 hand gestures measured by 1 channel

7.2% margin over the best baseline

Models	Accuracy	Precision	Recall	F1 score	AUROC	AUPRC
Non-DL (KNN)	0.6766±0.0000	0.6500±0.0000	0.6821±0.0000	0.6442±0.0000	0.8190±0.0000	0.5231±0.0000
Random Init.	0.4219±0.0865	0.4751±0.0925	0.4963±0.1026	0.4886±0.0967	0.7129±0.1206	0.3358±0.1194
TS-SD	0.6937±0.0533	0.6806±0.0496	0.6883±0.0525	0.6785±0.0495	0.8708±0.0305	0.6261±0.0790
TS2vec	0.6453±0.0260	0.6287±0.0339	0.6451±0.0218	0.6261±0.0294	0.8890±0.0054	0.6670±0.0118
CLOCS	0.4731±0.0229	0.4639±0.0432	0.4766±0.0266	0.4392±0.0198	0.8161±0.0068	0.4916±0.0103
Mixing-up	0.7183±0.0123	0.7001±0.0166	0.7183±0.0123	0.6991±0.0145	**0.9127±0.0018**	0.7654±0.0071
TS-TCC	0.7593±0.0242	0.7668±0.0257	0.7566±0.0231	0.7457±0.0210	0.8866±0.0040	0.7217±0.0121
SimCLR	0.4383±0.0652	0.4255±0.1072	0.4383±0.0652	0.3713±0.0919	0.7721±0.0559	0.4116±0.0971
TF-C (Ours)	**0.7824±0.0237**	**0.7982±0.0496**	**0.8011±0.0322**	**0.7991±0.0296**	0.9052±0.0136	**0.7861±0.0149**

On average, our TF-C model claims a large margin of 15.4% over all baselines. Further, the strongest baseline varies across scenarios (i.e., TS-TCC in FD-A → FD-B; Mixing-up in SleepEEG → Epilepsy)

Results: One-to-many transfer learning

- Pre-train a model on a dataset and use it across a broad **range of tasks across many target datasets from diverse domains**
  - SleepEEG → {EPILEPSY, FD-B, GESTURE, EMG}

Scenarios	Models	Accuracy	Precision	Recall	F1 score	AUROC	AUPRC
SleepEEG → Epilepsy	Non-DL (KNN)	0.8525±0.0000	0.8639±0.0000	0.6431±0.0000	0.6791±0.0000	0.6434±0.0000	0.6279±0.0000
	Random Init.	0.8983±0.0656	0.9213±0.1369	0.7447±0.1135	0.7959±0.1208	0.8578±0.2153	0.6489±0.1926
	TS-SD	0.8952±0.0522	0.8018±0.2244	0.7647±0.1485	0.7767±0.1855	0.7677±0.2452	0.7940±0.1825
	TS2vec	0.9395±0.0044	0.9059±0.0116	0.9039±0.0118	0.9045±0.0067	0.9587±0.0086	0.9430±0.0103
	CLOCS	**0.9507±0.0027**	0.9301±0.0067	**0.9127±0.0165**	**0.9206±0.0066**	0.9803±0.0023	0.9609±0.0116
	Mixing-up	0.8021±0.0000	0.4011±0.0000	0.5000±0.0000	0.4451±0.0000	0.9743±0.0081	0.9618±0.0104
	TS-TCC	0.9253±0.0098	0.9451±0.0049	0.8181±0.0257	0.8633±0.0215	0.9842±0.0034	0.9744±0.0043
	SimCLR	0.9071±0.0344	0.9221±0.0166	0.7864±0.1071	0.8178±0.0998	0.9045±0.0539	0.9128±0.0205
	TF-C (Ours)	0.9495±0.0249	**0.9456±0.0108**	0.8908±0.0216	0.9149±0.0534	**0.9811±0.0237**	**0.9703±0.0199**
SleepEEG → FD-B	Non-DL (KNN)	0.4473±0.0000	0.2847±0.0000	0.3275±0.0000	0.2284±0.0000	0.4946±0.0000	0.3308±0.0000
	Random Init.	0.4736±0.0623	0.4829±0.0529	0.5235±0.1023	0.4911±0.0590	0.7864±0.0349	0.7528±0.0254
	TS-SD	0.5566±0.0210	0.5710±0.0535	0.6054±0.0272	0.5703±0.0328	0.7196±0.0113	0.5693±0.0532
	TS2vec	0.4790±0.0113	0.4339±0.0092	0.4842±0.0197	0.4389±0.0107	0.6463±0.0130	0.4442±0.0162
	CLOCS	0.4927±0.0310	0.4824±0.0316	0.5873±0.0387	0.4746±0.0485	0.6992±0.0099	0.5501±0.0365
	Mixing-up	0.6789±0.0246	0.7146±0.0343	**0.7613±0.0198**	0.7273±0.0228	0.8209±0.0035	0.7707±0.0042
	TS-TCC	0.5499±0.0220	0.5279±0.0293	0.6396±0.0178	0.5418±0.0338	0.7329±0.0203	0.5824±0.0468
	SimCLR	0.4917±0.0437	0.5446±0.1024	0.4760±0.0885	0.4224±0.1138	0.6619±0.0219	0.5009±0.0477
	TF-C (Ours)	**0.6938±0.0231**	**0.7559±0.0349**	0.7202±0.0257	**0.7487±0.0268**	**0.8965±0.0135**	**0.7871±0.0267**
SleepEEG → Gesture	Non-DL (KNN)	0.6833±0.0000	0.6501±0.0000	0.6833±0.0000	0.6443±0.0000	0.8190±0.0000	0.5232±0.0000
	Random Init.	0.4219±0.0629	0.4751±0.0175	0.4963±0.0679	0.4886±0.0459	0.7129±0.0166	0.3358±0.1439
	TS-SD	0.6922±0.0444	0.6698±0.0472	0.6867±0.0488	0.6656±0.0443	0.8725±0.0324	0.6185±0.0966
	TS2vec	0.6917±0.0333	0.6545±0.0358	0.6854±0.0349	0.6570±0.0392	0.8968±0.0123	0.6989±0.0346
	CLOCS	0.4433±0.0518	0.4237±0.0794	0.4433±0.0518	0.4014±0.0602	0.8073±0.0109	0.4460±0.0384
	Mixing-up	0.6933±0.0231	0.6719±0.0232	0.6933±0.0231	0.6497±0.0306	0.8915±0.0261	0.7279±0.0558
	TS-TCC	0.7188±0.0349	0.7135±0.0352	0.7167±0.0373	0.6984±0.0360	0.9099±0.0085	0.7675±0.0201
	SimCLR	0.4804±0.0594	0.5946±0.1623	0.5411±0.1946	0.4955±0.1870	0.8131±0.0521	0.5076±0.1588
	TF-C (Ours)	**0.7642±0.0196**	**0.7731±0.0355**	**0.7429±0.0268**	**0.7572±0.0311**	**0.9238±0.0159**	**0.7961±0.0109**
SleepEEG → Emg	Non-DL (KNN)	0.4390±0.0000	0.3772±0.0000	0.5143±0.0000	0.3979±0.0000	0.6025±0.0000	0.4084±0.0000
	Random Init.	0.7780±0.0729	0.5909±0.0625	0.6667±0.0135	0.6238±0.0267	0.9109±0.1239	0.7771±0.1427
	TS-SD	0.4606±0.0000	0.1545±0.0000	0.3333±0.0000	0.2111±0.0000	0.5005±0.0126	0.3775±0.0110
	TS2vec	0.7854±0.0318	**0.8040±0.0750**	0.6785±0.0396	0.6766±0.0501	**0.9331±0.0164**	**0.8436±0.0372**
	CLOCS	0.6985±0.0323	0.5306±0.0750	0.5354±0.0291	0.5139±0.0409	0.7923±0.0573	0.6484±0.0680
	Mixing-up	0.3024±0.0534	0.1099±0.0126	0.2583±0.0456	0.1541±0.0204	0.4506±0.1718	0.3660±0.1635
	TS-TCC	0.7889±0.0192	0.5851±0.0974	0.6310±0.0991	0.5904±0.0952	0.8851±0.0113	0.7939±0.0386
	SimCLR	0.6146±0.0582	0.5361±0.1724	0.4990±0.1214	0.4708±0.1486	0.7799±0.1344	0.6392±0.1596
	TF-C (Ours)	**0.8171±0.0287**	0.7265±0.0353	**0.8159±0.0289**	**0.7683±0.0311**	0.9152±0.0211	0.8329±0.0137

- Pre-training and fine-tuning datasets SleepEEG → {FD-B, GESTURE, EMG} are diverse. This gap leads to poor baseline performance
- TF-C has high tolerance to diverse datasets and can can serve as a universal model when no relevant pre-training datasets are available
- TF-C is the best performer in 14/18 settings with an 8.4% performance gain
- TF-C consistently outperforms KNN and Random Init. by a large margin of 42.8% and 25.1%

Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency, NeurIPS 2022

Is TF-C principle needed for pre-training time series models?

- Removing $\mathcal{L}_{TF-C}$, $\mathcal{L}_{T}$, and $\mathcal{L}_{F}$ result in performance degradation of 6.1%, 7.2%, and 6.7%
- Replacing $\mathcal{L}_{TF-C}$ with a loss measuring consistency in time ($\mathcal{L}_{TT-C}$) or frequency ($\mathcal{L}_{FF-C}$): performance gain cannot be simply explained by contrastive loss; time-frequency consistency is crucial
  - $\mathcal{L}_{TF-C}$ outperforms $\mathcal{L}_{TT-C}$ and $\mathcal{L}_{FF-C}$ by 5.3% and 7.2%

	Accuracy	**Precision**	**Recall**	**F1 score**
W/o $\mathcal{L}_{\text{C}}$ and $\mathcal{L}_{\text{T}}$	0.7159+-0.0128	0.7211+-0.0428	0.7246+-0.0428	0.7239+-0.0429
W/o $\mathcal{L}_{\text{C}}$ and $\mathcal{L}_{\text{F}}$	0.7327+-0.0328	0.7246+-0.0311	0.7339+-0.0307	0.7317+-0.0356
W/o $\mathcal{L}_{\text{C}}$	0.7428+-0.0297	0.7289+-0.0278	0.7451+-0.0263	0.7377+-0.0308
Replace $\mathcal{L}_{\text{C}}$ with $\mathcal{L}_{\text{FF-C}}$	0.7259+-0.0072	0.7319+-0.0256	0.7338+-0.0133	0.7341+-0.0194
Replace $\mathcal{L}_{\text{C}}$ with $\mathcal{L}_{\text{TT-C}}$	0.7124+-0.0091	0.7256+-0.0169	0.7231+-0.0197	0.7296+-0.0209
Full Model (TF-C)	**0.7642+-0.0196**	**0.7731+-0.0355**	**0.7429+-0.0268**	**0.7572+-0.0311**

Self-supervised pre-training: Recap

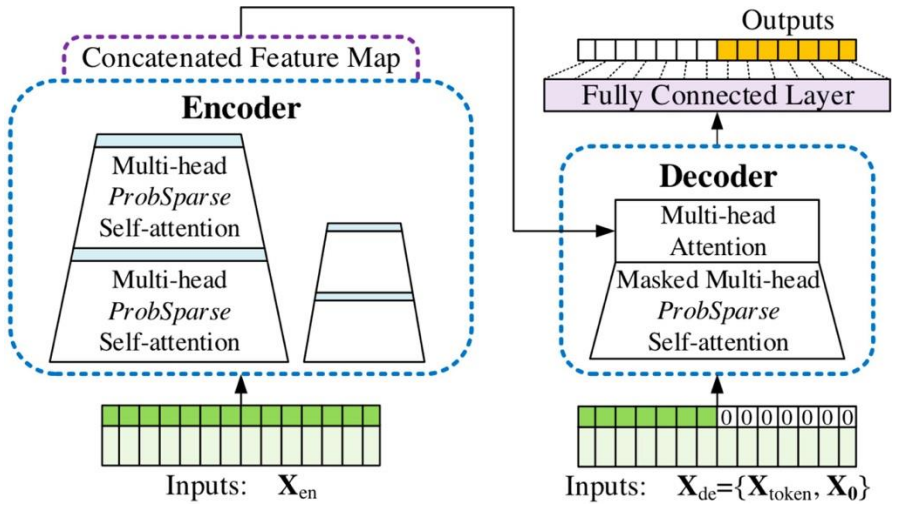
- Pre-training for time series poses **unique challenges** due to potential mismatches between pre-training and target domains, e.g., shifts in temporal dynamics, evolving trends, and long-range and short effects
- **Self-supervised approach**: TF-C uses contrastive learning to inject TF-C into a pre-training model, bringing the encoded time-based and frequency-based representations and their local neighborhoods close together in the latent space
- Strong **generalization**: Our findings have implications for building broadly generalizable pre-training models for time series

Sequence-to-sequence neural models: Beyond natural language understanding

- State-of-the-art for sequence modeling
- Self attention
- No-recurrent units, allowing parallel computation
- Widely used in almost all language tasks now
  - Machine translation
  - Text generation
  - Question answering

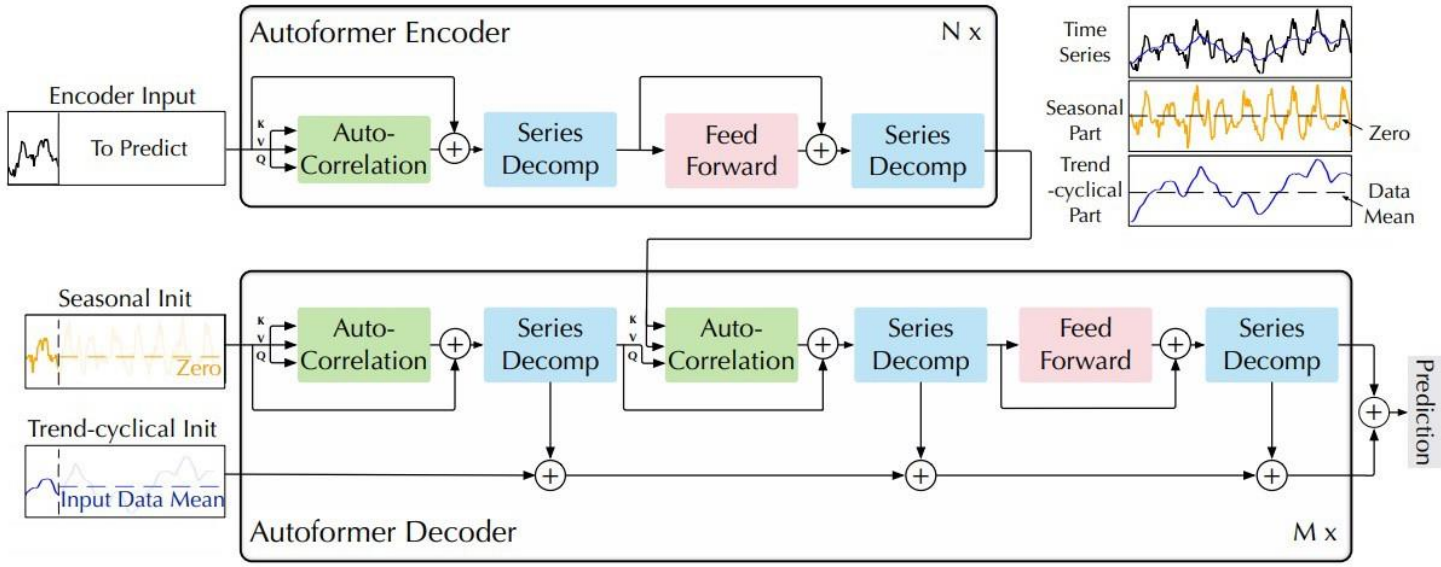
Transformer architectures for time series

