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

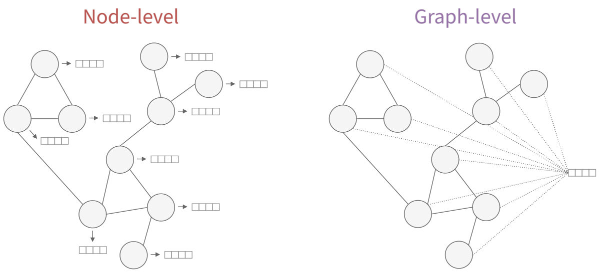
Module 2

Spatiotemporal Graph Neural Networks (STGNNs)

Message passing

Graph deep learning

Goal: Learn representations of nodes, edges, or entire graphs by exploiting **relational structure**.

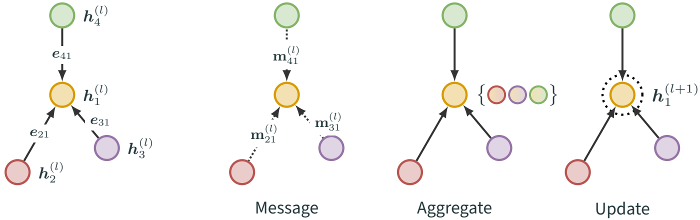


Applications include social networks, recommender systems, molecules and protein structures, and physics.

Approach used here: Use neural networks to process graphs via *message passing* operations across neighboring nodes.

Message passing

Node representations are iteratively updated through three steps:



$$\mathbf{m}_{ij}^{(l)} = \text{Message}^{(l)}\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \mathbf{e}_{ij}\right) \qquad \bar{\mathbf{h}}_i^{(l)} = \text{Aggregate}^{(l)}\left(\{\mathbf{m}_{ij}^{(l)} : j \in \mathcal{N}(i)\}\right)$$

$$\mathbf{h}_i^{(l+1)} = \text{Update}^{(l)}\left(\mathbf{h}_i^{(l)}, \bar{\mathbf{h}}_i^{(l)}\right)$$

Message passing – some remarks

$$\mathbf{h}_i^{(l+1)} = \text{Update}^{(l)}\left(\mathbf{h}_i^{(l)}, \text{Aggregate}^{(l)}\left(\left\{\text{Message}^{(l)}\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \mathbf{e}_{ij}\right) : j \in \mathcal{N}(i)\right\}\right)\right)$$

- graphs generally do not provide regular grids, nodes come in no particular order
- operations performed on graphs should not depend on the ordering of nodes (node-permutation invariance/equivariance)
  - Aggregate is desirable to be permutation invariant (sum, mean, max)
- k layers $\Rightarrow$ k-hop receptive field
- Parameters shared across nodes

Some models from the literature

Graph Convolution Network (GCN) [10]

$$\text{Message:} \quad \mathbf{m}_{ij}^{(l)} = W\mathbf{h}_j^{(l)} \qquad \text{Aggregate:} \quad \bar{\mathbf{h}}_i^{(l)} = \frac{1}{|\mathcal{N}(i)|}\sum_j \mathbf{m}_{ij}^{(l)} \qquad \text{Update:} \quad \mathbf{h}_i^{(l+1)} = \sigma(\bar{\mathbf{h}}_i^{(l)})$$

for symmetric graphs, with W a learnable weight matrix and σ an activation function.

Graph Attention Network (GAT) [11]

$$\text{Message:} \quad \mathbf{m}_{ij}^{(l)} = \alpha_{ij}\mathbf{h}_j^{(l)} \qquad \text{Aggregate:} \quad \bar{\mathbf{h}}_i^{(l)} = \sum_j \mathbf{m}_{ij}^{(l)} \qquad \text{Update:} \quad \mathbf{h}_i^{(l+1)} = \sigma(\bar{\mathbf{h}}_i^{(l)})$$

with $e_{ij} = a(W\mathbf{h}_i^{(l)}, W\mathbf{h}_j^{(l)})$, a a learnable function (e.g., MLP) and $\alpha_{ij} = \text{softmax}_j(e_{ij})$.

[10] Kipf *et al.*, “Semi-supervised classification with graph convolutional networks”, arXiv 2016.
[11] Veličković *et al.*, “Graph Attention Networks”, ICLR 2018.

Disregard node ordering

When predicting the i-th time series, say $\boldsymbol{x}^i_{t+h}$, we process information $\{\boldsymbol{x}^j_t : j \in \mathcal{N}(i)\}$ from its neighbors $\mathcal{N}(i)$. Usually the order in which they are presented carries no meaning.

