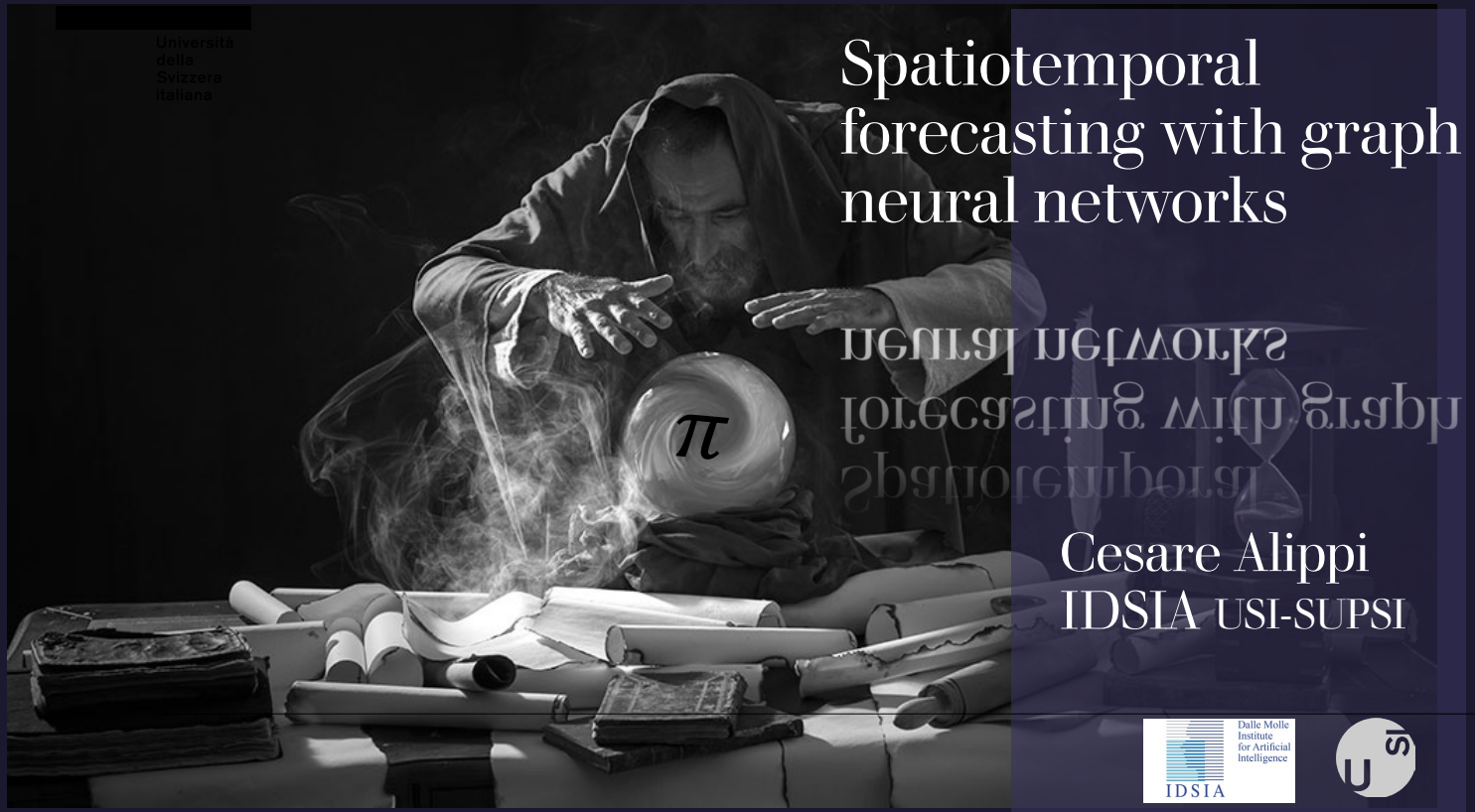
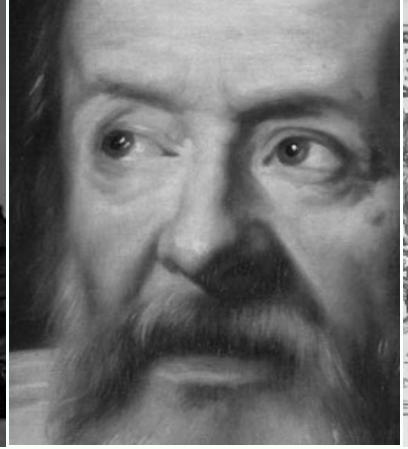


Spatiotemporal forecasting with graph neural networks

Cesare Alippi
IDSIA USI-SUPSI

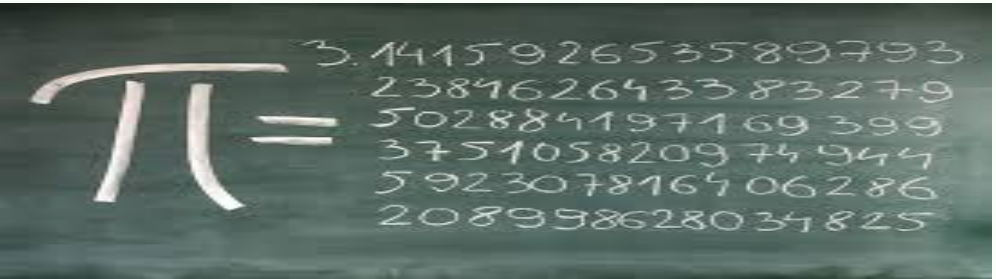
Dalle Molle Institute for Artificial Intelligence
IDSIA


The secret dream: get a glimpse of the divine through the way the world gets ruled

Scientific approach:
algorithms that “see” into the future

Sensors, measurements
Assumptions and family of models
Model design and validation



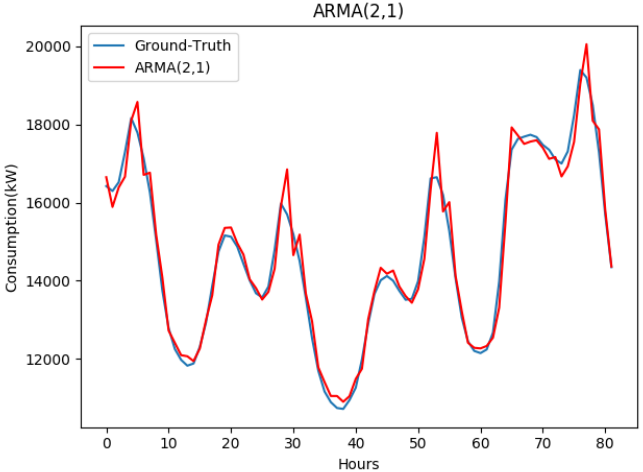
The future is just a step ahead…

Predictive models

Time invariant family of models

Linear:
AR, MA, ARMA, OE, ARX, ARMAX, ARIMA(TV), State space, Kalman filters…

Nonlinear:
RNN, GP, TCN, SEQ2SEQ, LSTM, GRU, TRANSFORMERS…



Predictive models

Exploting space&time dependencies

Exploit the existence of a –possibly latent- manifold in **space and time**

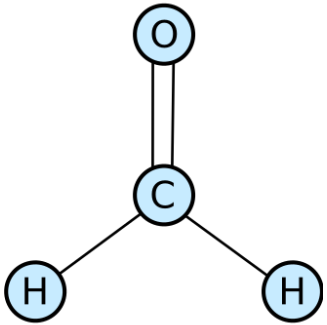
Inductive bias: functional/relational dependencies