Informer – for long term time series forecasting



arXiv:2012.07436

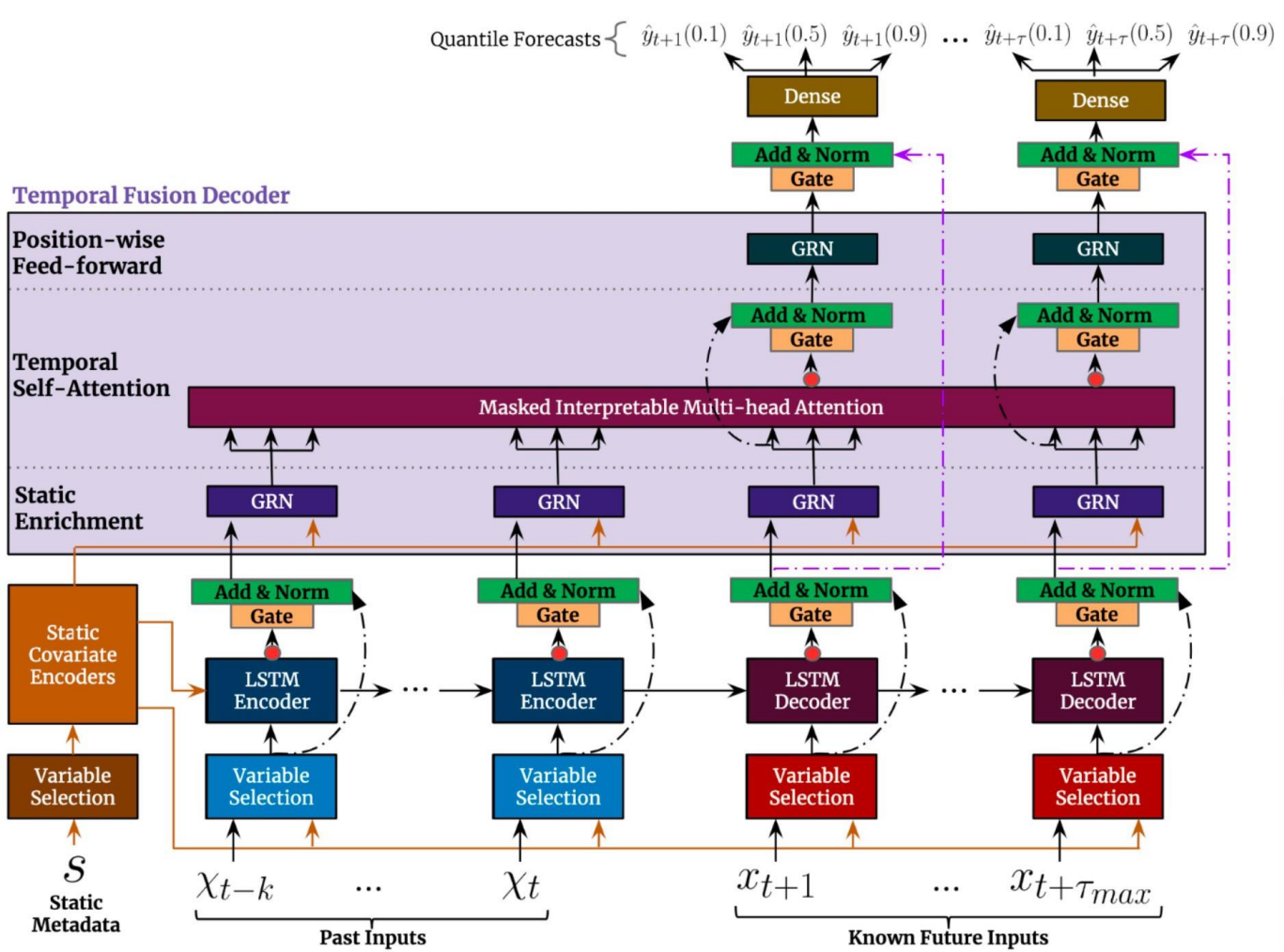
Autoformer – decomposing the time series components

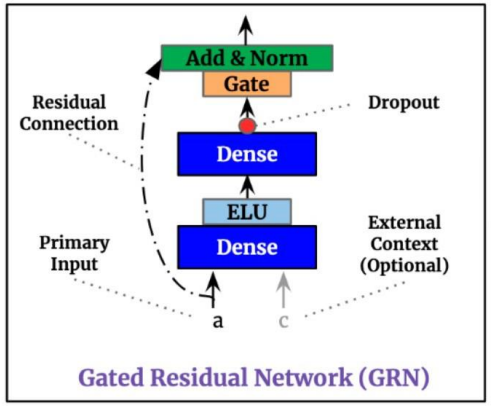


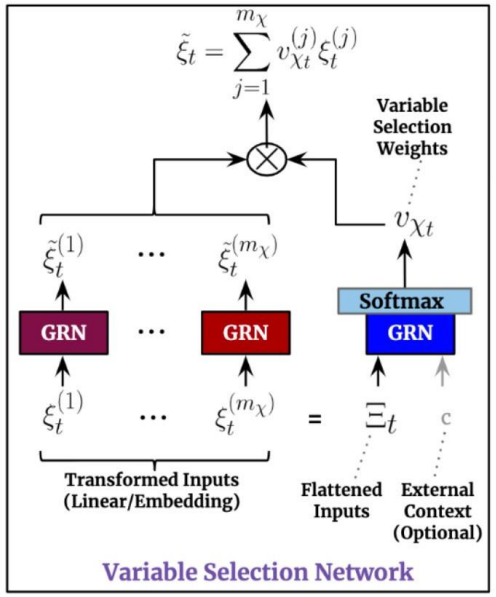
arXiv:2106.13008

…and many more

Transformer architectures for time series



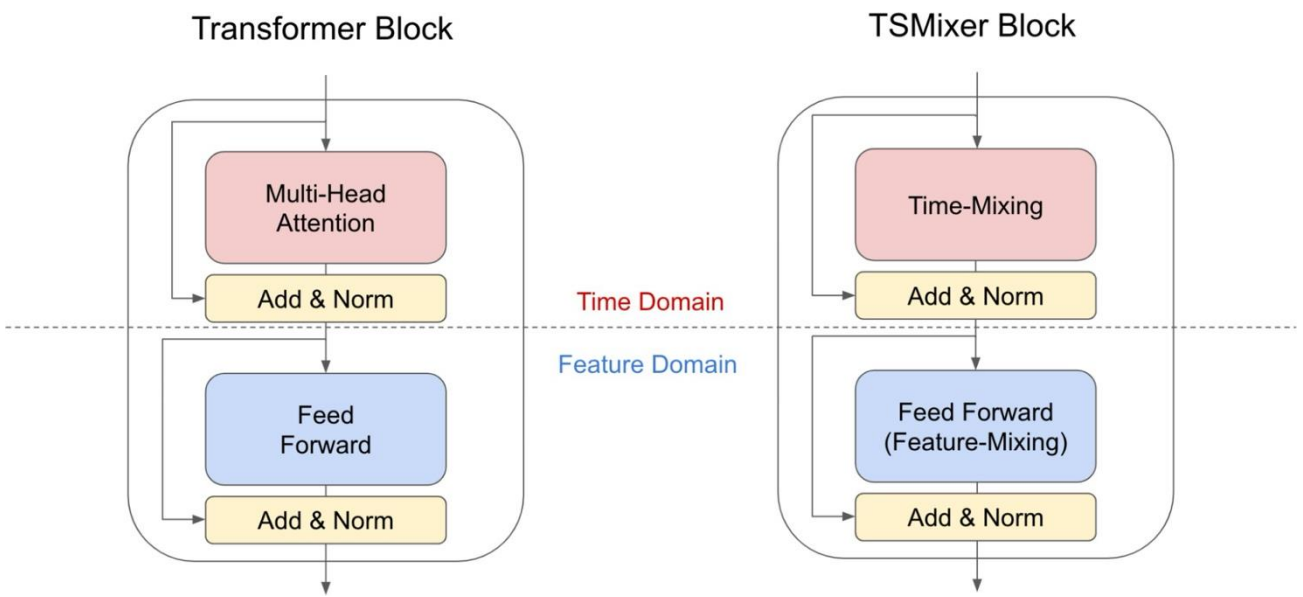


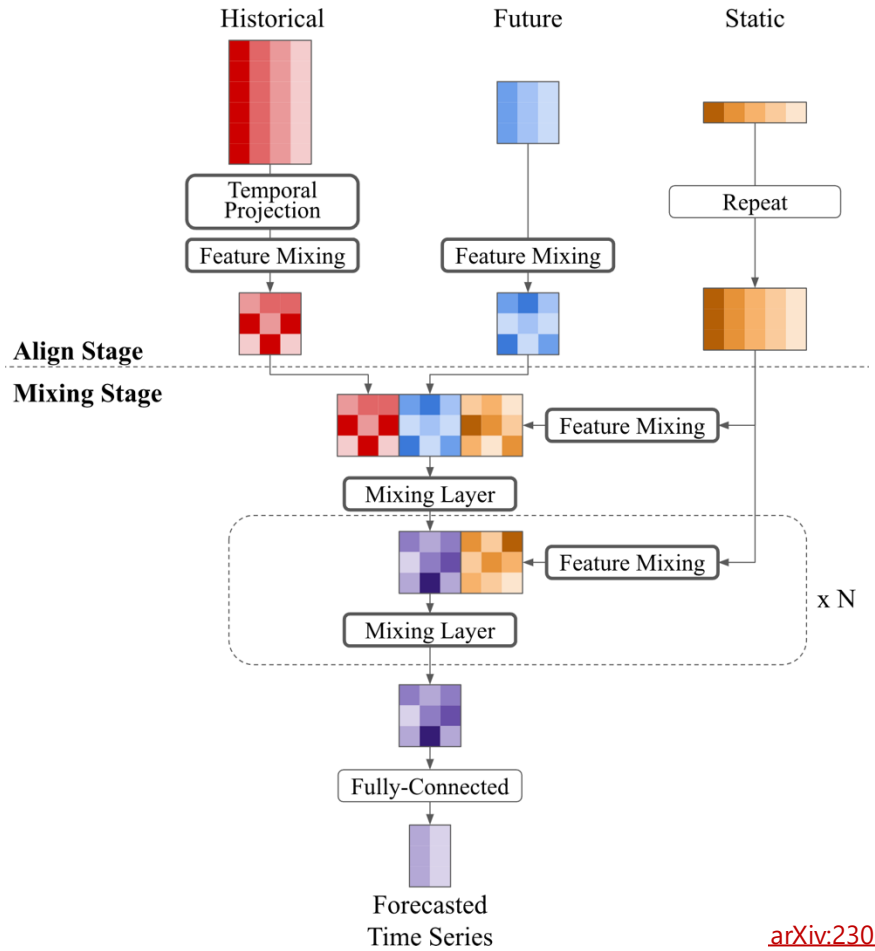


arXiv:1912.09363

Temporal fusion transformer

Transformer architectures for time series





arXiv:2303.06053

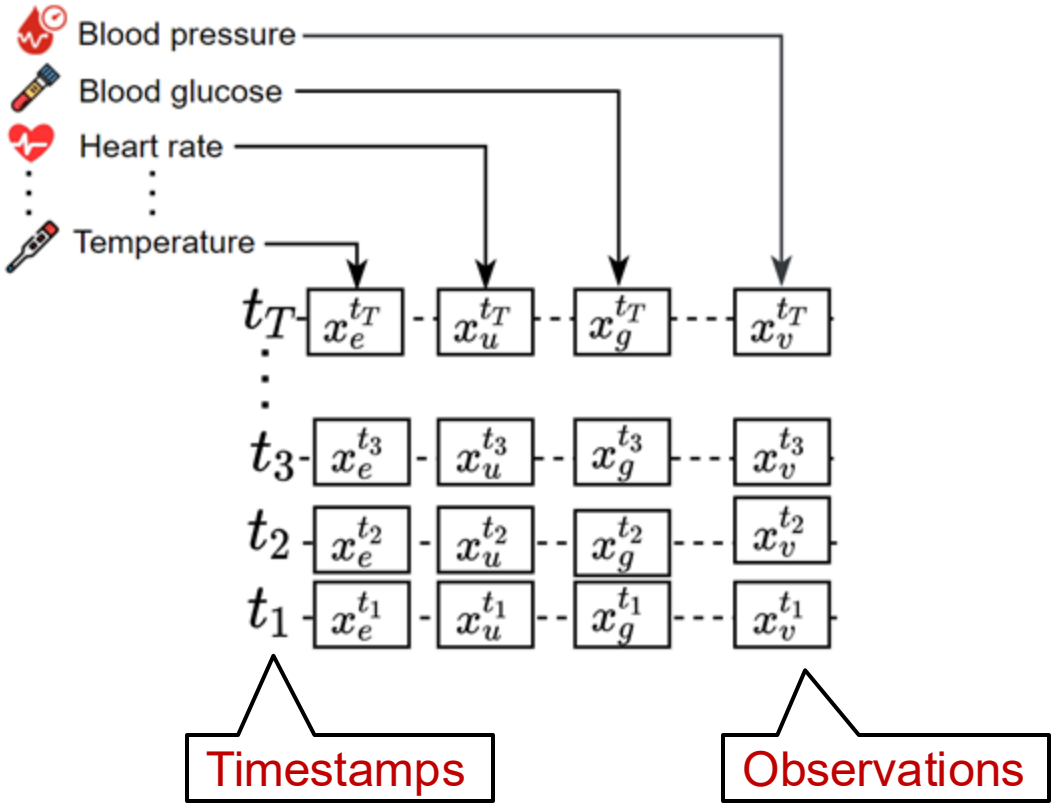
TSMixer – An all MLP architecture for time series forecasting

