Graph Deep Learning for Time Series and Spatiotemporal Data

Daniele **Zambon**

Graph Machine Learning Group (gmlg.ch)
The Swiss AI Lab IDSIA
Università della Svizzera italiana

Università di Padova, Italy · Spring 2026

Introduction

Traffic monitoring

Weather monitoring

Smart cities

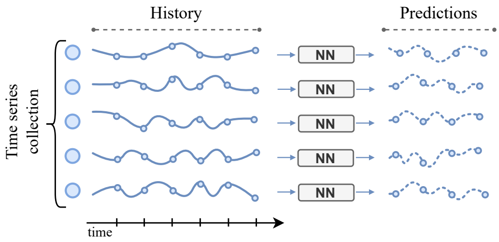
Physics

Stock markets

Deep learning for time series

Modern deep learning methods for time series prediction tasks (e.g., forecasting) rely on a single neural network trained on a collection of related time series.

- ☺ Each time series is processed **independently**.
- ☺ Parameters are **shared**.
- ☺ Effective and **sample efficient**.
- ☹ **Dependencies are neglected**.

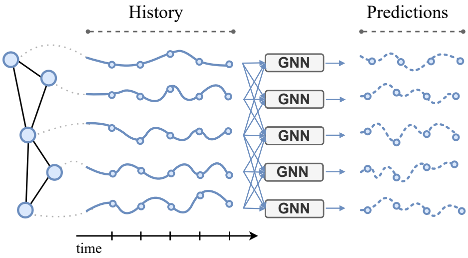


[1] Salinas *et al.*, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks", IJF 2020.
[2] Benidis *et al.*, "Deep Learning for Time Series Forecasting: Tutorial and Literature Survey", ACM CS 2022.

Graph deep learning for time series and spatiotemporal data

We will show graph deep learning (GDL) provides appropriate operators to go beyond these limitations.

- ☺ **Dependencies** are **embedded into the processing** as inductive biases.
- ☺ Operate on **sets** of **correlated time series**.
- ☺ Parameters are **shared**.

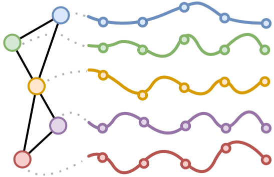


☹ There are inherent **challenges** in applying this processing to data from the real world.

[3] Cini, Marisca, Zambon, and Alippi, "Graph Deep Learning for Time Series Forecasting", ACM CSUR 2025.

What this tutorial is about

This tutorial presents advances coming from the combination of



1. deep learning for time series and
2. deep learning on graphs.

The objective of the tutorial is to provide:

1. a **comprehensive framework** for graph-based time series and spatiotemporal processing models;
2. methods to address challenges and potential pitfalls;
3. tools and guidelines for real-world applications and developing new methods.

This presentation is complemented by a tutorial paper [3].

[3] Cini, Marisca, Zambon, and Alippi, "Graph Deep Learning for Time Series Forecasting", ACM CSUR 2025.

Course outline

Part 1

1) Graph-based Processing of Time Series
2) Spatiotemporal Graph Neural Networks
3) Global and Local models

Hands-on Sessions

Tutorial paper [3]

Part 2

4) Latent Graph Learning
5) Scalability & Missing-Data Handling
6) Model Quality Assessment

Future Directions

Material developed in collaboration with:

Andrea **Cini**

Ivan **Marisca**

[3] Cini, Marisca, Zambon, and Alippi, "Graph Deep Learning for Time Series Forecasting", ACM CSUR 2025.