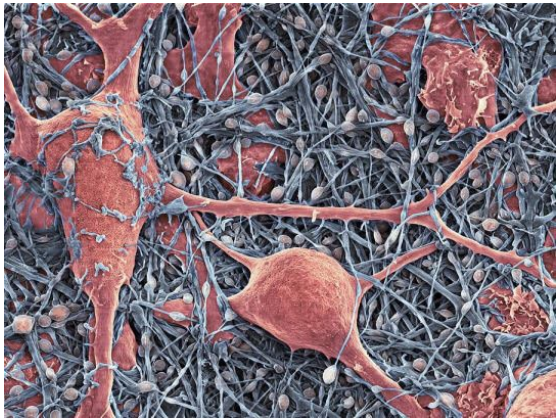
Presentation outline

- Why graphs
- AI-based graph processing
- Some projects

Why graphs

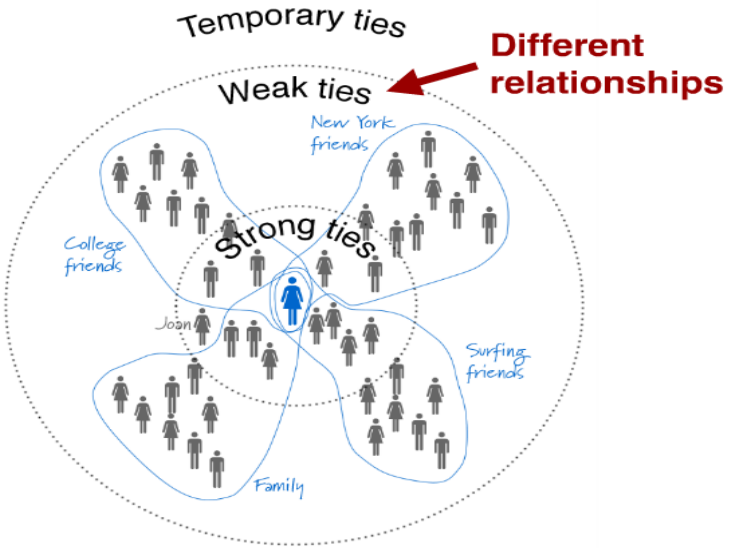
In many applications graphs come naturally





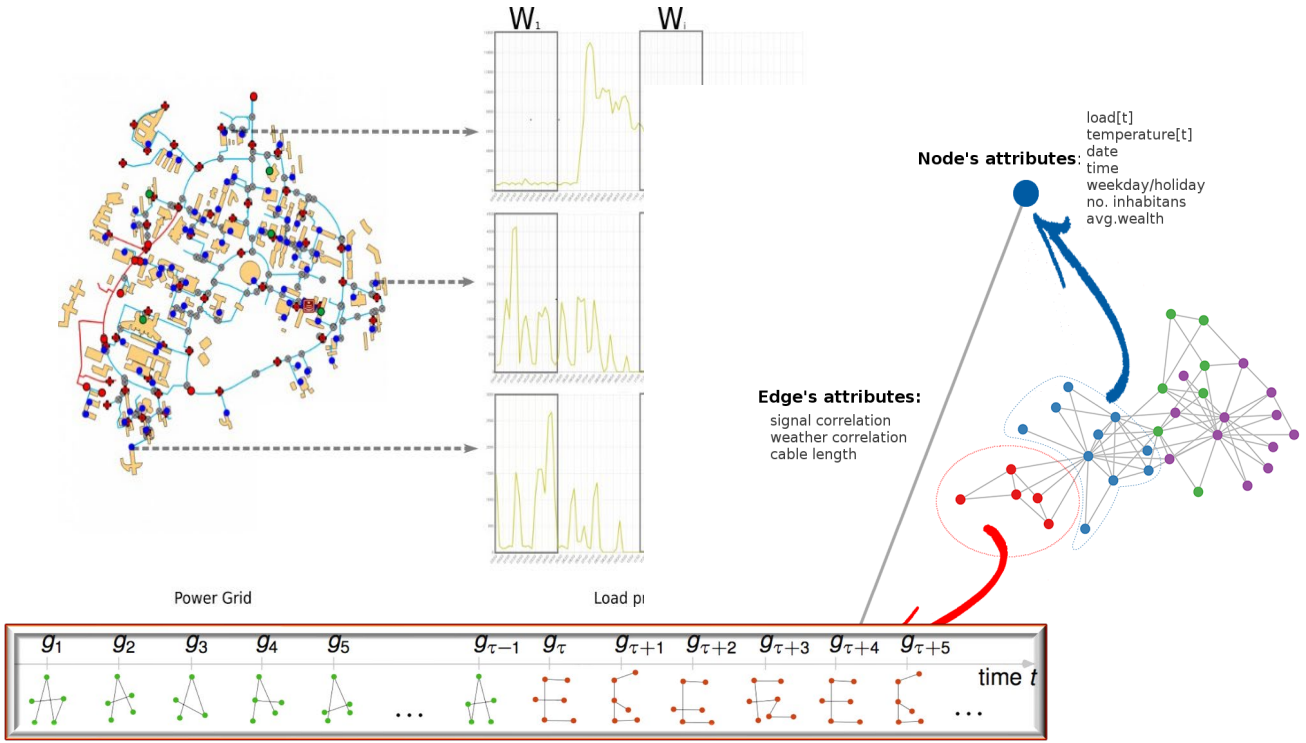
Why graphs

In others are latent



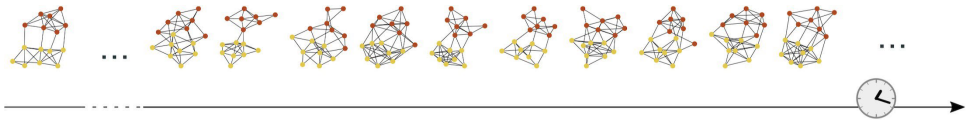
Why graphs

In others, we can derive graphs from time series (signals)

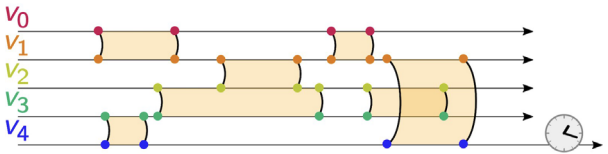


Graph representation of temporal signals

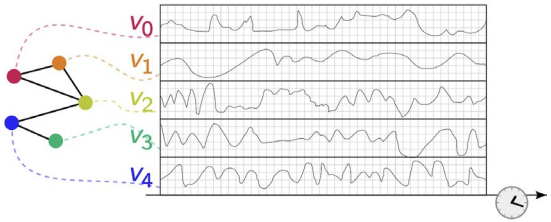
There are several settings in which time comes into play when considering graph data.



Graph streams



Dynamic Graphs

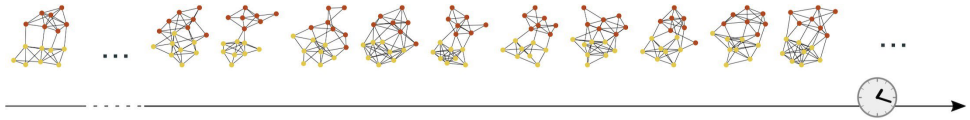


Spatiotemporal graph signals

Image credits: Daniele Zambon.

Graph streams

The most general setting

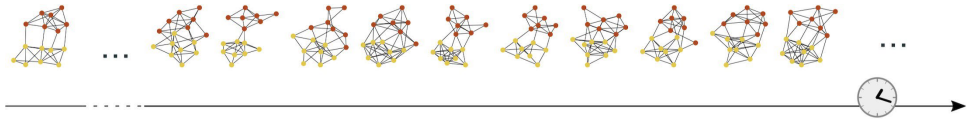


- Graphs are sampled from a stochastic process $G_t \sim P$.
- Nodes are not identified.
- Arbitrary changes in topology.
- Difficult to track changes, defining statistics is not trivial.