Today's lecture

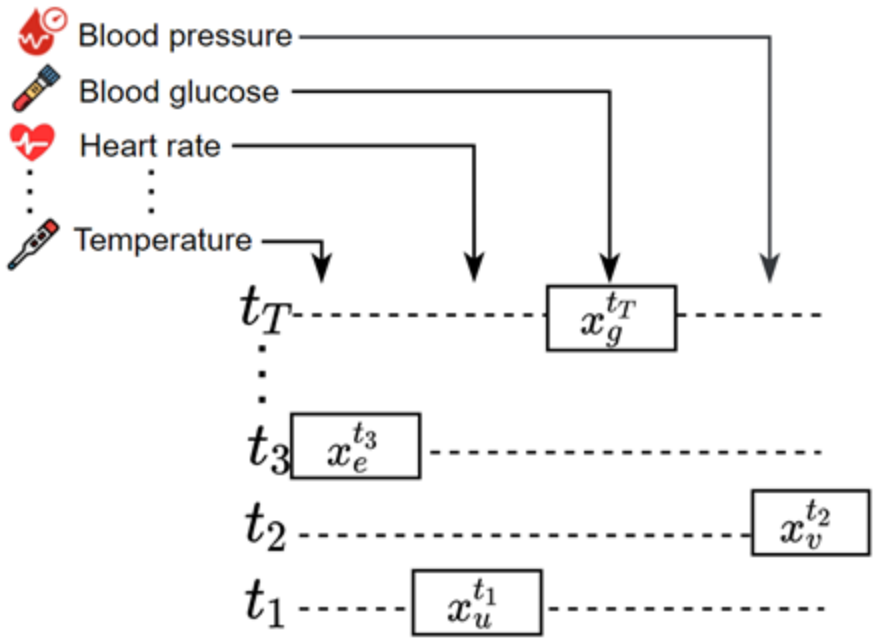
1. **Time series tasks in healthcare**
2. **Self-supervised pre-training for time series**
3. **Learning representations** of regular and irregular time series
4. **Understanding time series models**

Irregular vs. regular time series

Regular time series



Irregular time series



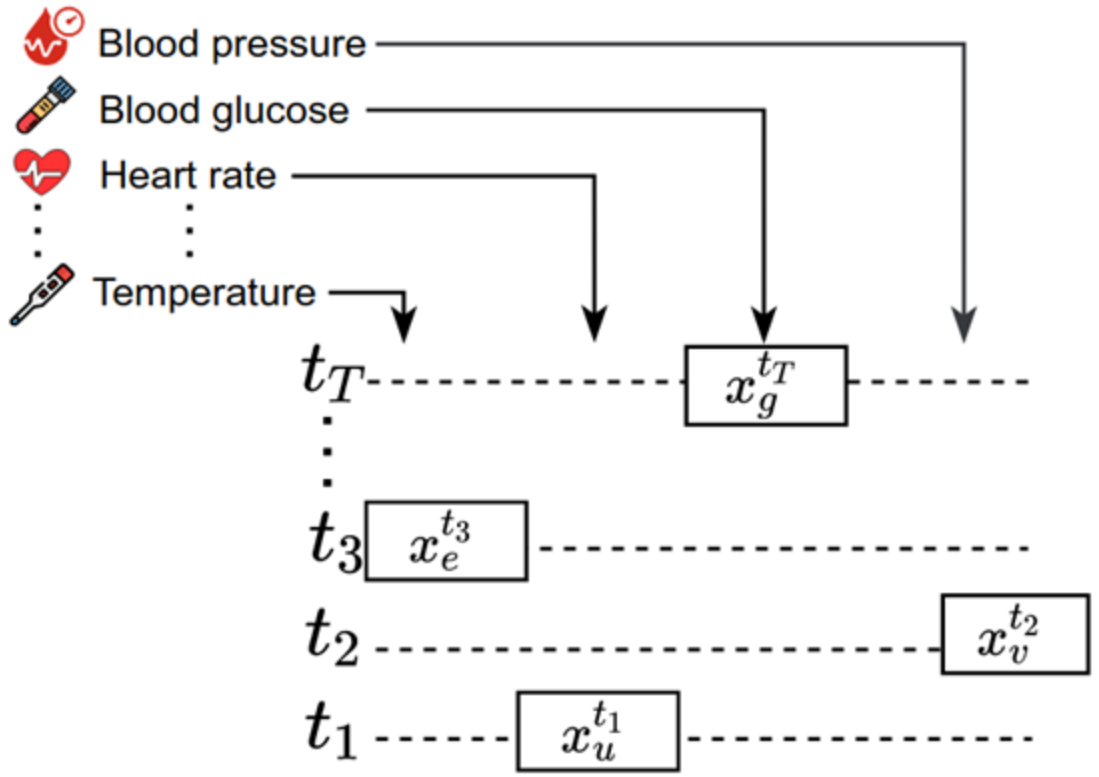
Why are irregular time series challenging?

Prevailing methods:

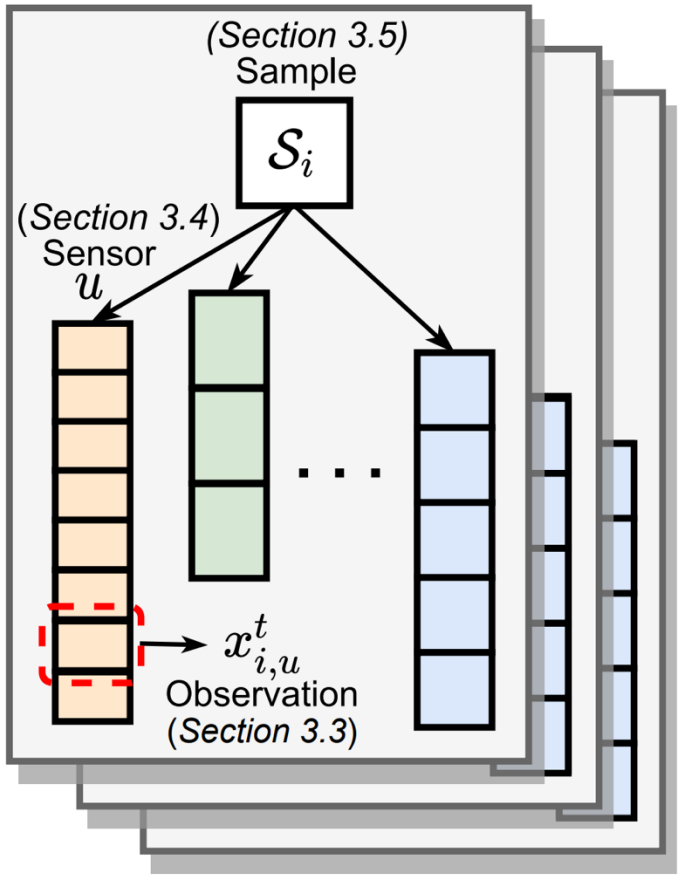
- Assume aligned measurements
- Assume fixed-sized input data
- Impute or fill-in missing values

Irregular time series:

- Observations across sensors are not aligned
- Varying times among adjacent observations
- Arbitrary length: different samples have varyin
- Different subsets of sensors recorded



Problem definition



Input:

- Dataset D of irregular time series samples
- Every sample S_i can have multiple sensors
- Every sensor can have arbitrary number of irregularly sampled observations/readouts

- Raindrop learns a function $f: S_i \rightarrow \boldsymbol{z}_i$ that maps S_i to a fixed-length representation z_i suitable for downstream tasks of interest, such as classification
- Using learned z_i, one can predict label $\hat{y}_i \in \{1, \ldots, C\}$ for S_i

Model irregularity by leveraging sensor dependencies

- Sensors are not independent of each other:
  - Inter-sensor dependencies contain useful information about time series
- Idea: Leverage relational structure among sensors
  - Learn latent graph structures from multivariate time series and model time-varying inter-sensor dependencies through neural message passing
  - Model sample-varying and time-varying relational structure in irregular time series

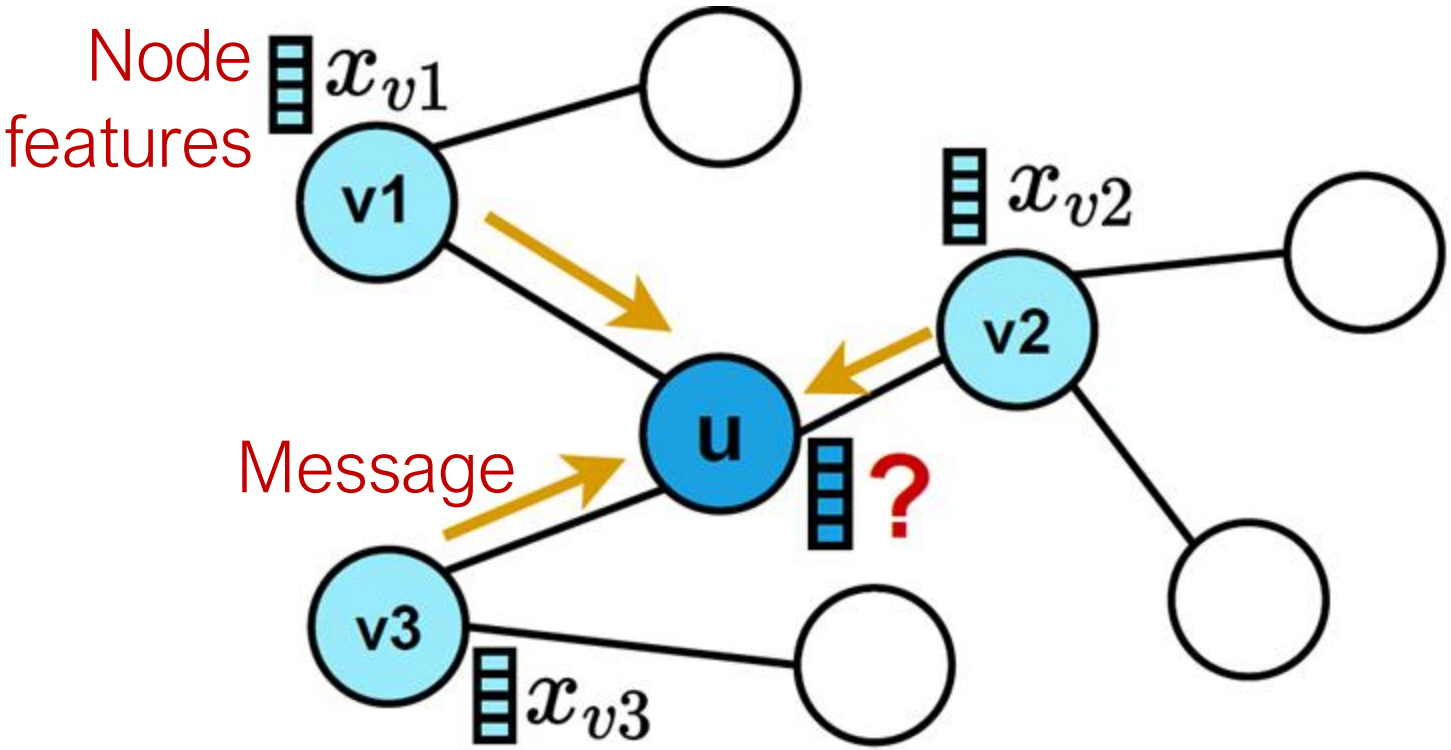
Next: What motivates the use of inter-sensor dependencies?

Raindrop: Observations as “raindrops” hitting a “surface”

- **Observations** (raindrops) hit the **sensor graph** (surface) asynchronously and at irregular times
- Observations are processed by **passing messages to neighboring sensors** (creating ripples), taking into account learned sensor dependencies

Next: How to learn inter-sensor dependencies that can vary across samples and time?