Given a node permutation π, we prefer models $\mathcal{F}_\Theta$ such that

$$\mathcal{F}_\Theta\left(\boldsymbol{x}^i_t, \pi \star \mathcal{G}^i_t\right) = \mathcal{F}_\Theta\left(\boldsymbol{x}^i_t, \mathcal{G}^i_t\right).$$

where $\mathcal{G}^i$ is the subset of inputs related to node i.

Remarks:

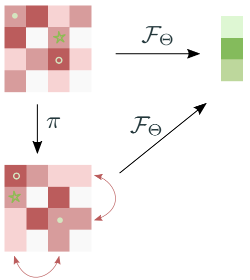
- For L-layer networks, the receptive field might extend to L-hop neighbors.
- Permuation π acts on both the time series, but the spatial graph structure.

Invariance and equivariance to node ordering

For any graph g, represented here by a weighted adjacency matrix, and any node permutation π.

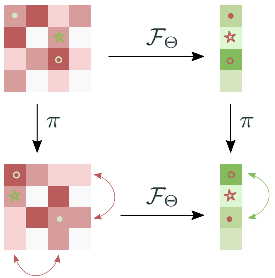
Permutation invariance

$$\mathcal{F}_\Theta\left(\pi \star g\right) = \mathcal{F}_\Theta\left(g\right)$$



Permutation equivariance

$$\mathcal{F}_\Theta\left(\pi \star g\right) = \pi \star \mathcal{F}_\Theta\left(g\right)$$



Architectures

Time-and-Space

In T&S models, representations at every node and time step are obtained by jointly propagating representation through time and space.

$$\boldsymbol{H}_{t-1}^{l+1} = \mathsf{STMP}^{l}\left(\boldsymbol{H}_{\leq t-1}^{l}, \mathcal{E}_{\leq t-1}\right)$$

Several options exist.

- Integrate MP into neural operators for sequential data.
  - Graph recurrent architectures, spatiotemporal convolutions, spatiotemporal attention, ...
- Use sequence molding operators to compute messages.
  - Temporal graph convolutions, spatiotemporal cross-attention, ...
- Product graph representations.

Example 1: From Recurrent Neural Networks...

Consider a standard GRU cell [12].

$$\boldsymbol{r}_t^i = \sigma\left(\boldsymbol{\Theta}_r\left[\boldsymbol{x}_t^i || \boldsymbol{h}_{t-1}^i\right] + \boldsymbol{b}_r\right) \tag{1}$$

$$\boldsymbol{u}_t^i = \sigma\left(\boldsymbol{\Theta}_u\left[\boldsymbol{x}_t^i || \boldsymbol{h}_{t-1}^i\right] + \boldsymbol{b}_u\right) \tag{2}$$

$$\boldsymbol{c}_t^i = \tanh\left(\boldsymbol{\Theta}_c\left[\boldsymbol{x}_t^i || \boldsymbol{r}_t^i \odot \boldsymbol{h}_{t-1}^i\right] + \boldsymbol{b}_c\right) \tag{3}$$

$$\boldsymbol{h}_t^i = \left(1 - \boldsymbol{u}_t^i\right) \odot \boldsymbol{c}_t^i + \boldsymbol{u}_t^i \odot \boldsymbol{h}_{t-1}^i \tag{4}$$

Time series are processed independently for each node or as a single multivariate time series.

[12] Chung *et al.*, "Empirical evaluation of gated recurrent neural networks on sequence modeling" 2014.

...to Graph Convolutional Recurrent Neural Networks

We can obtain a T&S model by implementing the gates of the GRU with MP blocks:

$$\boldsymbol{Z}_t^l = \boldsymbol{H}_t^{l-1} \tag{5}$$

$$\boldsymbol{R}_t^l = \sigma\left(\mathsf{MP}_r^l\left(\left[\boldsymbol{Z}_t^l || \boldsymbol{H}_{t-1}^l\right], \mathcal{E}_t\right)\right), \tag{6}$$

$$\boldsymbol{O}_t^l = \sigma\left(\mathsf{MP}_o^l\left(\left[\boldsymbol{Z}_t^l || \boldsymbol{H}_{t-1}^l\right], \mathcal{E}_t\right)\right), \tag{7}$$

$$\boldsymbol{C}_t^l = \tanh\left(\mathsf{MP}_c^l\left(\left[\boldsymbol{Z}_t^l || \boldsymbol{R}_t^l \odot \boldsymbol{H}_{t-1}^l\right], \mathcal{E}_t\right)\right), \tag{8}$$

$$\boldsymbol{H}_t^l = \boldsymbol{O}_t^l \odot \boldsymbol{H}_{t-1}^l + (1 - \boldsymbol{O}_t^l) \odot \boldsymbol{C}_t^l, \tag{9}$$

These T&S models are known as graph convolutional recurrent neural networks (GCRNNs) [13].