[1] D. Zambon, "Anomaly and Change Detection in Sequences of Graphs", 2022.

Graph streams

We start from the most general setting.

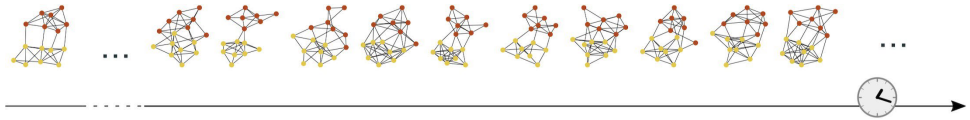


- Graphs are sampled from a stochastic process $G_t \sim P$.
- Nodes are not identified.
- Arbitrary changes in topology.
- Difficult to track changes, defining statistics is not trivial.

[1] D. Zambon, "Anomaly and Change Detection in Sequences of Graphs", 2022.

Graph streams

We start from the most general setting.

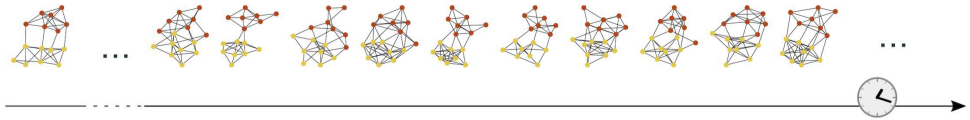


- Graphs are sampled from a stochastic process $G_t \sim P$.
- Nodes are not identified. → No correspondence between nodes at different time steps.
- Arbitrary changes in topology.
- Difficult to track changes, defining statistics is not trivial.

[1] D. Zambon, "Anomaly and Change Detection in Sequences of Graphs", 2022.

Graph streams

We start from the most general setting.

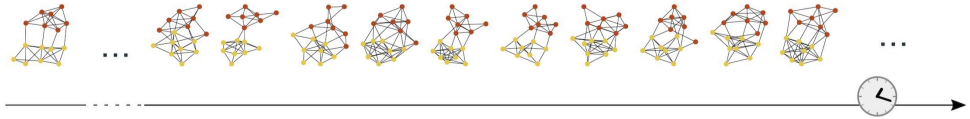


- Graphs are sampled from a stochastic process $G_t \sim P$.
- Nodes are not identified.
- Arbitrary changes in topology.
- Difficult to track changes, defining statistics is not trivial.

[1] D. Zambon, "Anomaly and Change Detection in Sequences of Graphs", 2022.

Graph streams

We start from the most general setting.

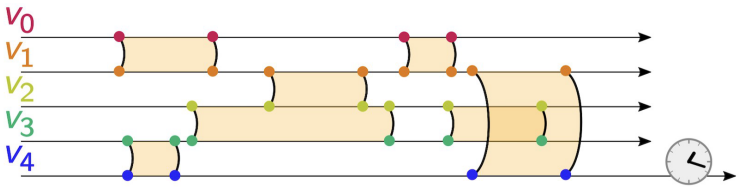


- Graphs are sampled from a stochastic process $G_t \sim P$.
- Nodes are not identified.
- Arbitrary changes in topology.
- Difficult to track changes, defining statistics is not trivial.

[1] D. Zambon, "Anomaly and Change Detection in Sequences of Graphs", 2022.

Dynamic graphs

We look at interactions that happen over time as sequences of **relational events**.

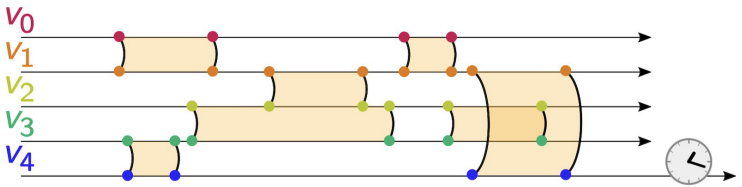


- Dynamic graphs model systems where relationships and node attributes evolve with time.
- Event-based paradigm: dynamic graphs model sequences of interactions among nodes.
- Powerful paradigm to model social/interaction networks and build recommender systems.

[2] E. Rossi *et al.*, "Temporal graph networks for deep learning on dynamic graphs", 2020.

Dynamic graphs

We can look at interactions that happen over time as sequences of **relational events**.

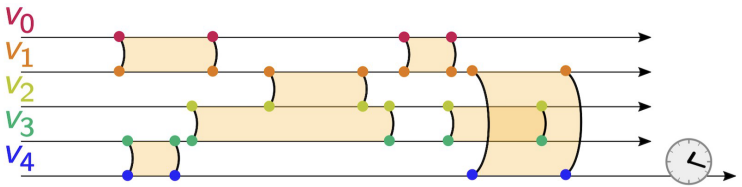


- Dynamic graphs model systems where relationships and node attributes evolve over time.
- Event-based paradigm: dynamic graphs model sequences of interactions among nodes.
- Powerful paradigm to model social/interaction networks and build recommender systems.

[2] E. Rossi *et al.*, "Temporal graph networks for deep learning on dynamic graphs", 2020.

Dynamic graphs

We can look at interactions that happen over time as sequences of **relational events**.



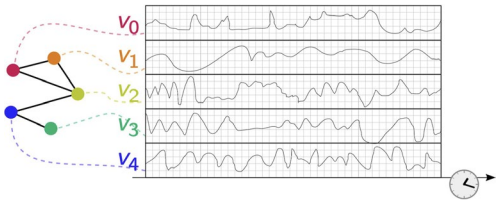
- Dynamic graphs model systems where relationships and node attributes evolve over time.
- Event-based paradigm: dynamic graphs model sequences of interactions among nodes.
- Powerful paradigm to model social/interaction networks and build recommender systems.

[2] E. Rossi *et al.*, "Temporal graph networks for deep learning on dynamic graphs", 2020.

Spatiotemporal graphs

Spatiotemporal graphs capture the setting typical of sensor networks.

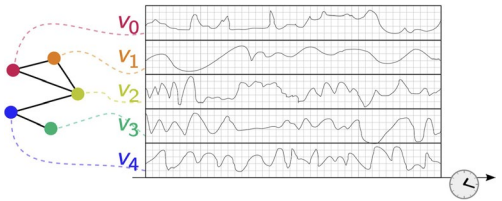
- A different approach to model multivariate time series coming from multiple sources.
- Each node (sensor) is associated with a time series (eventually with multiple channels)
- Edges describe (soft) functional dependencies among sensors
  - E.g.: causality, physical constraints, etc.



Spatiotemporal graphs

Spatiotemporal graphs capture the setting typical of sensor networks.

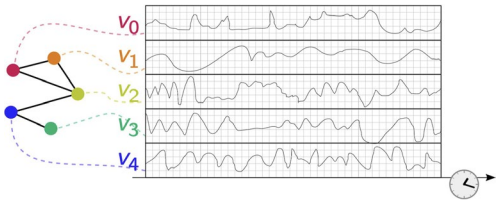
- A different approach to model multivariate time series coming from multiple sources.
- Each node (sensor) is associated with a time series (eventually with multiple channels)
- Edges describe (soft) functional dependencies among sensors
  - E.g.: causality, physical constraints, etc.



Spatiotemporal graphs

Spatiotemporal graphs capture the setting typical of sensor networks.

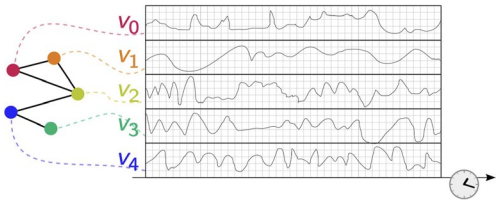
- A different approach to model multivariate time series coming from multiple sources.
- Each node (sensor) is associated with a time series (eventually with multiple channels)
- Edges describe (soft) functional dependencies among sensors
  - E.g.: causality, physical constraints, etc.