Module 1

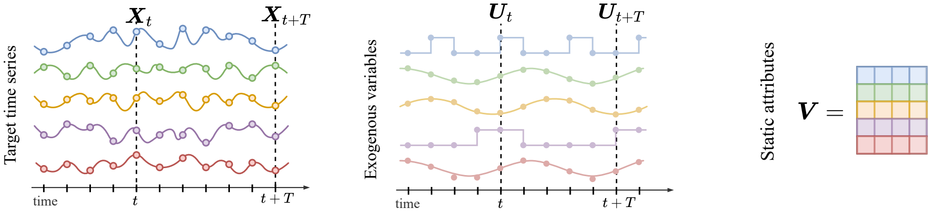
Graph-based Processing of Correlated Time Series

Correlated time series & spatiotemporal data

Collections of time series

We consider a set $\mathcal{D}$ of N correlated time series. Each i-th time series can be associated with:

- **observations** $\boldsymbol{x}_t^i \in \mathbb{R}^{d_x}$ at each time step t;
- **exogenous variables** $\boldsymbol{u}_t^i \in \mathbb{R}^{d_u}$ at each time step t;
- a vector of **static (time-independent) attributes** $\boldsymbol{v}^i \in \mathbb{R}^{d_v}$.



Capital letters denote the stacked N time series, i.e., $\boldsymbol{X}_t \in \mathbb{R}^{N \times d_x}$, $\boldsymbol{U}_t \in \mathbb{R}^{N \times d_u}$.

$\rightarrow$ We call spatial the dimension spanning the collection.

[3] Cini, Marisca, Zambon, and Alippi, “Graph Deep Learning for Time Series Forecasting”, ACM CSUR 2025.

Correlated time series

We consider a time-invariant stochastic process generating each time series as

$$\boldsymbol{x}_t^i \sim p^i\left(\boldsymbol{x}_t^i \middle| \boldsymbol{X}_{<t}, \boldsymbol{U}_{\leq t}, \boldsymbol{V}\right) \quad \text{for all } i = 1 \dots N, t = 0, \dots, T-1$$

and assume the existence of a causality à la Granger among time series.

Furthermore time series

- are assumed

  a) homogenous, b) synchronous, c) regularly sampled.
- can be generated by different processes.

Notation:

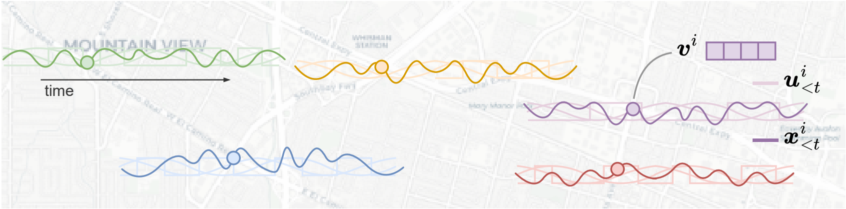
$$\mathcal{X}_t = \langle \boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{V} \rangle$$

$$\mathcal{X}_{<t} = [\mathcal{X}_0, \cdots, \mathcal{X}_{t-2}, \mathcal{X}_{t-1}]$$

❗ Assumptions a),b),c) can be relaxed as we will discuss in the 2nd part.

Example: Traffic monitoring system

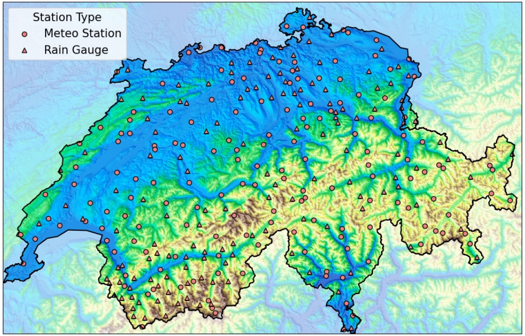
Consider a sensor network monitoring the speed of vehicles at crossroads.



- $\boldsymbol{X}_{<t}$ collects past traffic speed measurements.
- $\boldsymbol{U}_t$ stores identifiers for time-of-the-day and day-of-the-week.
- $\boldsymbol{V}$ collects static sensor's features, e.g., type or number of lanes of the monitored road.

$\rightarrow$ Strong dependencies among time series that reflect the road network.

Example: Ground-station weather monitoring system



- $\boldsymbol{X}_{<t}$ collects ground-level measurements of several meteorological variable.
- $\boldsymbol{U}_t$ stores identifiers for time-of-the-day and day-of-the-year.
- $\boldsymbol{V}$ collects static sensor's features, such as elevation, slope, and aspect of the station location.