[13] Seo *et al.*, “Structured sequence modeling with graph convolutional recurrent networks”, ICONIP 2018.

Popular GCRNNs

The first GCRNN has been introduced in [13], with message passing (MP) blocks implemented as polynomial graph convolutional filters.

GCRNNs have become popular in traffic forecasting with the Diffusion Convolutional Recurrent Neural Network (DCRNN) architecture [14].

DCRNN relies on a bidirectional diffusion convolution:

$$\boldsymbol{H}_t' = \sum_{k=0}^{K} \left(\boldsymbol{D}_{t,\text{out}}^{-1}\boldsymbol{A}_t\right)^k \boldsymbol{H}_t \boldsymbol{\Theta}_1^{(k)} + \left(\boldsymbol{D}_{t,\text{in}}^{-1}\boldsymbol{A}_t^\top\right)^k \boldsymbol{H}_t \boldsymbol{\Theta}_2^{(k)} \tag{10}$$

[13] Seo *et al.*, “Structured sequence modeling with graph convolutional recurrent networks”, ICONIP 2018.
[14] Li *et al.*, “Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting”, ICLR 2018.

Example 2: Spatiotemporal convolutional networks (i)

Spatiotemporal convolutional networks (STCNs) instead **alternate spatial and temporal convolutions**:

1. Compute intermediate representations by using a temporal convolutional layer:

$$\boldsymbol{z}^{i,l}_{t-W:t} = \mathsf{TCN}^{l}\left(\boldsymbol{h}^{i,l-1}_{t-W:t}\right) \qquad \forall\, i$$

   where TCN^{l} indicates a temporal convolutional layer.

2. Compute the updated representation at each time step by using a graph convolution:

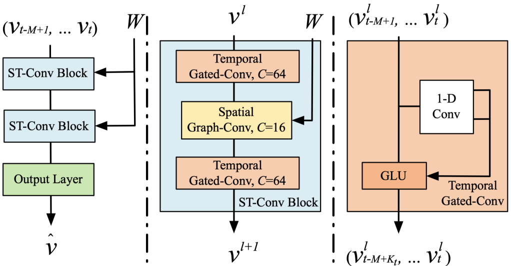
$$\boldsymbol{H}^{l}_{t} = \mathsf{MP}^{l}\left(\boldsymbol{Z}^{l}_{t}, \mathcal{E}_t\right) \qquad \forall\, t$$

Spatiotemporal convolutional networks (ii)

The first example of such architecture is the STGCN by Yu et al. [15].

The model is obtained by stacking STMP blocks consisting of

- a (gated) temporal convolution;
- a polynomial graph convolution;
- a second (gated) temporal convolution.



Courtesy of [15].

More advanced implementations exist, e.g., see Graph Wavenet [16].

[15] Yu *et al.*, “Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting”, IJCAI 2018.
[16] Wu *et al.*, “Graph wavenet for deep spatial-temporal graph modeling”, IJCAI 2019.

Example 3: Temporal Graph Convolution

A more integrated approach instead consists of implementing a temporal propagation mechanism in the message function.

For example, we can design STMP layers s.t.

$$\boldsymbol{h}^{i,l}_{t-W:t} = \mathsf{TCN}^{l}_{1}\left(\boldsymbol{h}^{i,l-1}_{t-W:t}, \underset{j\in\mathcal{N}_t(i)}{\mathrm{AGGR}}\left\{\mathsf{TCN}^{l}_{2}\left(\boldsymbol{h}^{i,l-1}_{t-W:t}, \boldsymbol{h}^{j,l-1}_{t-W:t}, \boldsymbol{e}^{ji}_{t-W:t}\right)\right\}\right).$$

Analogous models can be built with any sequence modeling architecture.

→ **Example:** many rely on attention-based operators [17][18].

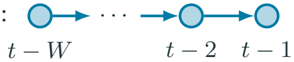
[17] Marisca *et al.*, "Learning to Reconstruct Missing Data from Spatiotemporal Graphs with Sparse Observations", NeurIPS 2022.
[18] Wu *et al.*, "TraverseNet: Unifying Space and Time in Message Passing for Traffic Forecasting", TNNLS 2022.

Example 4: Product graph representations

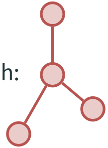
An alternative option is to consider the sequence $\mathcal{G}_{t-W:t}$ as a **single graph** with temporal and spatial edges.

In particular, **product graph representations** can be obtained by combining the two edge sets.