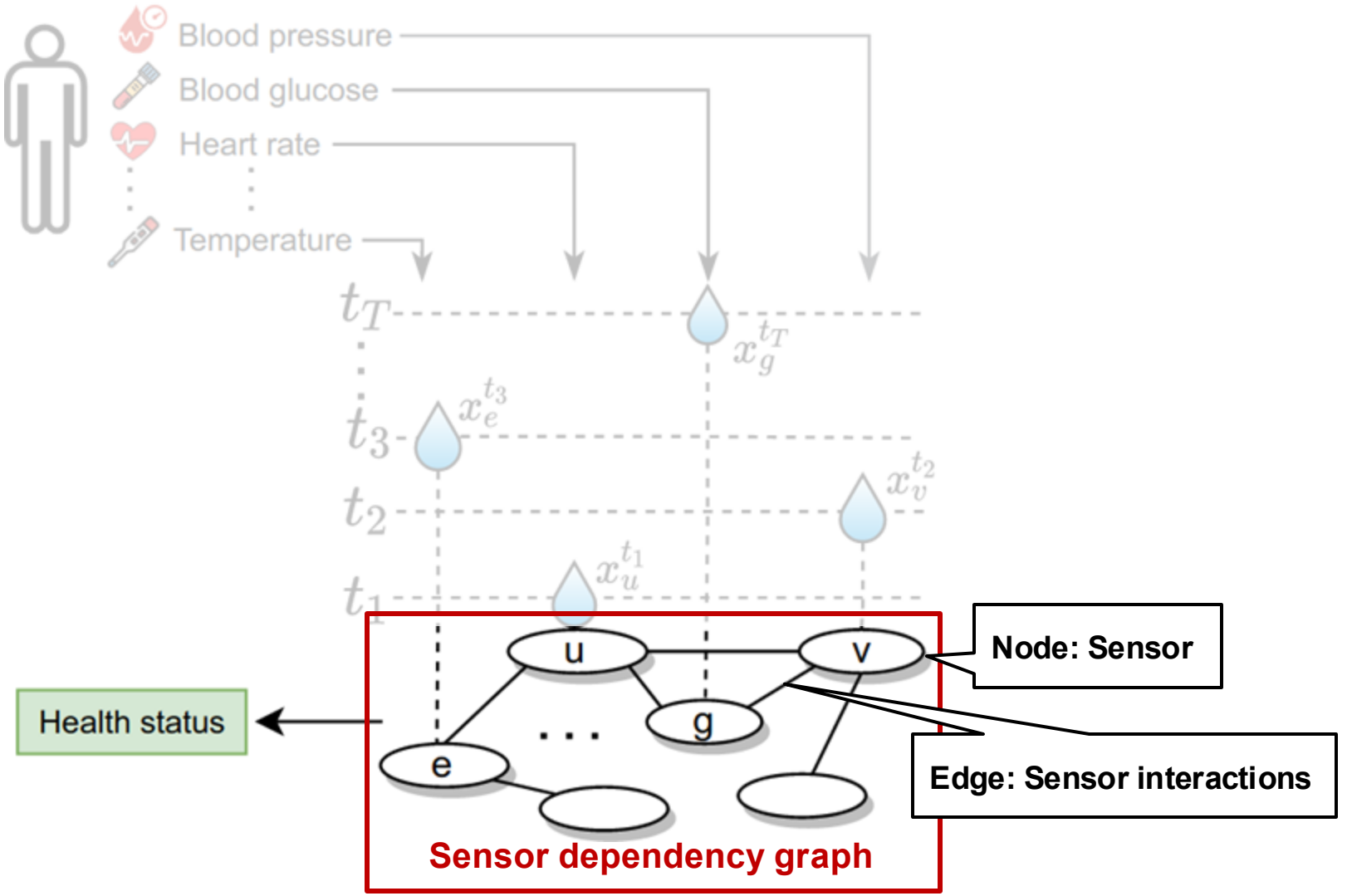
Sensor dependency graphs



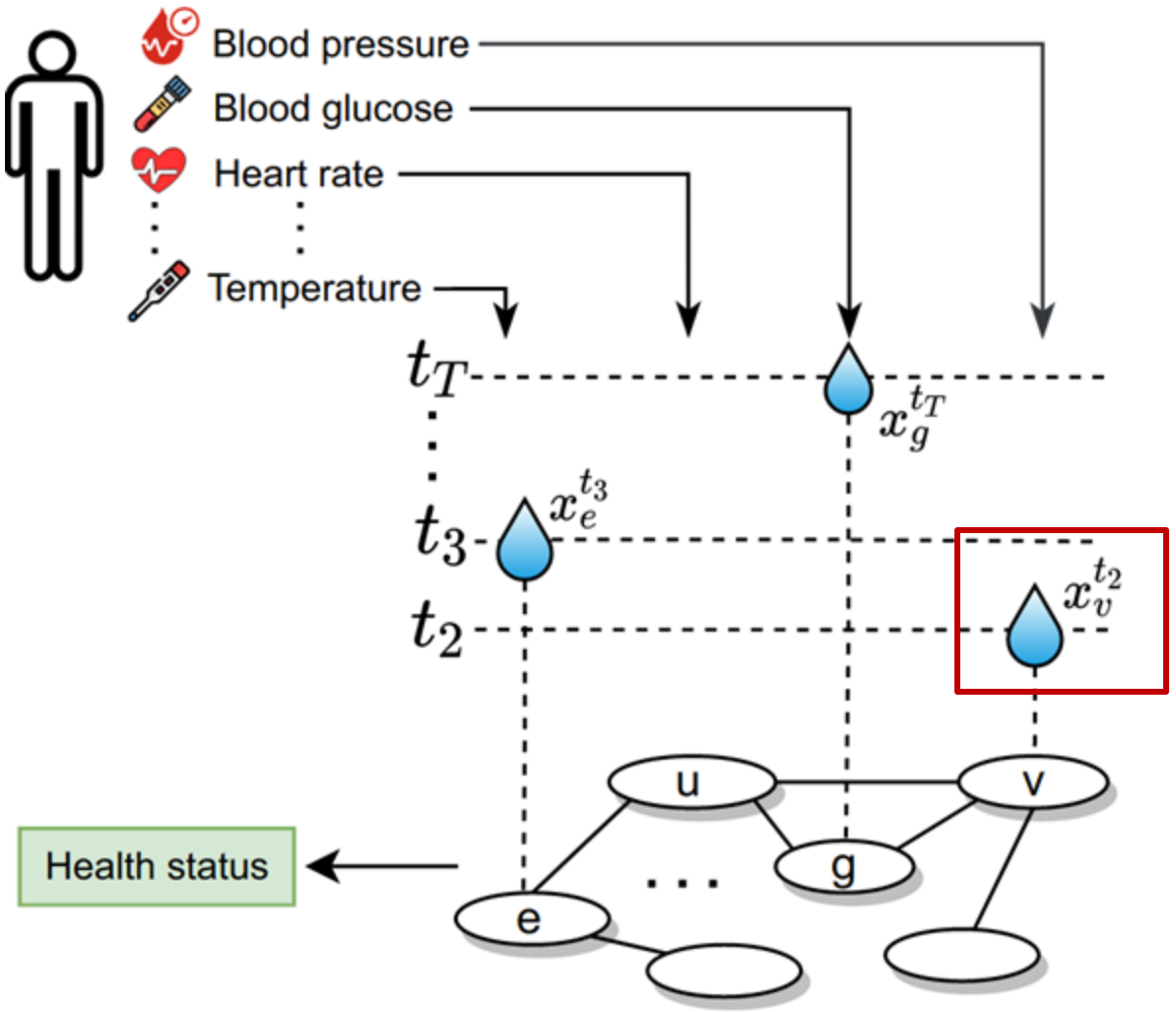
$$x_u = f(x_{v1}, x_{v2}, x_{v3})$$

Generate embedding of node *u* by capturing node dependencies through message passing

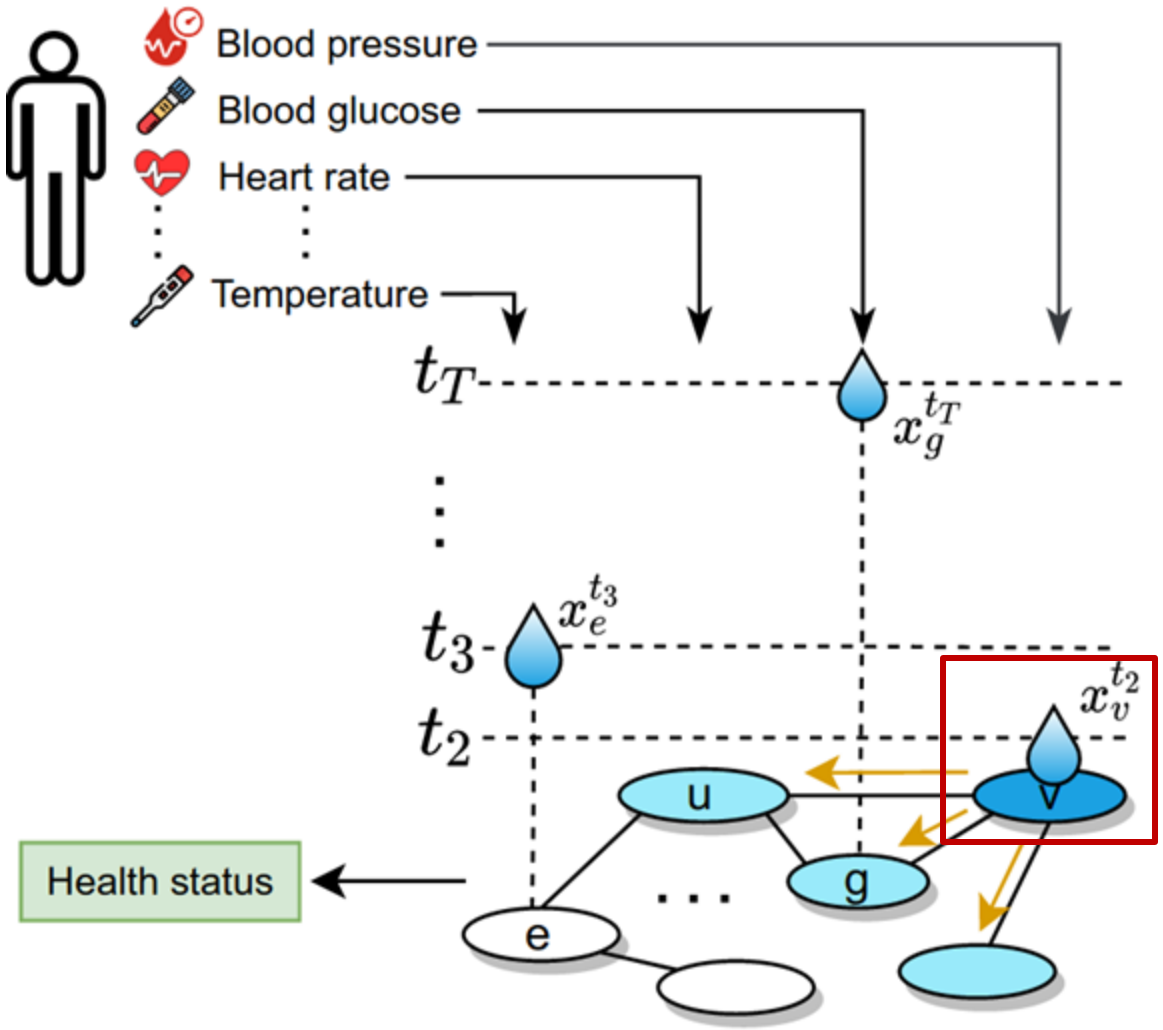
Sensor dependency graphs



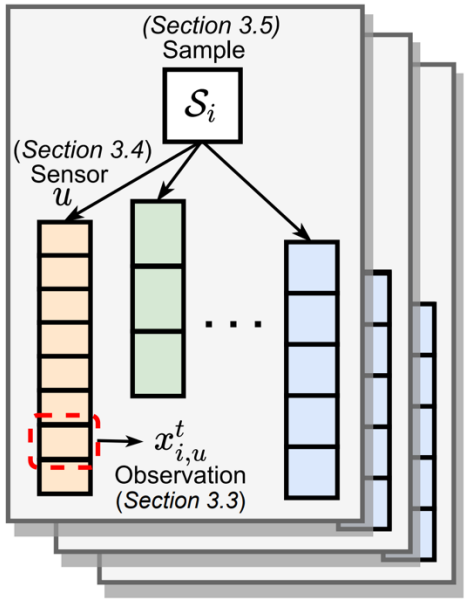
Passing messages between neighboring sensors in every sample



Passing messages between neighboring sensors in every sample



Overview of Raindrop



Hierarchical learning

- Step 1: Construct sensor dependency graphs
  - For every sample, initialize a fully-connected graph
  - During training, update neighbors & edge weights:
    - Graphs are time-sensitive
    - Graphs are sample-sensitive
    - Similar graphs for similar samples
- Step 2: Sensor u is activated = its value is observed
  - Use message passing to generate observations for neighbors of active sensor u
- Step 3: Sensor embeddings
  - For sensor u, aggregate observation embeddings across all timestamps into u's embedding
- Step 4: Sample embedding
  - Gather embeddings across all sensors into a representation of sample S_i using a readout function

Datasets & evaluation setup

- **P19: PhysioNet Sepsis Early Prediction**
  - 40,336 patients, 34 sensors
  - Classification: Sepsis occurring or not

- **P12: PhysioNet Mortality Prediction**
  - 11,988 patients, 36 sensors
  - Classification: Length of stay in the ICU (>3 days or not)

- **PAM: PAMAP2 Physical Activity Monitoring**
  - 5,333 samples, 17 sensors
  - 8-class classification: 8 activities of daily lives

Setting 1/3: Time series classification

Given irregular sensor readouts of a given sample, predict a label for it

Methods	P19		P12		PAM			
	AUROC	AUPRC	AUROC	AUPRC	Accuracy	Precision	Recall	F1 score
Transformer	83.2 ± 1.3	47.6 ± 3.8	65.1 ± 5.6	95.7 ± 1.6	83.5 ± 1.5	84.8 ± 1.5	86.0 ± 1.2	85.0 ± 1.3
Trans-mean	84.1 ± 1.7	47.4 ± 1.4	66.8 ± 4.2	95.9 ± 1.1	83.7 ± 2.3	84.9 ± 2.6	86.4 ± 2.1	85.1 ± 2.4
GRU-D	83.9 ±1.7	46.9 ± 2.1	67.2 ± 3.6	95.9 ± 2.1	83.3 ± 1.6	84.6 ± 1.2	85.2 ± 1.6	84.8 ± 1.2
SeFT	78.7 ± 2.4	31.1 ± 2.8	66.8 ± 0.8	96.2 ± 0.2	67.1 ± 2.2	70.0 ± 2.4	68.2 ± 1.5	68.5 ± 1.8
mTAND	80.4 ± 1.3	32.4 ± 1.8	65.3 ± 1.7	96.5 ± 1.2	74.6 ± 4.3	74.3 ± 4.0	79.5 ± 2.8	76.8 ± 3.4
IP-Net	84.6 ± 1.3	38.1 ± 3.7	72.5 ± 2.4	96.7 ± 0.3	74.3 ± 3.8	75.6 ± 2.1	77.9 ± 2.2	76.6 ± 2.8
DGM^2-O	86.7 ± 3.4	44.7 ± 11.7	71.2 ± 2.5	96.9 ± 0.4	82.4 ± 2.3	85.2 ± 1.2	83.9 ± 2.3	84.3 ± 1.8
MTGNN	81.9 ± 6.2	39.9 ± 8.9	67.5 ± 3.1	96.4 ± 0.7	83.4 ± 1.9	85.2 ± 1.7	86.1 ± 1.9	85.9 ± 2.4
RAINDROP	**87.0 ± 2.3**	**51.8 ± 5.5**	**72.1 ± 1.3**	**97.0 ± 0.4**	**88.5 ± 1.5**	**89.9 ± 1.5**	**89.9 ± 0.6**	**89.8 ± 1.0**

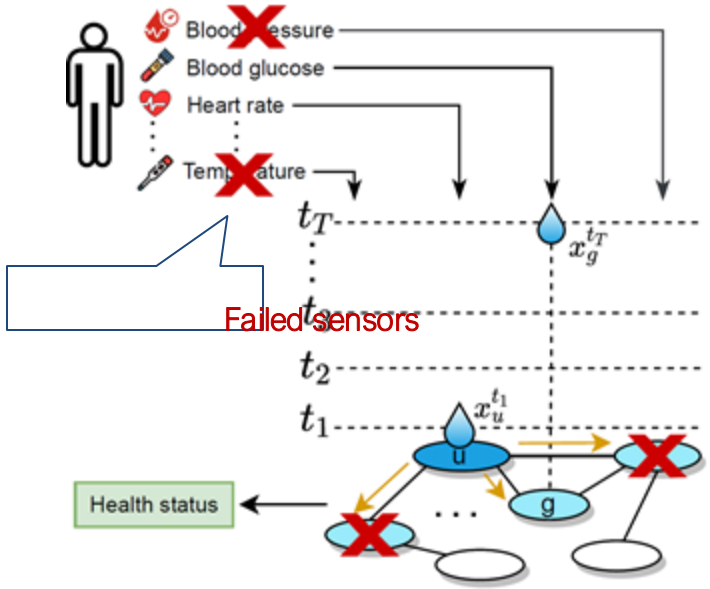
- Raindrop achieves strong performance across three benchmarks
- In binary classification (P19 and P12), it outperforms baselines by 5.3% in AUROC and 4.8% in AUPRC
- In a challenging 8-way classification (PAM), it outperforms baselines by 5.7% in accuracy and 5.5% in F1 score

Setting 2/3: Leave-sensors-out

Dataset P19:

- 38,803 patients, 34 sensors
- Label: Sepsis or not
- Missing sensors: 10-50%

Larger missing rate → Large improvement in performance



Missing rate	Model	Accuracy	Precision	Recall	F1 score
10%	Transformer	60.3 ± 2.4	57.8 ± 9.3	59.8 ± 5.4	57.2 ± 8.0
	Trans-mean	60.4 ± 11.2	61.8 ± 14.9	60.2 ± 13.8	58.0 ± 15.2
	GRU-D	65.4 ± 1.7	72.6 ± 2.6	64.3 ± 5.3	63.6 ± 0.4
	SeFT	58.9 ± 2.3	62.5 ± 1.8	59.6 ± 2.6	59.6 ± 2.6
	mTAND	58.8 ± 2.7	59.5 ± 5.3	64.4 ± 2.9	61.8 ± 4.1
	RAINDROP	**77.2 ± 2.1**	**82.3 ± 1.1**	**78.4 ± 1.9**	**75.2 ± 3.1**
20%	Transformer	63.1 ± 7.6	71.1 ± 7.1	62.2 ± 8.2	63.2 ± 8.7
	Trans-mean	61.2 ± 3.0	**74.2 ± 1.8**	63.5 ± 4.4	64.1 ± 4.1
	GRU-D	64.6 ± 1.8	73.3 ± 3.6	63.5 ± 4.6	64.8 ± 3.6
	SeFT	35.7 ± 0.5	42.1 ± 4.8	38.1 ± 1.3	35.0 ± 2.2
	mTAND	33.2 ± 5.0	36.9 ± 3.7	37.7 ± 3.7	37.3 ± 3.4
	RAINDROP	**66.5 ± 4.0**	72.0 ± 3.9	**67.9 ± 5.8**	**65.1 ± 7.0**
30%	Transformer	31.6 ± 10.0	26.4 ± 9.7	24.0 ± 10.0	19.0 ± 12.8
	Trans-mean	42.5 ± 8.6	45.3 ± 9.6	37.0 ± 7.9	33.9 ± 8.2
	GRU-D	45.1 ± 2.9	51.7 ± 6.2	42.1 ± 6.6	47.2 ± 3.9
	SeFT	32.7 ± 2.3	27.9 ± 2.4	34.5 ± 3.0	28.0 ± 1.4
	mTAND	27.5 ± 4.5	31.2 ± 7.3	30.6 ± 4.0	30.8 ± 5.6
	RAINDROP	**52.4 ± 2.8**	**60.9 ± 3.8**	**51.3 ± 7.1**	**48.4 ± 1.8**
40%	Transformer	23.0 ± 3.5	7.4 ± 6.0	14.5 ± 2.6	6.9 ± 2.6
	Trans-mean	25.7 ± 2.5	9.1 ± 2.3	18.5 ± 1.4	9.9 ± 1.1
	GRU-D	46.4 ± 2.5	**64.5 ± 6.8**	42.6 ± 7.4	44.3 ± 7.9
	SeFT	26.3 ± 0.9	29.9 ± 4.5	27.3 ± 1.6	22.3 ± 1.9
	mTAND	19.4 ± 4.5	15.1 ± 4.4	20.2 ± 3.8	17.0 ± 3.4
	RAINDROP	**52.5 ± 3.7**	53.4 ± 5.6	**48.6 ± 1.9**	**44.7 ± 3.4**
50%	Transformer	21.4 ± 1.8	2.7 ± 0.2	12.5 ± 0.4	4.4 ± 0.3
	Trans-mean	21.3 ± 1.6	2.8 ± 0.4	12.5 ± 0.7	4.6 ± 0.2
	GRU-D	37.3 ± 2.7	29.6 ± 5.9	32.8 ± 4.6	26.6 ± 5.9
	SeFT	24.7 ± 1.7	15.9 ± 2.7	25.3 ± 2.6	18.2 ± 2.4
	mTAND	16.9 ± 3.1	12.6 ± 5.5	17.0 ± 1.6	13.9 ± 4.0
	RAINDROP	**46.6 ± 2.6**	**44.5 ± 2.6**	**42.4 ± 3.9**	**38.0 ± 4.0**

Setting 3/3: Group-wise classification

- **Split by age:** Patients older than 65 years vs. younger patients
- **Split by gender:** Male vs. female patients
- Use one group as training set and randomly split the other group into validation (50%) and test set (50%)

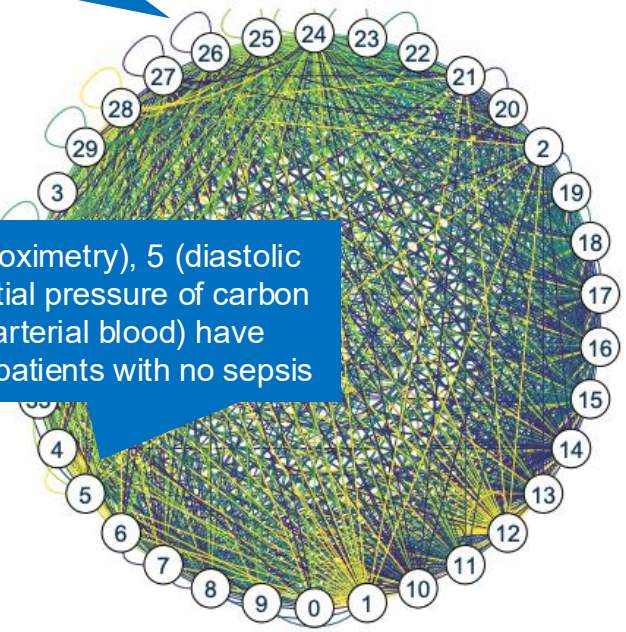
Model	Generalizing to a new patient group							
	Train: Young → Test: Old		Train: Old → Test: Young		Train: Male → Test: Female		Train: Female → Test: Male	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Transformer	76.2 ± 0.7	30.5 ± 4.8	76.5 ± 1.1	33.7 ± 5.7	77.8 ± 1.1	26.0 ± 6.2	75.2 ± 1.0	30.3 ± 5.5
Trans-mean	80.6 ± 1.4	39.8 ± 4.2	78.4 ± 1.1	35.8 ± 2.9	80.2 ± 1.7	32.1 ± 1.9	76.4 ± 0.8	32.5 ± 3.3
GRU-D	76.5 ± 1.7	29.5 ± 2.3	79.6 ± 1.7	35.2 ± 4.6	78.5 ± 1.6	31.9 ± 4.8	76.3 ± 2.5	31.1 ± 2.6
SeFT	77.5 ± 0.7	26.6 ± 1.2	78.9 ± 1.0	32.7 ± 2.7	78.6 ± 0.6	31.1 ± 1.2	76.9 ± 0.5	26.4 ± 1.1
mTAND	79.0 ± 0.8	28.8 ± 2.3	79.4 ± 0.6	29.8 ± 1.2	78.0 ± 0.9	26.5 ± 1.7	78.9 ± 1.2	29.2 ± 2.0
RAINDROP	**83.2 ± 1.6**	**43.6 ± 4.7**	**82.0 ± 4.4**	**44.3 ± 3.6**	**85.0 ± 1.4**	**45.2 ± 2.9**	**81.2 ± 3.8**	**40.7 ± 2.9**

- Strong results in across-group scenarios; e.g., Raindrop outperforms strongest baseline by 4.8% in AUROC and 13.1% in AUPRC when asked to generalize to female patients
- Raindrop can generalize to new samples unseen during training:
  - Sensor dependency graphs are sample-specific and estimated using a sample's observations
  - Raindrop can adaptively generate sensor dependencies based on readouts of a test sample

Can Raindrop learn dynamics of sensors?

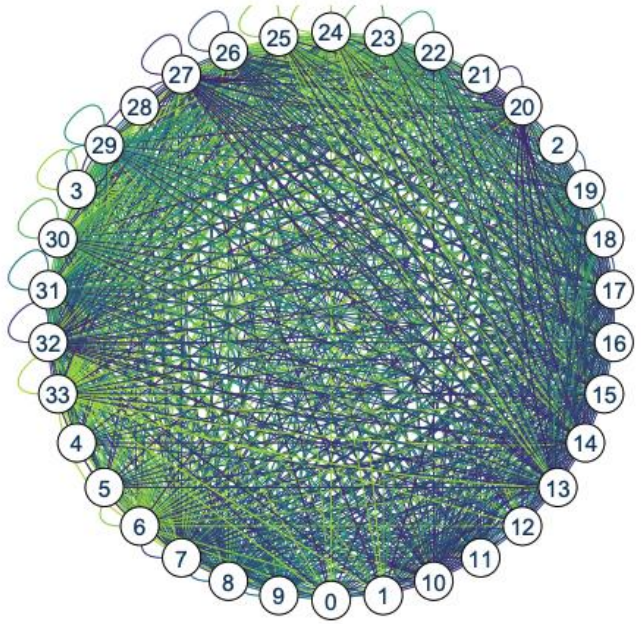
Nodes: sensors

Nodes 1 (pulse oximetry), 5 (diastolic BP), and 12 (partial pressure of carbon dioxide from arterial blood) have lower weights in patients with no sepsis



Patients with no sepsis

Sensor dependency graphs averaged across negative samples



Patients who develop sepsis

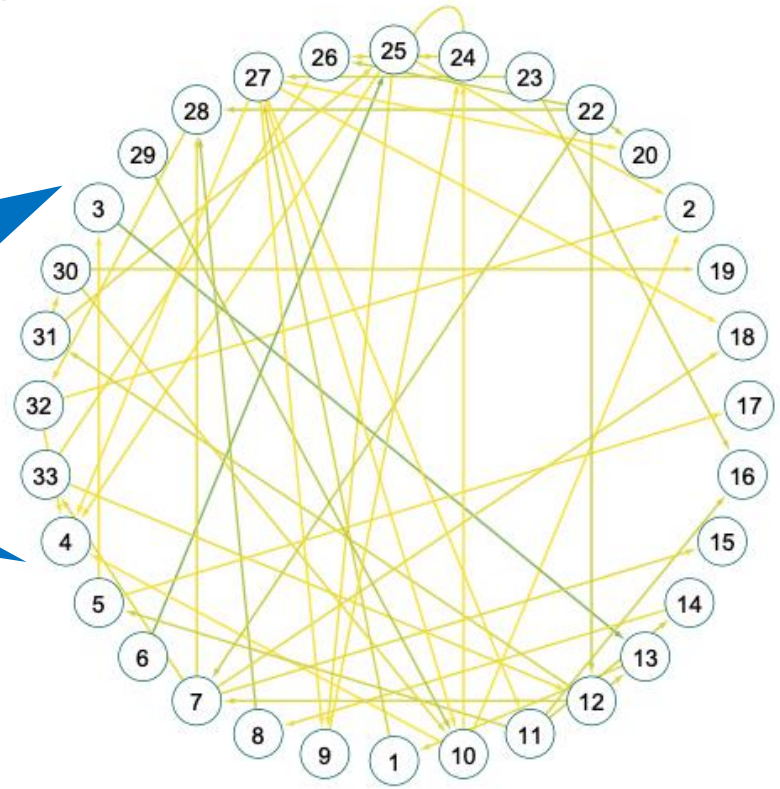
Sensor dependency graphs averaged across positive samples

Distinguishable patterns between graphs of negative and positive samples → Raindrop learns dynamics of sensors purely from observational data

Can Raindrop learn dynamics of sensors?

Edges from node 3 (systolic BP) to node 13 (oxygen saturation from arterial blood) and node 6 (respiration rate) to node 25 (potassium) are informative for distinguishing between w/ and w/o sepsis

Node 0 (heart rate) is not connected to any others sensor, indicating that heart rate carries little predictive value for the onset of sepsis



Show is differential inter-sensor graph between patients who will likely develop sepsis and those who won't

Edges are colored by the divergences; darker colors denote sensor dependencies that are more crucial to patient classification

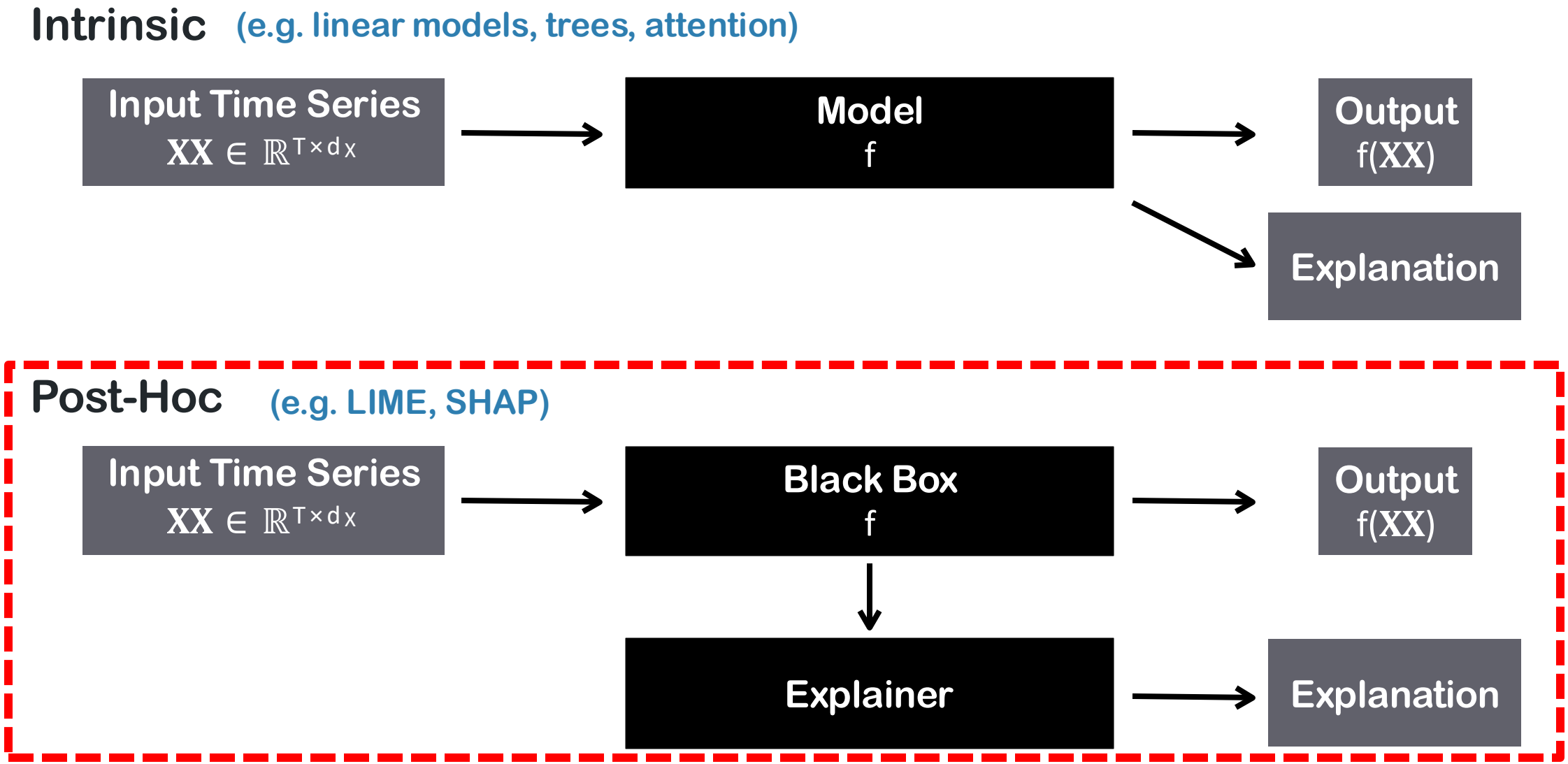
Irregular time series: Recap

- Irregular time series: Raindrop addresses the complexity of time series, e.g., misaligned observations, varying time gaps & varying numbers of observations per sensor
- Inter-sensor structure: Raindrop adopts neural message passing to model inter-sensor dependencies in irregular time series
- Great generalization: Raindrop has excellent performance in challenging settings, including setups where a subset of sensors malfunction (i.e., have no readouts at all)