$\rightarrow$ Time series correlations reflect the spatial proximity and topographic characteristics network.

[4] Zambon, Cattaneo, Marisca, Bhend, Nerini, and Alippi, *PeakWeather*, HuggingFace 2025.

Tackling prediction tasks

Time series forecasting

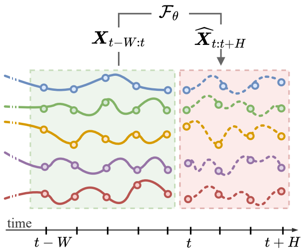
Multi-step time-series forecasting

Given a window of $W \geq 1$ past values

$$\mathcal{X}_{t-W:t} = [\mathcal{X}_{t-W}, \dots, \mathcal{X}_{t-1}],$$

predict $H \geq 1$ future observations

$$\boldsymbol{X}_{t+h} \qquad h = 1, \dots, H.$$



In particular, we are interested in learning a parametric model $\mathcal{F}(\cdot\,; \boldsymbol{\theta})$ s.t.

$$\mathcal{F}\left(\mathcal{X}_{t-W:t}, \boldsymbol{U}_{t:t+H}; \boldsymbol{\theta}\right) = \widehat{\boldsymbol{X}}_{t:t+H} \approx E_p\left[\boldsymbol{X}_{t:t+H}\right].$$

Remarks:

- Probabilistic predictors can be considered as well, but we focus on point forecasts.
- Similar ideas applies to other tasks, such as anomaly detection, classification, and imputation.

[5] Jin *et al.*, "A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection", IEEE

Training objective

For point predictors, parameters $\boldsymbol{\theta}$ can be learned by minimizing a cost function $\ell(\,\cdot\,,\,\cdot\,)$ (e.g., MSE) on a training set

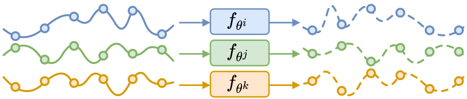
$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{NT} \sum_{t=1}^{T} \sum_{i=1}^{N} \ell\left(\widehat{\boldsymbol{x}}^i_{t:t+H}, \boldsymbol{x}^i_{t:t+H}\right)$$
$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{NT} \sum_{t=1}^{T} \sum_{i=1}^{N} \left\| \boldsymbol{x}^i_{t:t+H} - \widehat{\boldsymbol{x}}^i_{t:t+H} \right\|_2^2 .$$

❗ Choosing a different cost function allows for predicting other values.

→ **Example:** minimizing the MAE results in forecasts of the median.

Global and local predictors

Local models



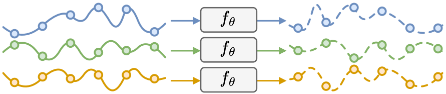
$$\hat{\boldsymbol{x}}^i_{t+h} = f\left(\boldsymbol{x}^i_{t-W:t}, \dots ; \boldsymbol{\theta}^i\right)$$

Example: Box-Jenkins method

☺ Tailored to each time series.

☹ Inefficient.

Global models



$$\hat{\boldsymbol{x}}^i_{t+h} = f\left(\boldsymbol{x}^i_{t-W:t}, \dots ; \boldsymbol{\theta}\right)$$

Example: DeepAR [1]

☺ Sample efficient.

☺ Allows for more complex models.

☹ Both approaches neglect dependencies among time series.

[1] Salinas *et al.*, “DeepAR: Probabilistic forecasting with autoregressive recurrent networks”, IJF 2020.
[6] Montero-Manso *et al.*, “Principles and algorithms for forecasting groups of time series: Locality and globality”, IJF 2021.