Spatiotemporal graphs

Spatiotemporal graphs capture the setting typical of sensor networks.

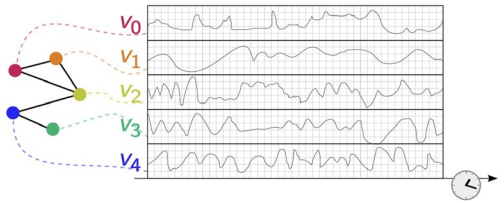
- A different approach to model multivariate time series coming from multiple sources.
- Each node (sensor) is associated with a time series (eventually with multiple channels)
- Edges describe (soft) functional dependencies among sensors
  - E.g.: causality, physical constraints, etc.



Spatiotemporal graphs

Spatiotemporal graphs capture the setting typical of sensor networks.

- A different approach to model multivariate time series coming from multiple sources.
- Each node (sensor) is associated with a time series (eventually with multiple channels)
- Edges describe (soft) functional dependencies among sensors
  - E.g.: causality, physical constraints, etc.



We focus on this setting

Spatiotemporal graph signals

Spatiotemporal graph signal

We consider a **graph signal** as a tuple $\mathbb{G}_t = \langle \boldsymbol{A}_t, \boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{E}_t \rangle$ where

- $\boldsymbol{A}_t \in \mathbb{R}^{N_t \times N_t}$ is a weighted adjacency matrix, with N_t being the number of nodes;
- $\boldsymbol{X}_t \in \mathbb{R}^{N_t \times d_x}$ is the node-attribute matrix;
  - $\boldsymbol{x}_t^i$ (the i-th row of $\boldsymbol{X}_t$) is the d_x-dimensional attribute vector associated with the i-th node;
- $\boldsymbol{U}_t \in \mathbb{R}^{N_t \times d_u}$ are exogenous variables (e.g., weather forecasts, datetime information);
- $\boldsymbol{E}_t \in \mathbb{R}^{E_t \times d_e}$ is an edge-attribute matrix, with E_t being the number of edges;

We focus on settings where the number of nodes is constant over time, i.e., $N_t = N$.

[3] A. Cini *et al.*, "Taming Local Effects in Graph-based Spatiotemporal Forecasting", 2023.

Spatiotemporal graph signal

We consider a **graph signal** as a tuple $\mathbb{G}_t = \langle \boldsymbol{A}_t, \boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{E}_t \rangle$ where

- $\boldsymbol{A}_t \in \mathbb{R}^{N_t \times N_t}$ is a weighted adjacency matrix, with N_t being the number of nodes;
- $\boldsymbol{X}_t \in \mathbb{R}^{N_t \times d_x}$ is the node-attribute matrix;
  - $\boldsymbol{x}_t^i$ (the i-th row of $\boldsymbol{X}_t$) is the d_x-dimensional attribute vector associated with the i-th node;
- $\boldsymbol{U}_t \in \mathbb{R}^{N_t \times d_u}$ are exogenous variables (e.g., weather forecasts, datetime information);
- $\boldsymbol{E}_t \in \mathbb{R}^{E_t \times d_e}$ is an edge-attribute matrix, with E_t being the number of edges;

We focus on settings where the number of nodes is constant over time, i.e., $N_t = N$.

[3] A. Cini *et al.*, "Taming Local Effects in Graph-based Spatiotemporal Forecasting", 2023.

Spatiotemporal graph signal

We consider a **graph signal** as a tuple $\mathbb{G}_t = \langle \boldsymbol{A}_t, \boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{E}_t \rangle$ where

- $\boldsymbol{A}_t \in \mathbb{R}^{N_t \times N_t}$ is a weighted adjacency matrix, with N_t being the number of nodes;
- $\boldsymbol{X}_t \in \mathbb{R}^{N_t \times d_x}$ is the node-attribute matrix;
  - $\boldsymbol{x}_t^i$ (the i-th row of $\boldsymbol{X}_t$) is the d_x-dimensional attribute vector associated with the i-th node;
- $\boldsymbol{U}_t \in \mathbb{R}^{N_t \times d_u}$ are exogenous variables (e.g., weather forecasts, datetime information);
- $\boldsymbol{E}_t \in \mathbb{R}^{E_t \times d_e}$ is an edge-attribute matrix, with E_t being the number of edges;

We focus on settings where the number of nodes is constant over time, i.e., $N_t = N$.

[3] A. Cini *et al.*, "Taming Local Effects in Graph-based Spatiotemporal Forecasting", 2023.

Spatiotemporal graph signal

We consider a **graph signal** as a tuple $\mathbb{G}_t = \langle \boldsymbol{A}_t, \boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{E}_t \rangle$ where

- $\boldsymbol{A}_t \in \mathrm{R}^{N_t \times N_t}$ is a weighted adjacency matrix, with N_t being the number of nodes;
- $\boldsymbol{X}_t \in \mathrm{R}^{N_t \times d_x}$ is the node-attribute matrix;
  - $\boldsymbol{x}_t^i$ (the i-th row of $\boldsymbol{X}_t$) is the d_x-dimensional attribute vector associated with the i-th node;
- $\boldsymbol{U}_t \in \mathrm{R}^{N_t \times d_u}$ are exogenous variables (e.g., weather forecasts, datetime information);
- $\boldsymbol{E}_t \in \mathrm{R}^{E_t \times d_e}$ is an edge-attribute matrix, with E_t being the number of edges;

We focus on settings where the number of nodes is constant over time, i.e., $N_t = N$.

[3] A. Cini *et al.*, "Taming Local Effects in Graph-based Spatiotemporal Forecasting", 2023.

Spatiotemporal graph signal

We consider a **graph signal** as a tuple $\mathbb{G}_t = \langle \boldsymbol{A}_t, \boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{E}_t \rangle$ where

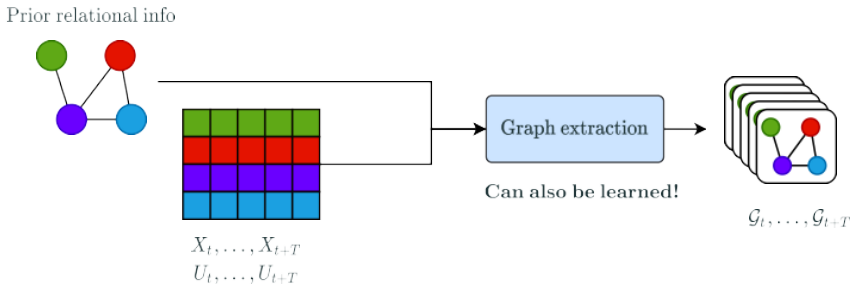
- $\boldsymbol{A}_t \in \mathrm{R}^{N_t \times N_t}$ is a weighted adjacency matrix, with N_t being the number of nodes;
- $\boldsymbol{X}_t \in \mathrm{R}^{N_t \times d_x}$ is the node-attribute matrix;
  - $\boldsymbol{x}_t^i$ (the i-th row of $\boldsymbol{X}_t$) is the d_x-dimensional attribute vector associated with the i-th node;
- $\boldsymbol{U}_t \in \mathrm{R}^{N_t \times d_u}$ are exogenous variables (e.g., weather forecasts, datetime information);
- $\boldsymbol{E}_t \in \mathrm{R}^{E_t \times d_e}$ is an edge-attribute matrix, with E_t being the number of edges;

We focus on settings where the number of nodes is constant over time, i.e., $N_t = N$.