Today's lecture

1. **Time series tasks in healthcare**
2. **Self-supervised pre-training for time series**
3. **Learning representations** of regular and irregular time series
4. **Understanding time series models**

Intrinsic vs. post-hoc interpretability



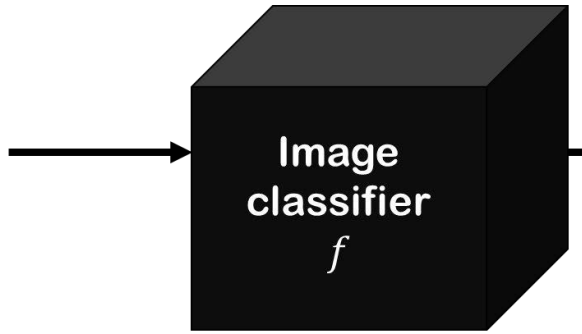
Standard feature importance scoring methods

Highlight most important features for the model

- **Integrated Gradient [Sundararajan et al. 2017]**

$$a_i(f, \boldsymbol{x}) = \left(x_i - x_i^0\right) \times \int_0^1 \frac{\partial f[\boldsymbol{x}^0 + t(\boldsymbol{x} - \boldsymbol{x}^0)]}{\partial x_i} dt$$

$\boldsymbol{x}$ = Image



$f(\boldsymbol{x})$ = Dog

- **SHAP [Lundberg et al. 2017]**

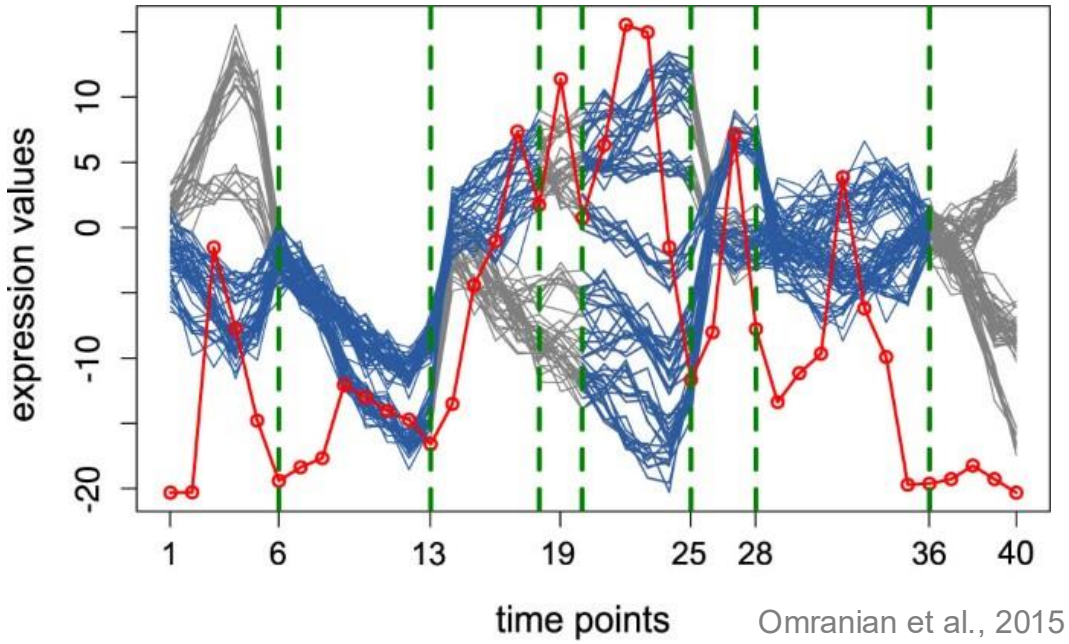
$$a_i(f, \boldsymbol{x}) = \sum_{S \subset [\dim \mathcal{X}] \setminus \{i\}} \frac{|S|!\,(\dim \mathcal{X} - |S| - 1)}{(\dim \mathcal{X})!} \left[f\left(\boldsymbol{x}_{S \cup \{i\}}\right) - f(\boldsymbol{x}_S)\right]$$

"Standard" feature importance methods perform poorly for time-series

[Ismail et al., NeurIPS 2020]

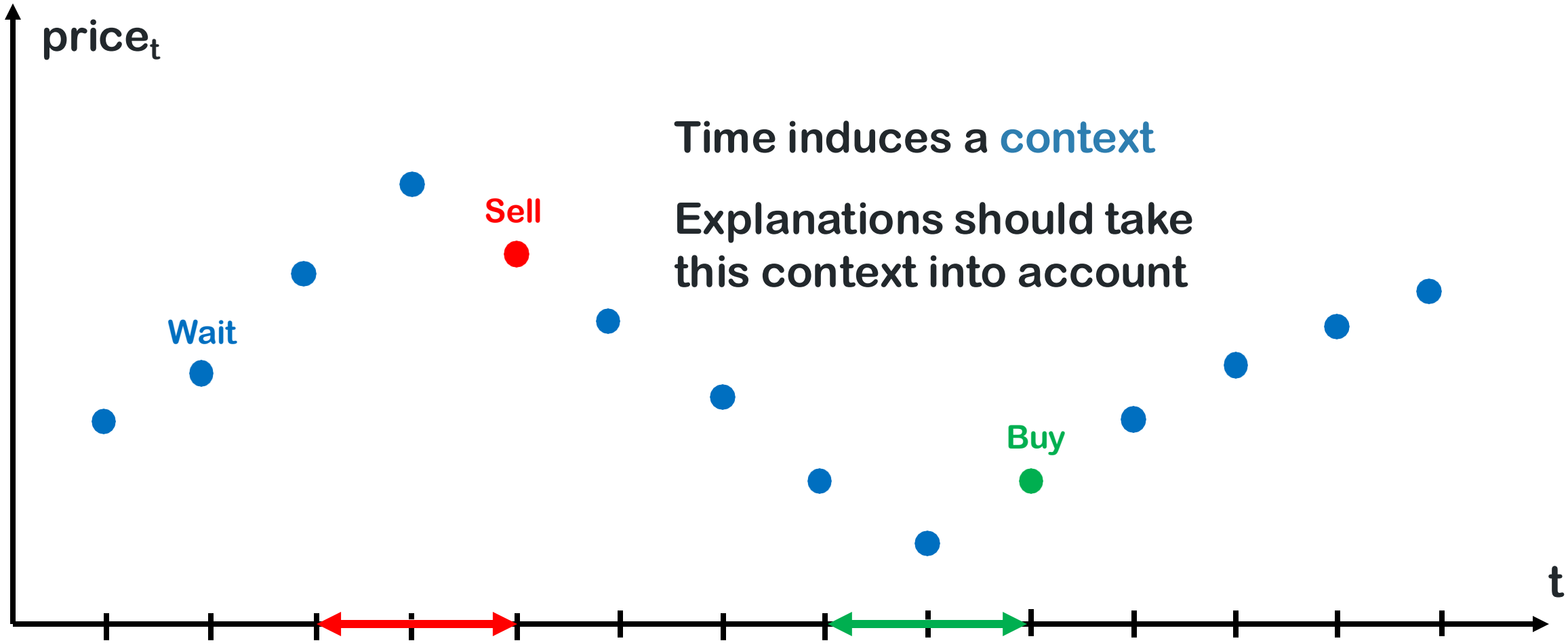
Challenges for explaining time series

- Not easily visually interpretable
  - Noisy samples
  - Dense informative features, unlike imaging and text modalities
- **Temporal patterns**
  - Only show up when looking at time segments and long-term behaviors
- Perturbations matter
  - Setting a value to zero does not ignore that time point
  - Temporal dependencies cannot be ignored



Omranian et al., 2015

What makes time series datasets different?



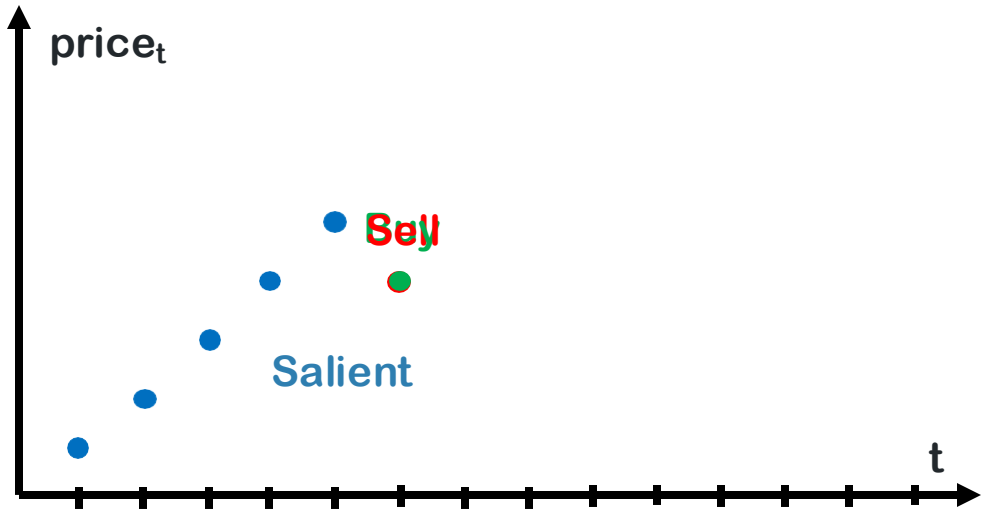
How to detect salient features?

Perturbation based detection

Premise: salient features affect the model's prediction

Detect salient features by feature perturbations

Feature perturbation affects prediction → Salient feature



How to take the time context into account?

Time context matters

Typical saliency methods treat each input $x_{t,i}$ as a feature

$\Longrightarrow$ Time dependency is ignored by the saliency method

Dynamic Perturbation Operator

Idea: perturb each $x_{t^*,i}$ by using neighbouring times:

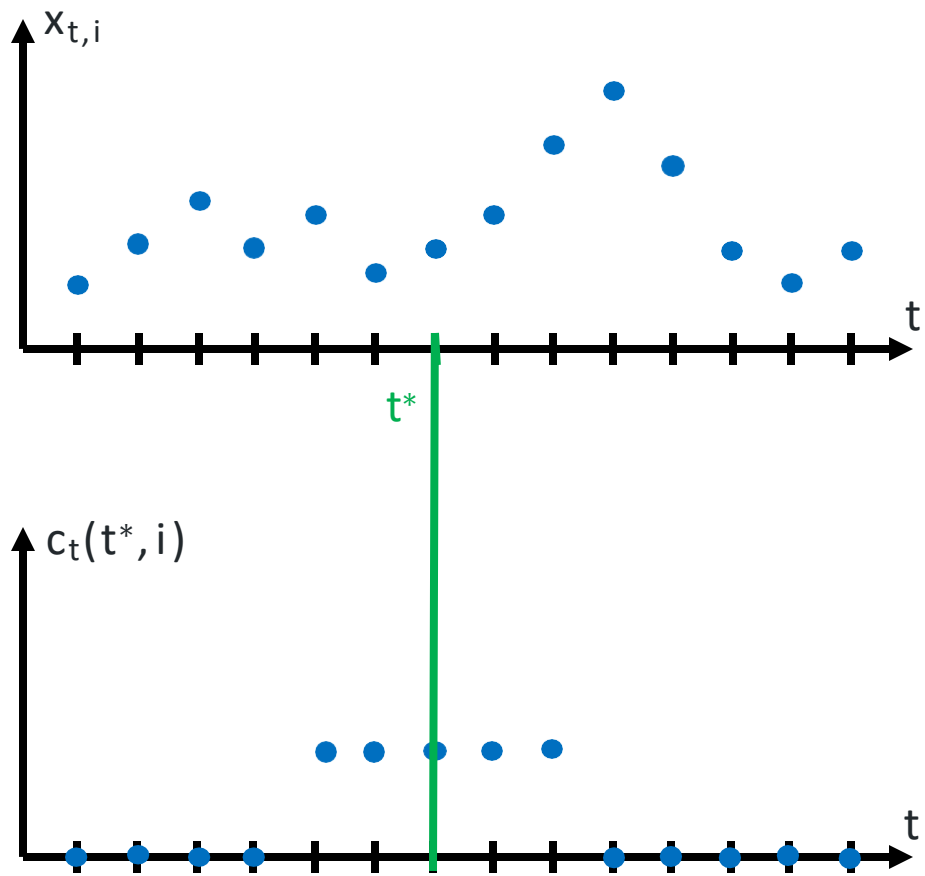
Perturbed input

$$\pi\left(x_{t^*,i}\,; t^*, i\right) = \sum_{t=t^*-W_1}^{t^*+W_2} c_t(t^*, i) \times x_{t,i}$$

Linear combination

$\Longrightarrow$ Time dependency is integrated in perturbation

Window perturbation:



[Crabbé, van der Schaar, ICML 2021]

How to take the time context into account?