Accounting for spatial dependencies

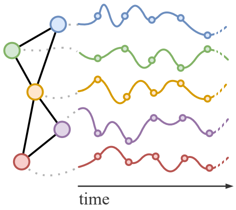
- One option is to consider the input as single multivariate time series
  - → Resulting predictors are **local**: $\widehat{\boldsymbol{X}}_{t+h} = f\left(\boldsymbol{X}_{t-W:t}, \ldots; \boldsymbol{\theta}\right).$
  - ☹ High sample complexity and poor scalability.
- Models operating on sets of time series would allow to keep parameters shared.
  - → Resulting predictors are **global**: $\widehat{\boldsymbol{X}}^{\mathcal{S}}_{t+h} = \mathcal{F}\left(\boldsymbol{X}^{\mathcal{S}}_{t-W:t}, \ldots; \boldsymbol{\theta}\right), \qquad \forall \mathcal{S} \subseteq \mathcal{D}$
  - ☺ Can be implemented by attention-based models (e.g, Transformers).
  - ☹ Does not exploit structural priors, high computational and sample complexity.
- Other methods (e.g., [7]) rely on dimensionality reduction to extract shared latent factors.
  - ☺ Might work well if data are low-rank.
  - ☹ Local and relational information are lost and can still suffer from, scalability issues.

[2] Benidis *et al.*, “Deep Learning for Time Series Forecasting: Tutorial and Literature Survey”, ACM CS 2022.
[7] Sen *et al.*, “Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting”, NeurIPS 2019.

Graph-based representation

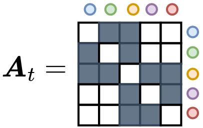
Relational information

💡 Exploit functional dependencies as an inductive bias to improve the forecasts.



We can model pairwise relationships existing at time step t with adjacency matrix $\boldsymbol{A}_t \in \{0,1\}^{N \times N}$.

- $\boldsymbol{A}_t$ can be **asymmetric** and **dynamic** (can vary with t).

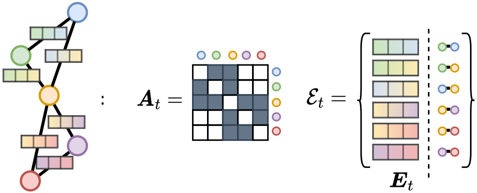


Relational information with attributes

Optional edge attributes $\boldsymbol{e}_t^{ij} \in \mathbb{R}^{d_e}$ can be associated to each non-zero entry of $\boldsymbol{A}_t$.

The set of attributed edges is denoted by

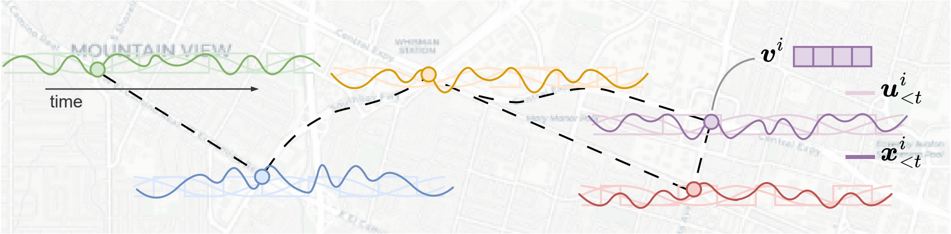
$$\mathcal{E}_t \doteq \{\langle (i,j), \boldsymbol{e}_t^{ij} \rangle \mid \forall i,j : \boldsymbol{A}_t[i,j] \neq 0\}.$$



$\rightarrow$ Edge attributes can be both **categorical** or **numerical**.