[3] A. Cini *et al.*, "Taming Local Effects in Graph-based Spatiotemporal Forecasting", 2023.

Spatiotemporal graph signal

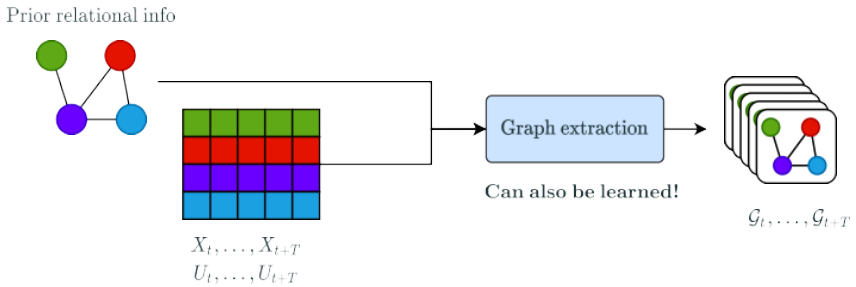
We can then model any multivariate time series $\boldsymbol{X}_{t:t+T} = \{\boldsymbol{X}_t, \ldots, \boldsymbol{X}_{t+T}\}$, together with covariates and additional relational information as a sequence of attributed graphs $\mathrm{G}_{t:t+T} = \{\mathrm{G}_t, \ldots, \mathrm{G}_{t+T}\}$



Note that relational information ($\boldsymbol{A}_t$) **might not be known** a priori and we might be required
to estimate the underlying graph directly from the data, taking **uncertainty** into account.

Spatiotemporal graph signal

We can then model any multivariate time series $\boldsymbol{X}_{t:t+T} = \{\boldsymbol{X}_t, \ldots, \boldsymbol{X}_{t+T}\}$, together with covariates and additional relational information as a sequence of attributed graphs $\mathrm{G}_{t:t+T} = \{\mathrm{G}_t, \ldots, \mathrm{G}_{t+T}\}$



Note that relational information ($\boldsymbol{A}_t$) **might not be known** a priori and we might be required to estimate the underlying graph directly from available data

Spatiotemporal Graph Neural Networks

Learning from Time and Space

Consider families of parametric models f_θ for node-level forecasting:

$$\hat{\boldsymbol{x}}^i_{t:t+H} = f_\theta(\mathcal{G}_{t-W:t}).$$

More precisely, we focus on those families where f_θ is a neural network. We now know how to use neural networks for processin

- the **time** dimension, with RNNs, TCNs, and Transformers;
- the **space** dimension, with CNNs and **GNNs**.

What about processing time and space altogether?

Learning from Time and Space

We consider families of parametric models f_θ for node-level forecasting:

$$\hat{\boldsymbol{x}}^i_{t:t+H} = f_\theta(\mathcal{G}_{t-W:t}).$$

More precisely, we focus on those families where f_θ is a neural network.

We now know how to use neural networks for processing:

- the **time** dimension, with RNNs, TCNs, and Transformers;
- the **space** dimension, with CNNs and **GNNs**.

What about processing time and space altogether?

Learning from Time and Space

We consider families of parametric models f_θ for node-level forecasting:

$$\hat{\boldsymbol{x}}^i_{t:t+H} = f_\theta(\mathcal{G}_{t-W:t}).$$

i

More precisely, we focus on those families where f_θ is a neural network.

We now know how to use neural networks for processing:

- the **time** dimension, with RNNs, TCNs, and Transformers;
- the **space** dimension, with CNNs and **GNNs**.

What about processing time and space altogether?

Learning from Time and Space

We consider families of parametric models f_θ for node-level forecasting:

$$\hat{\boldsymbol{x}}^i_{t:t+H} = f_\theta(\mathcal{G}_{t-W:t}).$$

i

More precisely, we focus on those families where f_θ is a neural network.

We now know how to use neural networks for processing:

- the **time** dimension, with RNNs, TCNs, and Transformers;
- the **space** dimension, with CNNs and **GNNs**.

What about processing time and space altogether?

Learning from Time and Space

We consider families of parametric models f_θ for node-level forecasting:

$$\hat{\boldsymbol{x}}^i_{t:t+H} = f_\theta(\mathcal{G}_{t-W:t}).$$

More precisely, we focus on those families where f_θ is a neural network.

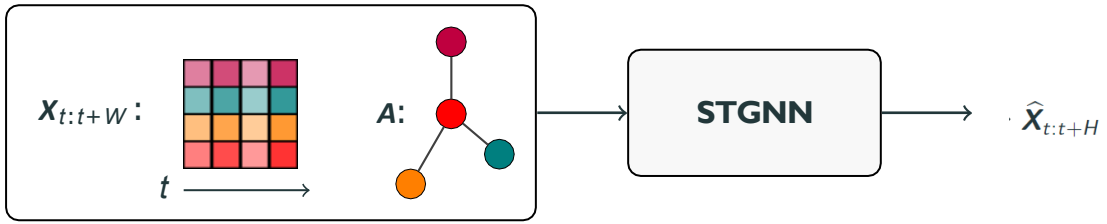
We now know how to use neural networks for processing:

- the **time** dimension, with RNNs, TCNs, and Transformers;
- the **space** dimension, with CNNs and **GNNs**.

What about processing **time and space altogether**?

Spatiotemporal Graph Neural Networks

We call Spatiotemporal Graph Neural Network (STGNN) a neural network exploiting both temporal and spatial relations of the input spatiotemporal graph signals.



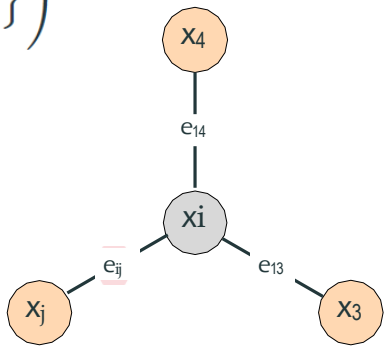
We consider families of models that exploit message-passing to process the spatial dimension

Spatiotemporal message passing

A general scheme for spatiotemporal message-passing networks:

$$\boldsymbol{z}^i_{t-W:t} = \gamma\left(\boldsymbol{x}^i_{t-W:t}, \underset{j\in\mathcal{N}(i)}{\texttt{Aggr}}\left\{\phi\left(\boldsymbol{x}^i_{t-W:t}, \boldsymbol{x}^j_{t-W:t}, \boldsymbol{e}^{ij}_{t-W:t}\right)\right\}\right)$$

- ϕ is the **message function.**
- $\texttt{Aggr}$ is a permutation invariant **aggregation function.**
- γ is the **update function.**

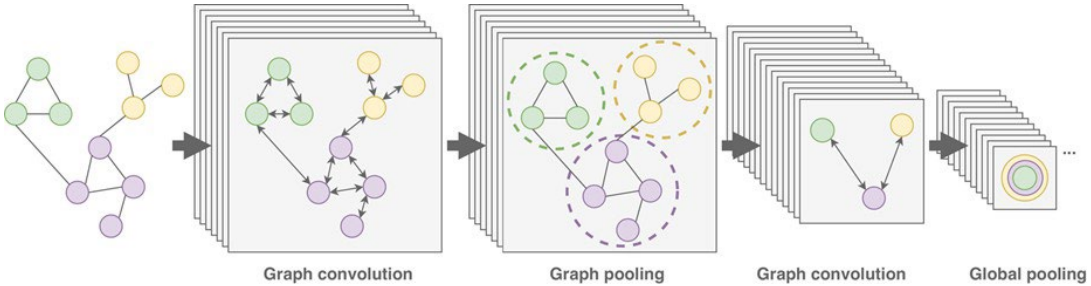


[3] A. Cini *et al.*, "Taming Local Effects in Graph-based Spatiotemporal Forecasting", 2023.