Time context matters

Typical saliency methods treat each input $x_{t,i}$ as a feature

$\Longrightarrow$ Time dependency is ignored by the saliency method

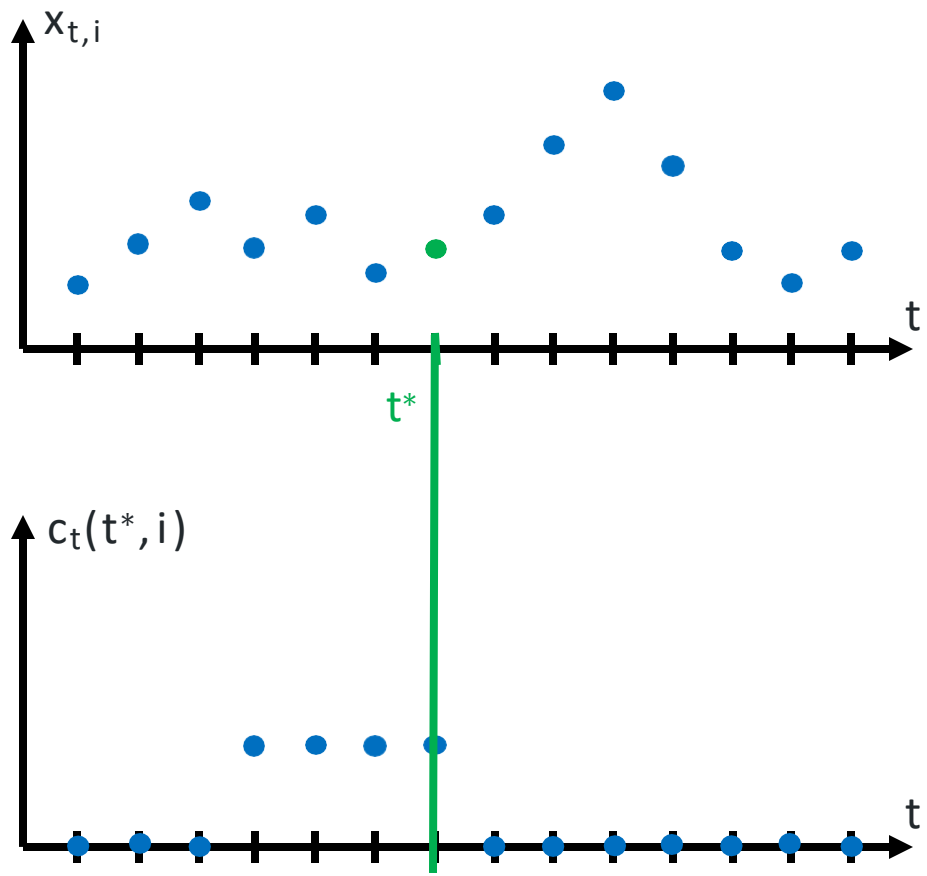
Dynamic Perturbation Operator

Idea: perturb each $x_{t^*,i}$ by using neighbouring times:

Perturbed input: $\pi(x_{t^*,i}; t^*, i)$ = Linear combination: $\sum_{t=t^*-W_1}^{t^*+W_2} c_t(t^*, i) \times x_{t,i}$

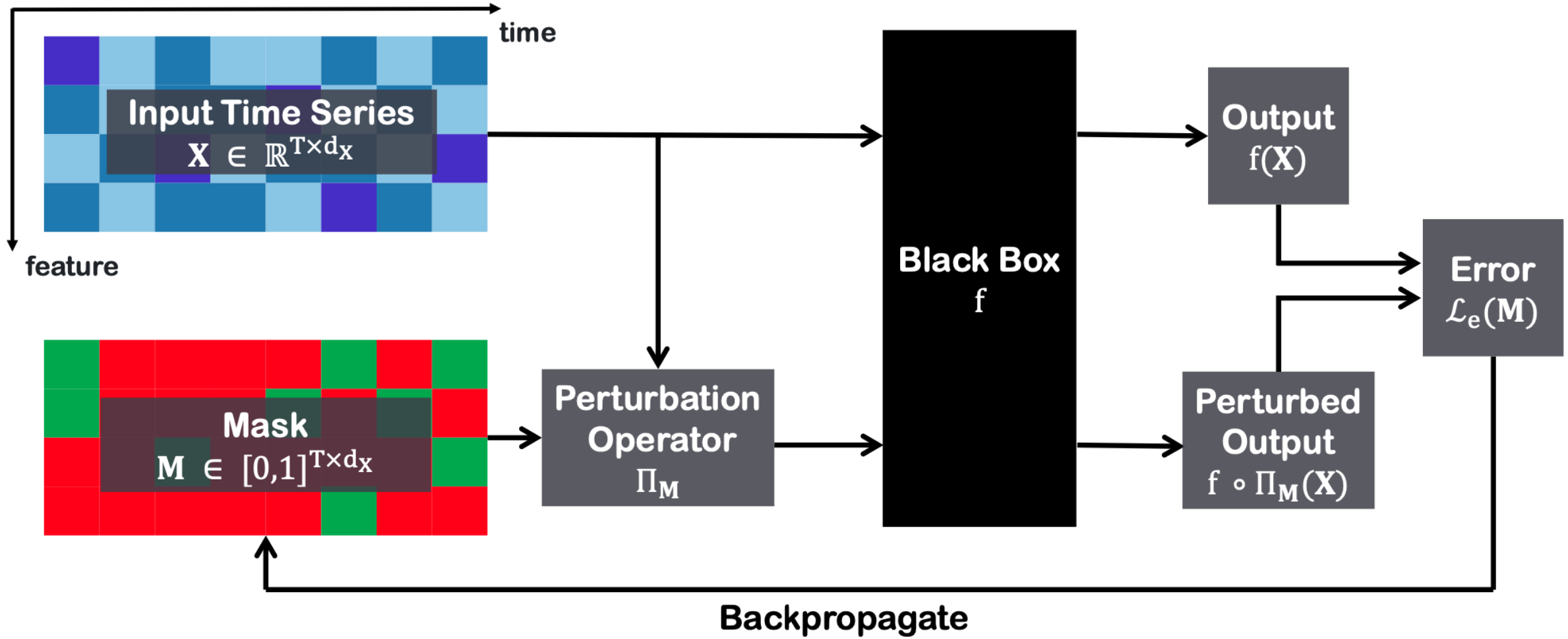
$\Longrightarrow$ Time dependency is integrated in perturbation

Past window perturbation:



[Crabbé, van der Schaar, ICML 2021]

Dynamask



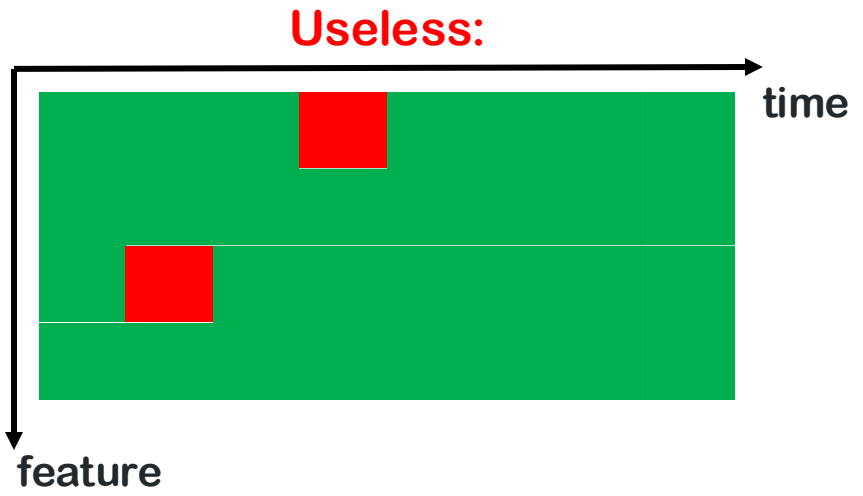
[Crabbé, van der Schaar, ICML 2021]

How to make saliency masks parsimonious?

What do we mean by parsimonious?

Masks should not highlight more features than necessary

⟹ We need to enforce feature selection



How to enforce parsimony?

The user selects the fraction a of most important features

We add a regularization to enforce sparsity:

$$\mathcal{L}_a(\mathbf{M}) = \|\text{vecsort}(\mathbf{M}) - \mathbf{r}_a\|^2$$

Sets the $(1-a) \times T \times d_X$ smallest mask coefficients to zero

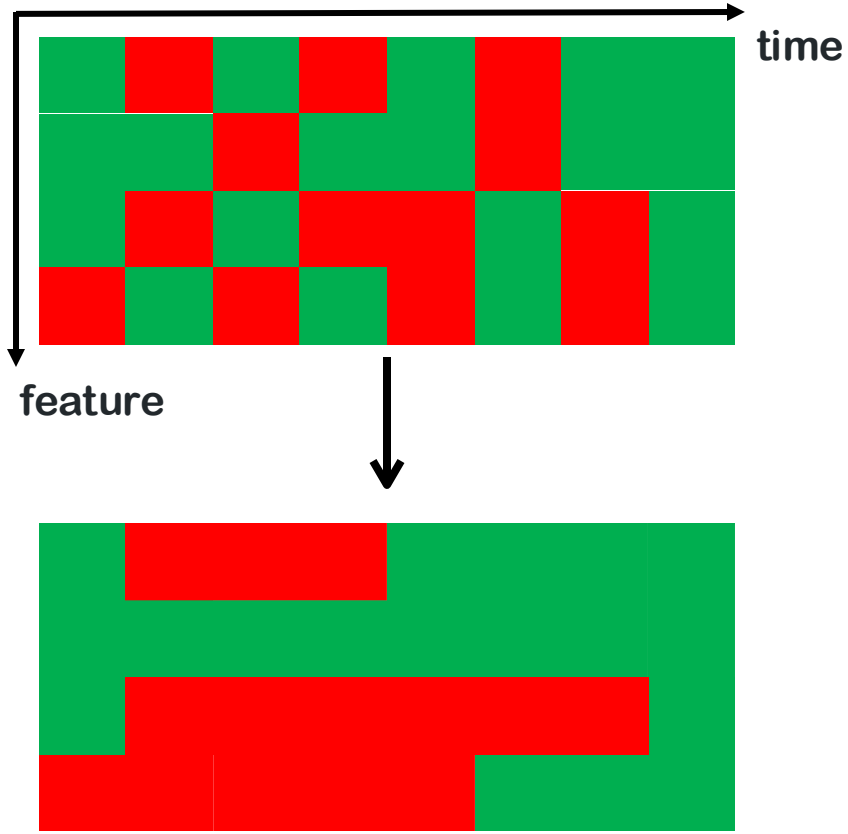
[Crabbé, van der Schaar, ICML 2021]

How to avoid quick variation in the saliency mask?

Quick time variations of the saliency

Might want to avoid **quick time variations** of the saliency

This can be a **prior belief** or a **preference** of the user



How to avoid this?

We add a regularization to penalize saliency jumps over time:

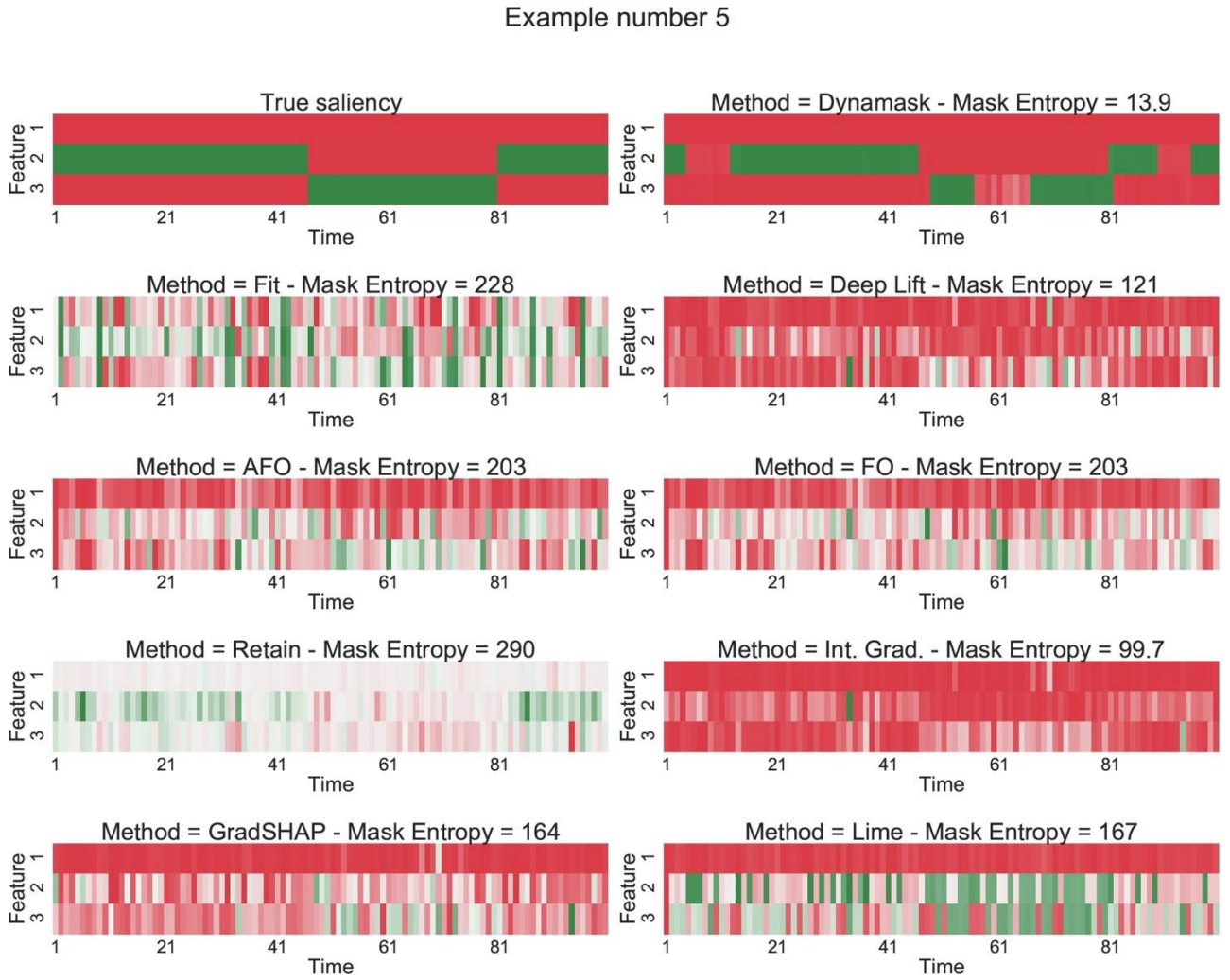
$$\mathcal{L}_c(\mathbf{M}) = \sum_{t=1}^{T-1} \sum_{i=1}^{d_X} |m_{t+1,i} - m_{t,i}|$$

[Crabbé, van der Schaar, ICML 2021]

Dynamask - Example

True saliency

Baseline saliency

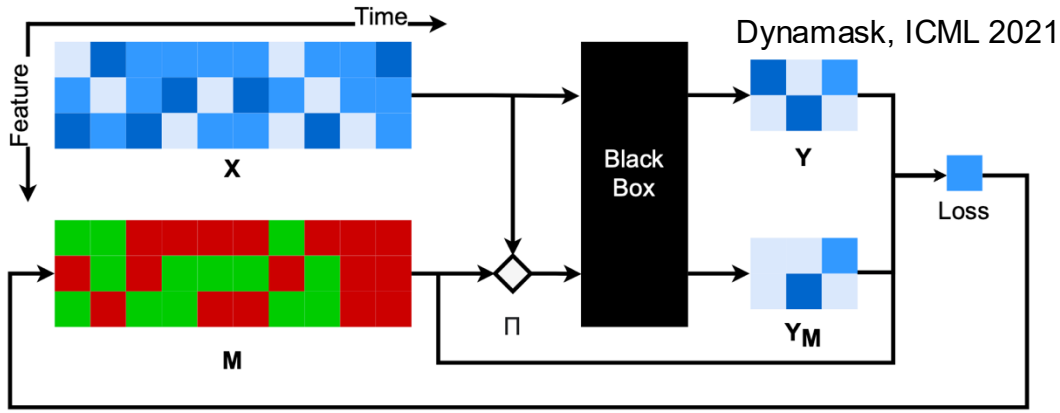


Dynamask saliency

[Crabbé, van der Schaar, ICML 2021]

Existing time series explainers are inadequate

1. **Perturbations are continuous**
   - Can deform shape of samples
2. **Give only instance-based explanations**
   - Cannot relate patterns across samples
3. **Fail to match performance of generic explainers**
   - Post-hoc methods suffer from a lack of faithfulness and stability