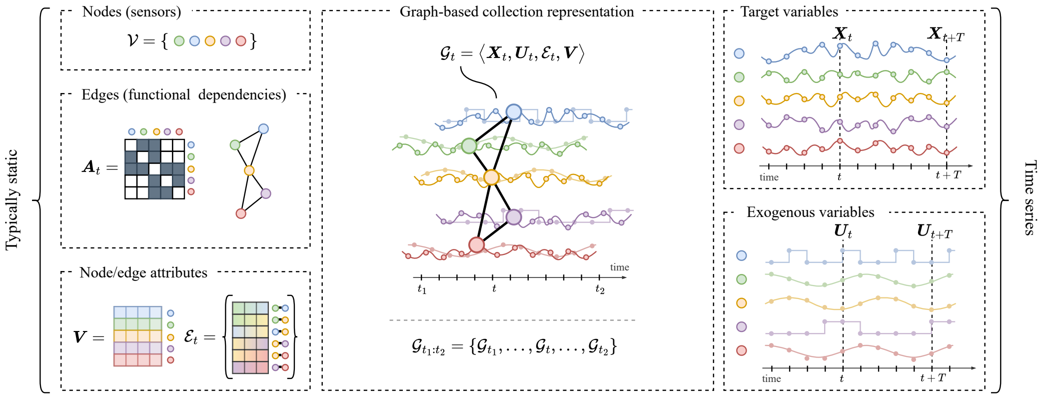
Example: Traffic monitoring system

Consider again the sensor network of the previous example.



- Edges in $\mathcal{E}$ can be obtained by considering the road network.
  - → Road closures and traffic diversions can be accounted for with a dynamic topology $\mathcal{E}_t$.

Graph-based representations for correlated time series



$\mathcal{G}_t \doteq \langle \boldsymbol{X}_t, \boldsymbol{U}_t, \mathcal{E}_t, \boldsymbol{V} \rangle$ contains the available information w.r.t. time step t.

[3] Cini, Marisca, Zambon, and Alippi, “Graph Deep Learning for Time Series Forecasting”, ACM CSUR 2025.

Graph-based processing

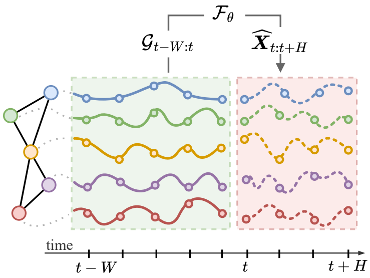
Relational inductive biases for time series forecasting

Forecasts can be conditioned on the available relational information $\mathcal{E}_{t-W:t}$

$$\widehat{\boldsymbol{X}}^{\mathcal{S}}_{t:T+H} = \mathcal{F}\left(\mathcal{G}^{\mathcal{S}}_{t-W:t}, \boldsymbol{U}^{\mathcal{S}}_{t:t+H}; \boldsymbol{\theta}\right) \qquad \forall \mathcal{S} \in \mathcal{D}$$

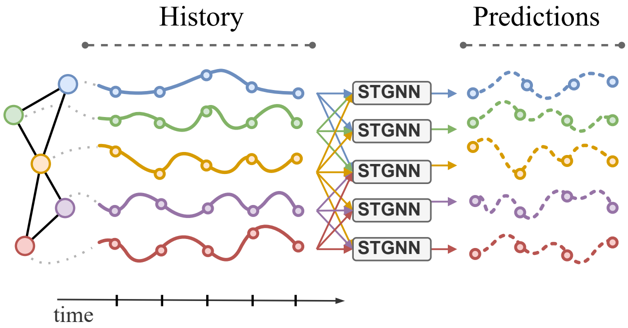
The conditioning can act as a regularization to localize predictions w.r.t. each node.

- ☺ Relational priors prune spurious correlations.
- ☺ More scalable than standard multivariate models.
- ☺ Can forecast any subset of correlated time series.



Spatiotemporal graph neural networks

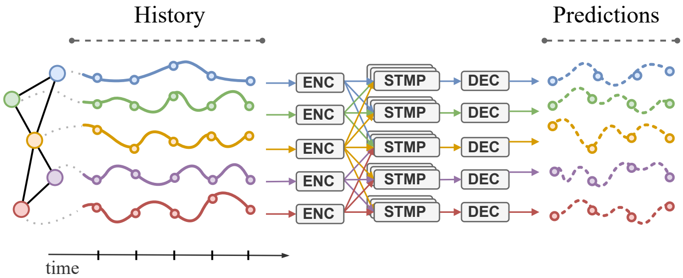
We call spatiotemporal graph neural networks (STGNNs) a neural network exploiting both temporal and spatial relations of the input spatiotemporal time series.