Spatiotemporal message passing

A general scheme for spatiotemporal message-passing networks:

$$\boldsymbol{z}^i_{t-W:t} = \gamma\left(\boldsymbol{x}^i_{t-W:t}, \underset{j\in\mathcal{N}(i)}{\texttt{Aggr}}\left\{\phi\left(\boldsymbol{x}^i_{t-W:t}, \boldsymbol{x}^j_{t-W:t}, \boldsymbol{e}^{ij}_{t-W:t}\right)\right\}\right)$$



[3] A. Cini *et al.*, "Taming Local Effects in Graph-based Spatiotemporal Forecasting", 2023.

Processing approaches

There exist different design paradigms on how to integrate temporal and spatial processing in a single architecture:

- **Time-then-Space**
  Embed each time series in a vector, which is then propagated over the graph.
- **Time-and-Space**
  Temporal and spatial processing are interleaved inside the same architecture's module.
- **Product graph**
  The sequence of graphs is transformed into a single graph, then processed with a GNN.

There exist different design paradigms on how to integrate temporal and spatial processing in a single architecture:

- **Time-then-Space**
  Embed each time series in a vector, which is then propagated over the graph.
- **Time-and-Space**
  Temporal and spatial processing are interleaved inside the same architecture's module.
- **Product graph**
  The sequence of graphs is transformed into a single graph, then processed with a GNN.

Processing approaches

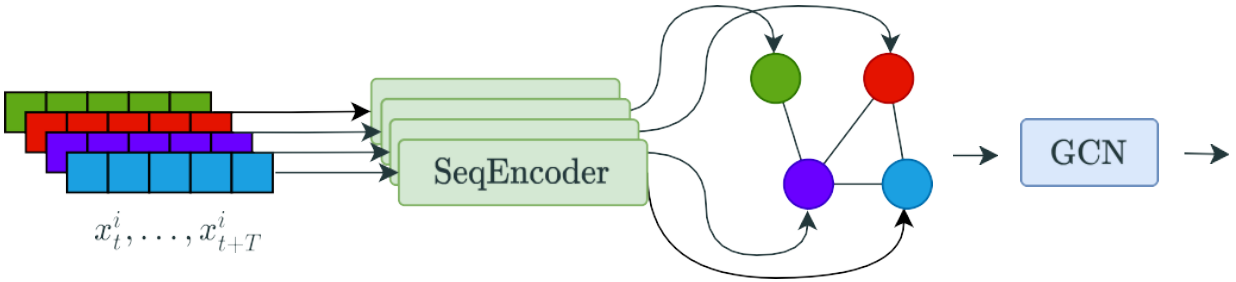
There exist different design paradigms on how to integrate temporal and spatial processing in a single architecture:

- **Time-then-Space**
  Embed each time series in a vector, which is then propagated over the graph.
- **Time-and-Space**
  Temporal and spatial processing are interleaved inside the same architecture's module.
- **Product graph**
  The sequence of graphs is transformed into a single graph, then processed with a GNN.

Time-then-Space

A straightforward approach to process spatiotemporal graphs is simply to:

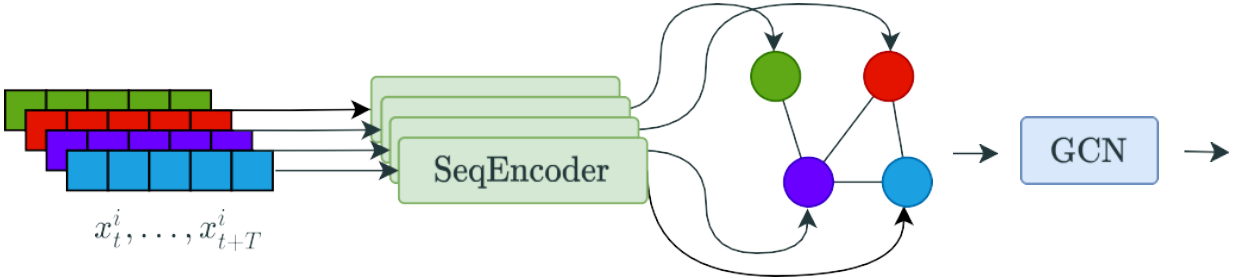
1. **Embed** each node-level time series in a vector.
2. Use (a stack of) any of the **graph convolutional layers** we have seen so far.



Time-then-Space

A straightforward approach to process spatiotemporal graphs is simply to:

1. **Embed** each node-level time series in a vector.
2. Use (a stack of) any of the **graph convolutional layers**



Time-and-Space

The idea is to use graph convolutional layers to implement (part of) neural operators for sequential data...

...or, conversely, implement message-passing networks by using temporal operators.

Different strategies exhist

- Interleaved spatial and temporal filters.
- Graph Convolutional Recurrent Neural Networks.
- Spatiotemporal Message Passing.

Time-and-Space

The idea is to use graph convolutional layers to implement (part of) neural operators for sequential data...

...or, conversely, implement message-passing networks by using temporal operators.

Different strategies, exhist

- Interleaved spatial and temporal filters.
- Graph Convolutional Recurrent Neural Networks.
- Spatiotemporal Message Passing.

Time-and-Space

The idea is to use graph convolutional layers to implement (part of) neural operators for sequential data...

...or, conversely, implement message-passing networks by using temporal operators.

Different strategies exist

- Interleaved spatial and temporal filters.
- Graph Convolutional Recurrent Neural Networks.
- Spatiotemporal Message Passing.

Spatiotemporal Graph Convolution Networks

We can build deep spatiotemporal convolutional neural networks by **alternating spatial and temporal convolutional filters**.

The main idea:

- Compute intermediate representations by using a node-wise temporal convolutional layer:

$$\boldsymbol{x}'^{i}_{t-W:t} = \xi\left(\boldsymbol{\Theta}_1 \star_\tau \boldsymbol{x}^{i}_{t-w:t}\right)$$

  where $\star_\tau$ indicates the temporal convolution operator and ξ is an activation function.

- Then, compute the updated representation by using a time-wise spatial convolution:

$$\boldsymbol{Z}_t = \sigma\left(\boldsymbol{A}_t \boldsymbol{X}'_t \boldsymbol{\Theta}_2\right)$$

Spatiotemporal Graph Convolution Networks

We build deep spatiotemporal convolutional neural networks by **alternating spatial and temporal convolutional filters**.

The main idea:

- Compute intermediate representations by using a node-wise temporal convolutional layer:

$$\boldsymbol{x}'^{i}_{t-W:t} = \xi\left(\boldsymbol{\Theta}_1 \star_{\mathcal{T}} \boldsymbol{x}^{i}_{t-w:t}\right)$$

where $\star_{\tau}$ indicates the temporal convolution operator and ξ is an activation function.

Spatiotemporal Graph Convolution Networks

We build deep spatiotemporal convolutional neural networks by **alternating spatial and temporal convolutional filters**.

The main idea:

- Compute intermediate representations by using a node-wise temporal convolutional layer:

$$\boldsymbol{x}'^{i}_{t-W:t} = \xi\left(\boldsymbol{\Theta}_1 \star_{\mathcal{T}} \boldsymbol{x}^{i}_{t-w:t}\right)$$

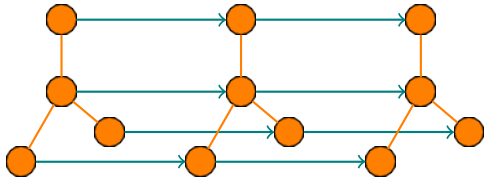
  where $\star_{\mathcal{T}}$ indicates the temporal convolution operator and ξ is an activation function.