Dynamask, ICML 2021

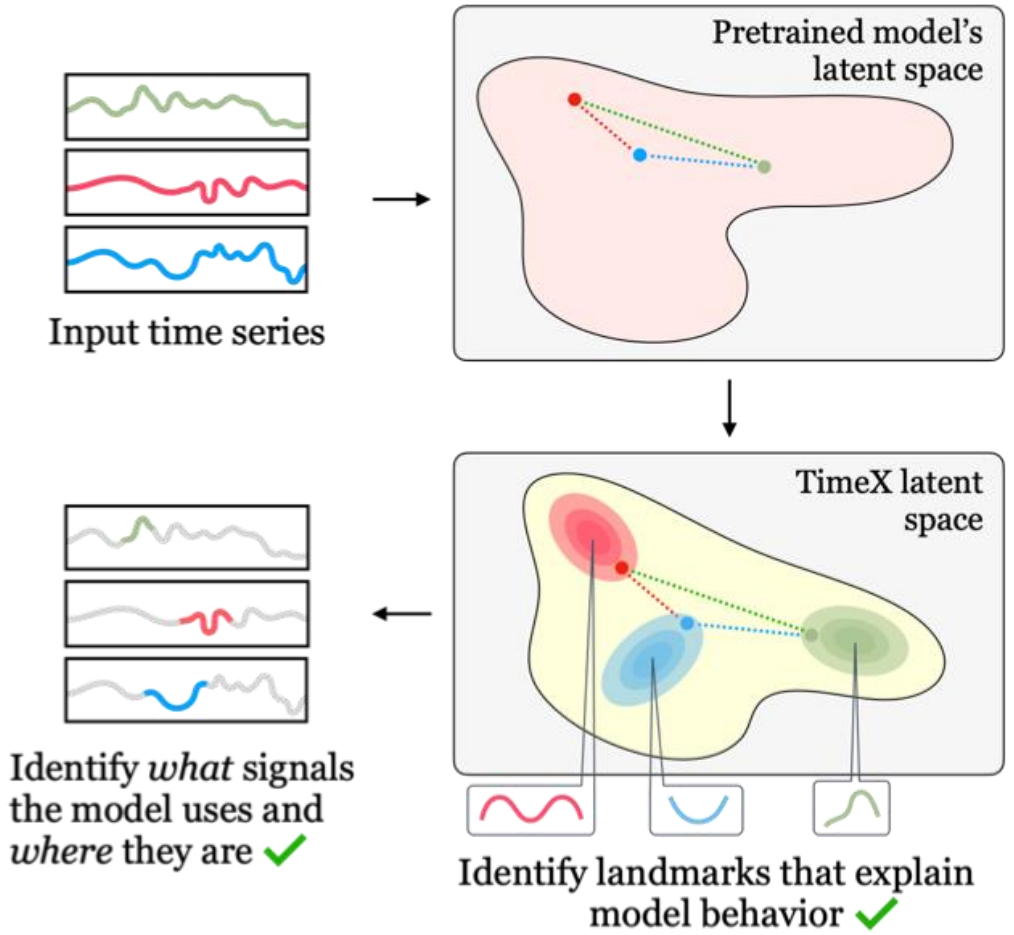
Desiderata for time series explanations

- Temporally connected and visually digestible
- Identify the location of predictive time series signals and underlying interpretable patterns
- Connect explanations across samples

Encoding Time-Series Explanations through Self-Supervised Model Behavior Consistency, NeurIPS 2023

TimeX is a time-series consistency explainer

- Surrogate model to **mimic the behavior of a pretrained time series model**
- TimeX makes inferences on masked samples
- **Model behavior consistency**
  - Enforces faithfulness at the level of the latent space
  - Learns a flexible latent space of explanations

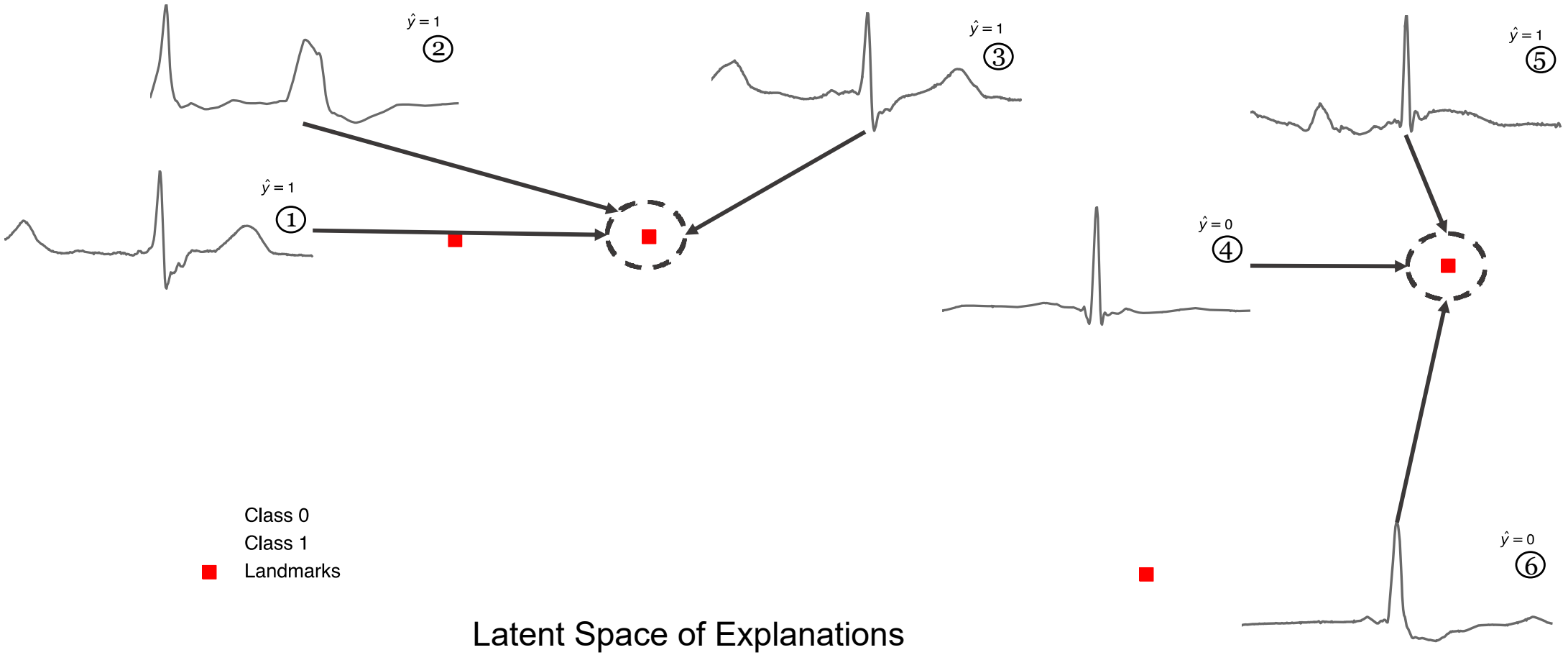


TimeX learns highly-faithful explanations

Method	SeqComb-MV AUPRC	SeqComb-MV AUP	SeqComb-MV AUR	LowVar AUPRC	LowVar AUP	LowVar AUR
IG	0.3298±0.0015	<u>0.7483</u>±0.0027	0.2581±0.0028	**0.8691**±0.0035	<u>0.4827</u>±0.0029	<u>0.8165</u>±0.0016
Dynamask	0.3136±0.0019	0.5481±0.0053	0.1953±0.0025	0.1391±0.0012	0.1640±0.0028	0.2106±0.0018
WinIT	0.2809±0.0018	0.7594±0.0024	0.2077±0.0021	0.1667±0.0015	0.1140±0.0022	0.3842±0.0017
CoRTX	0.3629±0.0021	0.5625±0.0006	0.3457±0.0017	<u>0.4983</u>±0.0014	0.3281±0.0027	0.4711±0.0013
SGT + Grad	<u>0.4893</u>±0.0005	0.4970±0.0005	**0.4289**±0.0018	0.3449±0.0010	0.2133±0.0029	0.3528±0.0015
TIMEX	**0.6878**±0.0021	**0.8326**±0.0008	<u>0.3872</u>±0.0015	**0.8673**±0.0033	**0.5451**±0.0028	**0.9004**±0.0024

Method	ECG AUPRC	ECG AUP	ECG AUR
IG	<u>0.4182</u>±0.0014	**0.5949**±0.0023	0.3204±0.0012
Dynamask	0.3280±0.0011	0.5249±0.0030	0.1082±0.0080
WinIT	0.3049±0.0011	0.4431±0.0026	<u>0.3474</u>±0.0011
CoRTX	0.3735±0.0008	0.4968±0.0021	0.3031±0.0009
SGT + Grad	0.3144±0.0010	0.4241±0.0024	0.2639±0.0013
TIMEX	**0.4721**±0.0018	<u>0.5663</u>±0.0025	**0.4457**±0.0018

Landmarks learn important patterns in ECG

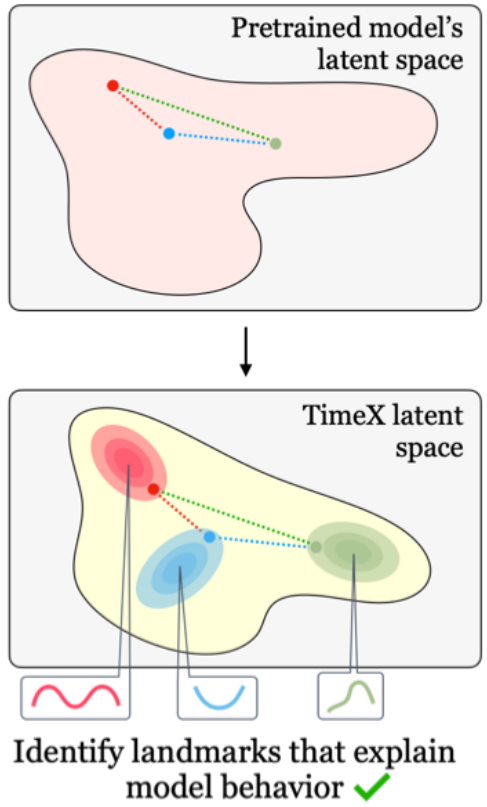


Latent Space of Explanations

Landmarks partition the latent space of explanations into interpretable temporal patterns

Explanations through self-supervised model behavior consistency: Recap

- TimeX is a SOTA time series explainer
- STEs learn discrete masks that represent explanations and prevent shortcut learning
- **Model behavior consistency** preserves faithfulness across latent spaces
- Landmarks find important common temporal patterns to increase interpretability

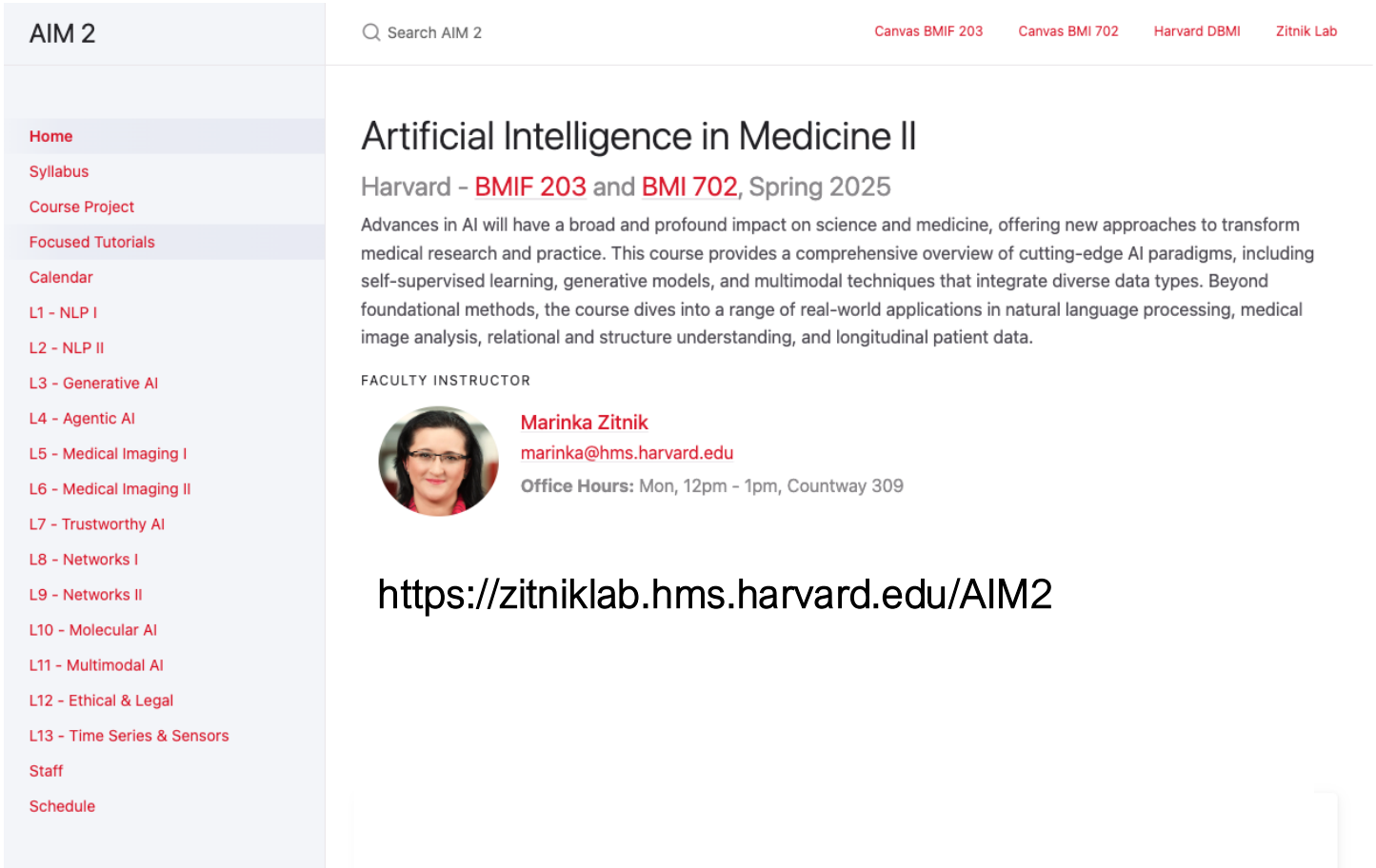


Today's lecture

1. **Time series tasks in healthcare**
2. **Self-supervised pre-training for time series**
3. **Learning representations** of regular and irregular time series
4. **Understanding time series models**

Thank you for an incredible semester!

Final project presentations: See you all tomorrow (Wed) from 9am-12pm



https://zitniklab.hms.harvard.edu/AIM2

This is just the beginning
You now have the tools to bridge AI and medicine to ask deeper questions, to build more powerful models, and to reimagine what is possible for human health

The future of AI in medicine will be shaped by those who think critically, work collaboratively, and innovate responsibly. Keep pushing the boundaries of what we know. Stay grounded in science, but never lose sight of imagination

The discoveries you make can change lives

I look forward to seeing the breakthroughs you will create