We focus on models based on message passing (MP).

A general recipe for building STGNNs

We consider STGNNs consisting of three main components



- ENC(·) is the **encoding** layer, e.g., implemented by an MLP.
- STMP(·) is a stack of **spatiotemporal message-passing (STMP)** layers.
- DEC(·) is the **readout** layer, e.g., implemented by an MLP.

[3] Cini, Marisca, Zambon, and Alippi, "Graph Deep Learning for Time Series Forecasting", ACM CSUR 2025.

A closer look

Representations are updated as follows.

$$\boldsymbol{h}_{t-1}^{i,0} = \textsc{Encoder}\left(\boldsymbol{x}_{t-1}^{i}, \boldsymbol{u}_{t-1}^{i}, \boldsymbol{v}^{i}\right), \tag{1}$$

$$\boldsymbol{H}_{t-1}^{l+1} = \text{STMP}^{l}\left(\boldsymbol{H}_{\leq t-1}^{l}, \mathcal{E}_{\leq t-1}\right), \quad l = 0, \dots, L-1 \tag{2}$$

$$\hat{\boldsymbol{x}}_{t:t+H}^{i} = \textsc{Decoder}\left(\boldsymbol{h}_{t-1}^{i,L}, \boldsymbol{u}_{t:t+H}^{i}\right). \tag{3}$$

- $\textsc{Enc}(\,\cdot\,)$ process each observation independently.
- $\text{STMP}(\,\cdot\,)$ is where propagation through time and space happens.
- $\textsc{Dec}(\,\cdot\,)$ maps each representation to predictions.

[3] Cini, Marisca, Zambon, and Alippi, “Graph Deep Learning for Time Series Forecasting”, ACM CSUR 2025.

Spatiotemporal message-passing (STMP)

STMP blocks can be defined as:

$$\boldsymbol{h}_t^{i,l+1} = \textsc{Up}^l\left(\boldsymbol{h}_{\leq t}^{i,l}, \underset{j\in\mathcal{N}_t(i)}{\textsc{Aggr}}\left\{\textsc{Msg}^l\left(\boldsymbol{h}_{\leq t}^{i,l}, \boldsymbol{h}_{\leq t}^{j,l}, \boldsymbol{e}_{\leq t}^{ji}\right)\right\}\right)$$

Each block processes **sequences** while accounting for **relational dependencies**.

As in standard MP operators:

- $\textsc{Msg}^l(\,\cdot\,)$ is a **message function**, e.g., implemented by *temporal convolutional layers*.
- $\textsc{Aggr}\{\,\cdot\,\}$ is a permutation invariant **aggregation function**.
- $\textsc{Up}^l(\,\cdot\,)$ is an **update function**, e.g., implemented by an RNN.

❗ Blocks can be implemented by composing MP and sequence modeling operators.
$\rightarrow$ Many possible designs exist.

[8] Gilmer *et al.*, “Neural message passing for quantum chemistry”, ICML 2017.
[3] Cini, Marisca, Zambon, and Alippi, “Graph Deep Learning for Time Series Forecasting”, ACM CSUR 2025.

Design paradigms for STGNNs

Depending on the implementation of the STMP blocks, we categorize STGNNs into:

- **Time-and-Space (T&S)**
  Temporal and spatial processing cannot be factorized in two separate steps.
- **Time-then-Space (TTS)**
  Each time series is embedded in a vector and then representations are propagated on the graph.
- **Space-then-Time (STT)**
  Spatial propagation is performed before processing the resulting time series.

References i

[1] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, **“DeepAR: Probabilistic forecasting with autoregressive recurrent networks,”** *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[2] K. Benidis, S. S. Rangapuram, V. Flunkert, *et al.*, **“Deep learning for time series forecasting: Tutorial and literature survey,”** *ACM Comput. Surv.*, vol. 55, no. 6, Dec. 2022, ISSN: 0360-0300. DOI: 10.1145/3533382. [Online]. Available: https://doi.org/10.1145/3533382.