- Then, compute the updated representation by using a time-wise spatial convolution:

$$\boldsymbol{Z}_t = \sigma\left(\tilde{\boldsymbol{A}}_t \boldsymbol{X}'_t \boldsymbol{\Theta}_2\right)$$

Product Graph

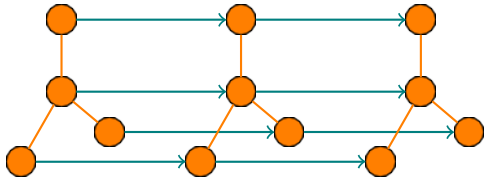
- **Cartesian product graph**
  Spatial graphs are kept and each node is connected to itself in the previous time instant.



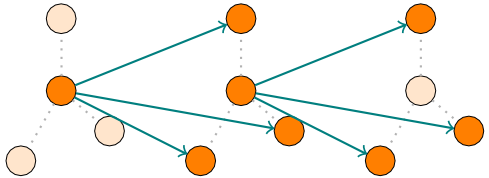
- **Kronecker product graph**
  Each node is connected **only** to its neighbors in the previous time instant.

Product Graph

- **Cartesian product graph**
  Spatial graphs are kept and each node is connected to itself in the previous time instant.



- **Kronecker product graph**
  Each node is connected **only** to its neighbors in the previous time instant.



Product Graph

Of course, spatial and temporal edges can (and should) be treated differently in the processing.

A possibility is to represent the product graph as a **heterogeneous graph**, assigning a different class to spatial and temporal edges.

[10] S. Yan *et al.*, "Spatial temporal graph convolutional networks for skeleton-based action recognition", 2018.

Product Graph

Of course, spatial and temporal edges can (and should) be treated differently in the processing.

A possibility is to represent the product graph as a **heterogeneous graph**, assigning a different class to spatial and temporal edges.

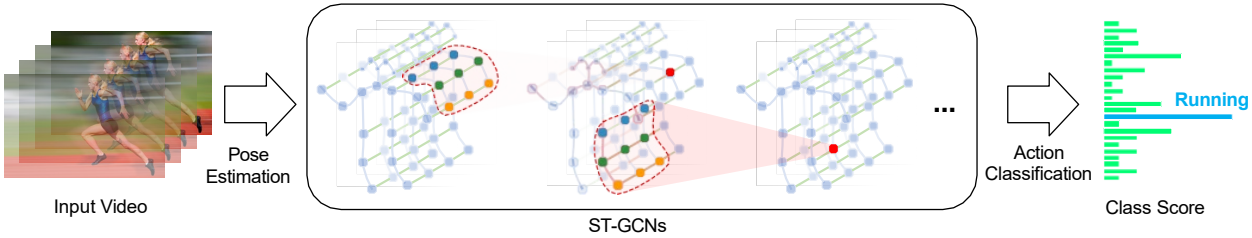
[10] S. Yan *et al.*, "Spatial temporal graph convolutional networks for skeleton-based action recognition", 2018.

Product Graph

Of course, spatial and temporal edges can (and should) be treated differently in the processing.

A possibility is to represent the product graph as a **heterogeneous graph**, assigning a different class to spatial and temporal edges.

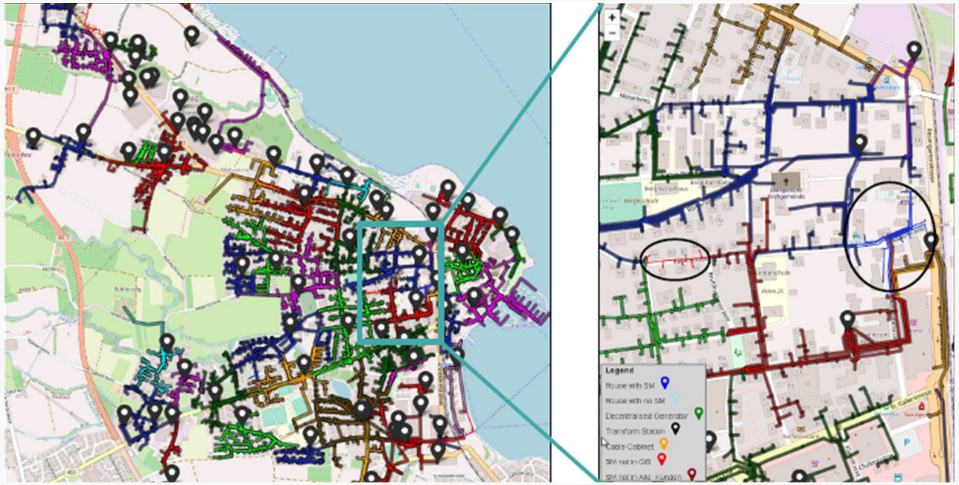
Some approaches in the literature process spatiotemporal data in this fashion, e.g. [10]:



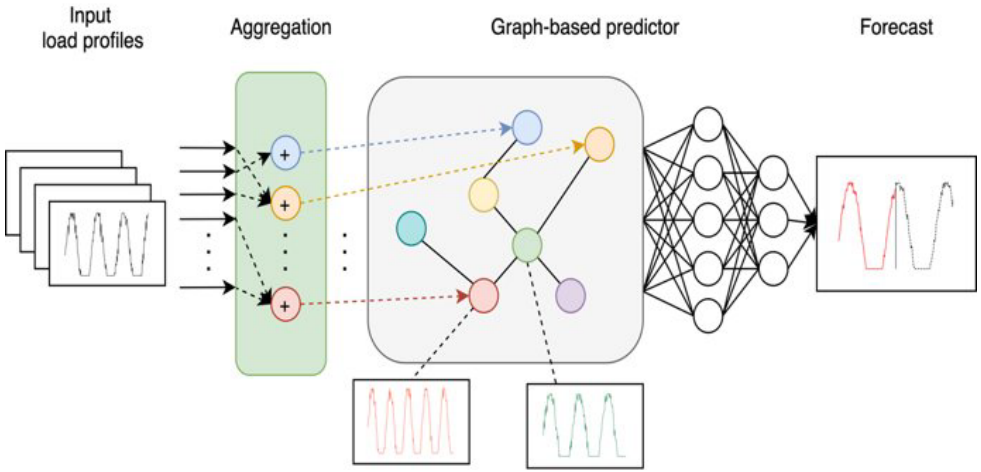
[10] S. Yan *et al.*, "Spatial temporal graph convolutional networks for skeleton-based action recognition", 2018.

SONDER – SFOE Project

- Day-ahead forecasts of peak consumption and quarter-hour consumptions for control of Battery Energy Storage System (BESS) and Peak Shaving
- Use case: smart grid of Arbon. 10k+ users



SONDER – Load forecast



- Learn to **predict the load at each node** with a single Graph Neural Network (GNN) model.
- Also **learn the graph edge weights** among nodes.

Model Name	MAE (MW)	SUM_MAE (MW)	SUM_MAPE (%)
Gated Graph Network	**0.030**	**0.648**	**6.33 %**
FCRNN-large	0.038	0.761	7.40 %
GRU-RNN (univariate)	-	0.678	6.70 %

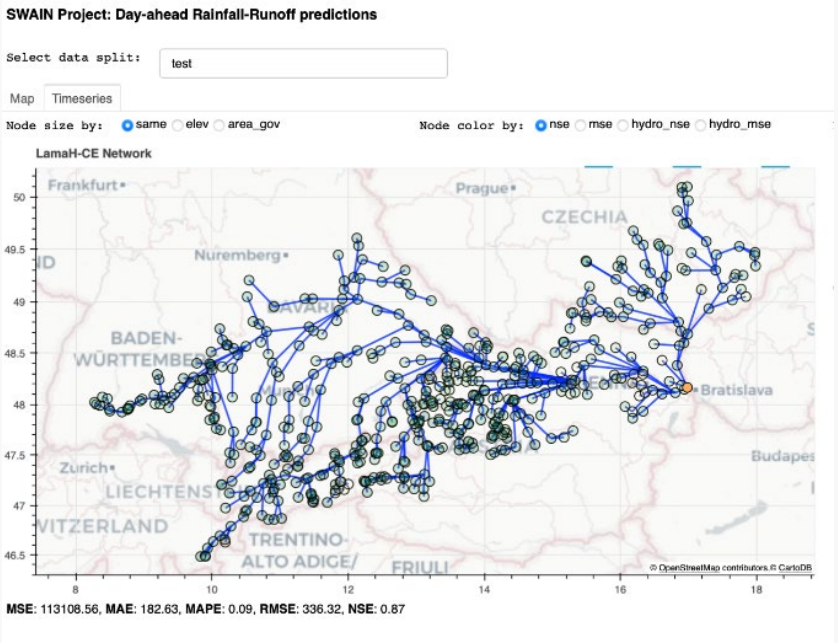
Results on 50 aggregates of meters

Model Name	MAE (kW)	NRMSE (%)	R2
Gated Graph Network	**23.572**	**2.50%**	**0.872**
FCRNN-large	29.52	3.02 %	0.813
FCRNN-small	31.568	3.17%	0.794

Results on single sample meters

Swain – ERA-Net/SNSF Project

- Predictive graph model for rainfall runoff and pollutants anomalies detection
- Principal use case: Danube basin (800+ measurement points)
- Used meteo data and GNNs to forecast water flows all over the basin.

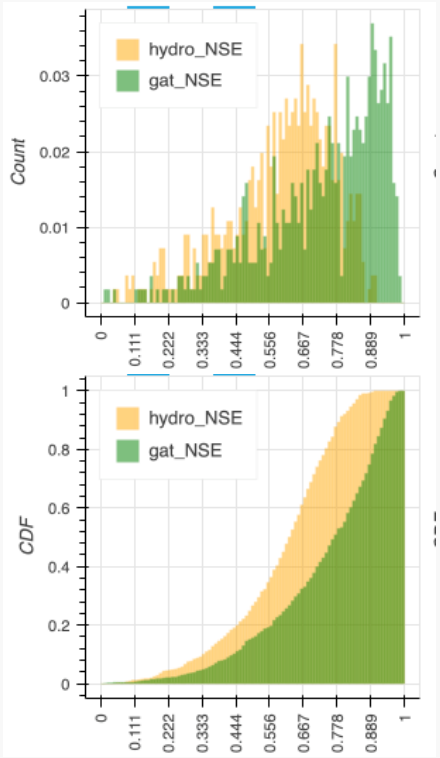


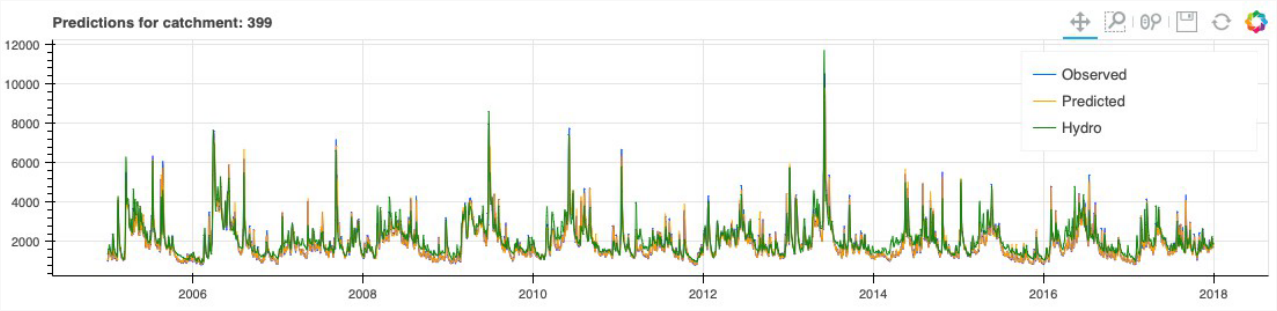
Waterflow Prediction

Graph models predict the waterflow simultaneously at each location outperforming traditional hydrological models with an improvement of 14% .

Graph models consider causal relationships among locations.

$$NSE = 1 - \frac{\sum_{t=1}^{T} (Q_o^t - Q_m^t)^2}{\sum_{t=1}^{T} \left(Q_o^t - \overline{Q}_o\right)^2}$$





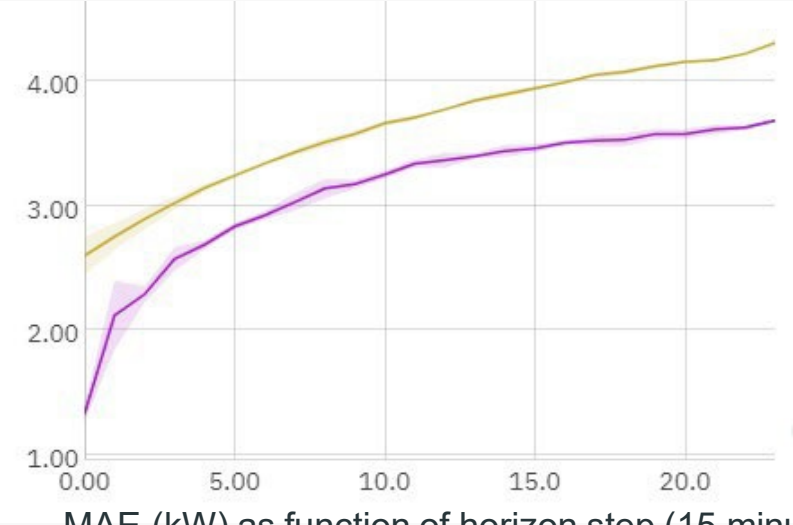
GraPV – Innosuisse Project

- Day-ahead and intraday solar farms production forecasting
- Outcome: GraphSight USI Spinoff
- **SODA** is a research project on PV power forecasting, carried out by **CSEM** and **BKW** focusing on **Intraday forecasting**: 24 timesteps ahead (15 minutes to 6h)
- SODA data are not publicly available, but performance on the public NREL dataset of simulated ~5000 PV production in California are given in paper
  Simeunović, J., et al. "Interpretable temporal-spatial graph attention network for multi-site PV power forecasting." Applied Energy 327 (2022).

Our solution w.r.t. the SODA project

		SODA (Simeunović et. al.)		Our	
Dataset	**Improvment**	**MAE (kW)**	**MRE (%)**	**MAE (kW)**	**MRE (%)**
California	**+ 10.7%**	3.28 ± 0.04	13.0 ± 0.1	**2.93 ± 0.03**	**11.6 ± 0.1**

Mean performance over all farms (± standard deviation)



MAE (kW) as function of horizon step (15 minutes)
Purple = Ours vs Yellow = SODA (Simeunović et. al.)

Concluding remarks

Even if the crystal ball looks foggy
useful forecasts can be provided given an appropriate predictive model family and expressive-enough data;

Inductive bias is a strong plus whenever available.

Graph Machine Learning Group

Thanks to my great team


Cesare **Alippi**


Slobodan **Lukovic**


Daniele **Zambon**


Andrea **Cini**


Ivan **Marisca**


Luca **Butera**


Alessandro **Manenti**


Tommaso **Marzi**


Stefano **Imoscopi**


Federico **Bombardieri**

gmlg.ch