[3] A. Cini, I. Marisca, D. Zambon, and C. Alippi, **“Graph Deep Learning for Time Series Forecasting,”** *ACM Comput. Surv.*, 2025, ISSN: 0360-0300. DOI: 10.1145/3742784. [Online]. Available: https://doi.org/10.1145/3742784.

[4] D. Zambon, M. Cattaneo, I. Marisca, J. Bhend, D. Nerini, and C. Alippi, ***PeakWeather*,** https://huggingface.co/datasets/MeteoSwiss/PeakWeather, 2025.

[5] M. Jin, H. Y. Koh, Q. Wen, *et al.*, **“A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection,”** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 466–10 485, 2024. DOI: 10.1109/TPAMI.2024.3443141.

References ii

[6] P. Montero-Manso and R. J. Hyndman, **"Principles and algorithms for forecasting groups of time series: Locality and globality,"** *International Journal of Forecasting*, vol. 37, no. 4, pp. 1632–1653, 2021.

[7] R. Sen, H.-F. Yu, and I. S. Dhillon, **"Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting,"** *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[8] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, **"Neural message passing for quantum chemistry,"** in *International conference on machine learning*, PMLR, 2017, pp. 1263–1272.

[9] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, **"Geometric deep learning: Grids, groups, graphs, geodesics, and gauges,"** *arXiv preprint arXiv:2104.13478*, 2021.

[10] T. N. Kipf and M. Welling, **"Semi-supervised classification with graph convolutional networks,"** *arXiv preprint arXiv:1609.02907*, 2016.

[11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, **"Graph attention networks,"** in *International Conference on Learning Representations*, 2018.

References iii

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, **"Empirical evaluation of gated recurrent neural networks on sequence modeling,"** *arXiv preprint arXiv:1412.3555*, 2014.

[13] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, **"Structured sequence modeling with graph convolutional recurrent networks,"** in *International Conference on Neural Information Processing*, Springer, 2018, pp. 362–373.

[14] Y. Li, R. Yu, C. Shahabi, and Y. Liu, **"Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,"** in *International Conference on Learning Representations*, 2018.

[15] B. Yu, H. Yin, and Z. Zhu, **"Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,"** in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.

[16] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, **"Graph wavenet for deep spatial-temporal graph modeling,"** in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 1907–1913.

[17] I. Marisca, A. Cini, and C. Alippi, **"Learning to reconstruct missing data from spatiotemporal graphs with sparse observations,"** in *Advances in Neural Information Processing Systems*, 2022.

References iv

[18] Z. Wu, D. Zheng, S. Pan, Q. Gan, G. Long, and G. Karypis, **"Traversenet: Unifying space and time in message passing for traffic forecasting,"** *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[19] M. Sabbaqi and E. Isufi, **"Graph-time convolutional neural networks: Architecture and theoretical analysis,"** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 14 625–14 638, Dec. 2023, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2023.3311912.

[20] J. Gao and B. Ribeiro, **"On the equivalence between temporal and static equivariant graph representations,"** in *International Conference on Machine Learning*, PMLR, 2022, pp. 7052–7076.

[21] A. Cini, I. Marisca, F. M. Bianchi, and C. Alippi, **"Scalable spatiotemporal graph neural networks,"** *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, pp. 7218–7226, Jun. 2023. DOI: 10.1609/aaai.v37i6.25880.

[22] V. G. Satorras, S. S. Rangapuram, and T. Januschowski, **"Multivariate time series forecasting with latent graph inference,"** *arXiv preprint arXiv:2203.03423*, 2022.

[23] A. Cini, I. Marisca, D. Zambon, and C. Alippi, **"Taming local effects in graph-based spatiotemporal forecasting,"** *arXiv preprint arXiv:2302.04071*, 2023.

References v

[24] L. Butera, G. D. Felice, A. Cini, and C. Alippi, **"On the regularization of learnable embeddings for time series forecasting,"** *Transactions on Machine Learning Research*, 2025, ISSN: 2835-8856. [Online]. Available: `https://openreview.net/forum?id=F5ALCh3GWG`.