Temporal graph:



Spatial graph:

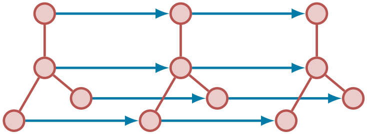


The resulting graph can be processed by any MP neural network.

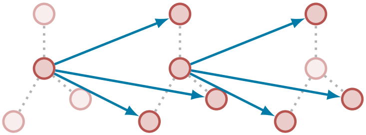
[19] Sabbaqi *et al.*, "Graph-time convolutional neural networks: Architecture and theoretical analysis", TPAMI 2023.

Building product graph representations

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



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



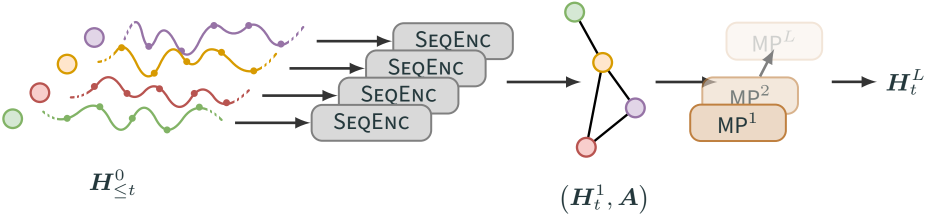
- ...

Time-then-Space models

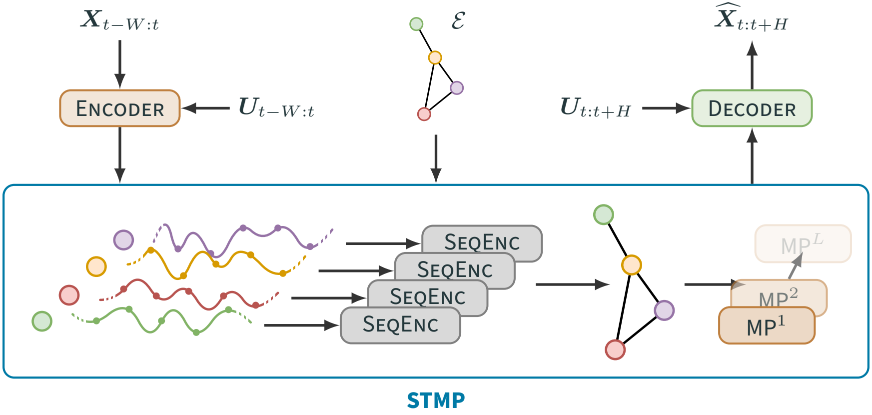
The general recipe for a TTS model consists in:

1. Embedding each node-level time series in a vector.
2. Propagating obtained encodings throughout the graph with a stack of MP layers.

1. $\boldsymbol{h}_t^{i,1} = \textsc{SeqEnc}\left(\boldsymbol{h}_{\leq t}^{i,0}\right)$ 2. $\boldsymbol{H}_t^{l+1} = \text{MP}^l\left(\boldsymbol{H}_t^l, \mathcal{E}_t\right)$



Full TTS model



Examples of TTS models

Several models can be built exploiting different sequence encodes and MP schemes.

- Popular methods have exploited both RNNs[20], [21] and MLPs/TCNs[22] as temporal encoders.
- Among message-passing implementations, both isotropic [20] and anisotropic [22] schemes have found applications.

[20] Gao *et al.*, “On the Equivalence Between Temporal and Static Equivariant Graph Representations”, ICML 2022.
[21] Cini *et al.*, “Scalable Spatiotemporal Graph Neural Networks”, AAAI 2023.
[22] Satorras *et al.*, “Multivariate time series forecasting with latent graph inference” 2022.

Pros & Cons of TTS models

Pros: ☺ Easy to implement and computationally efficient.

☺ We can reuse operators we already know.

Cons: ☹ The 2-step encoding might introduce information bottlenecks.

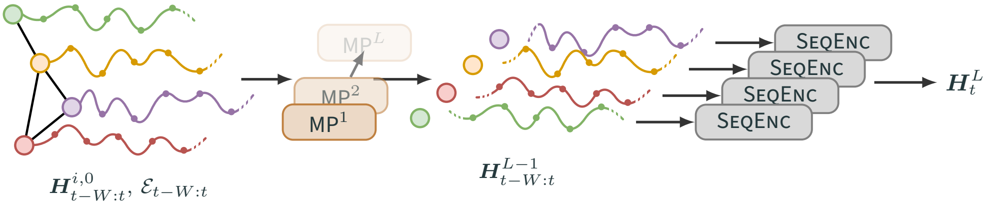
☹ Accounting for changes in topology and dynamic edge attributes can be more problematic.

Space-then-Time

In STT approaches the two processing steps of TTS models are inverted:

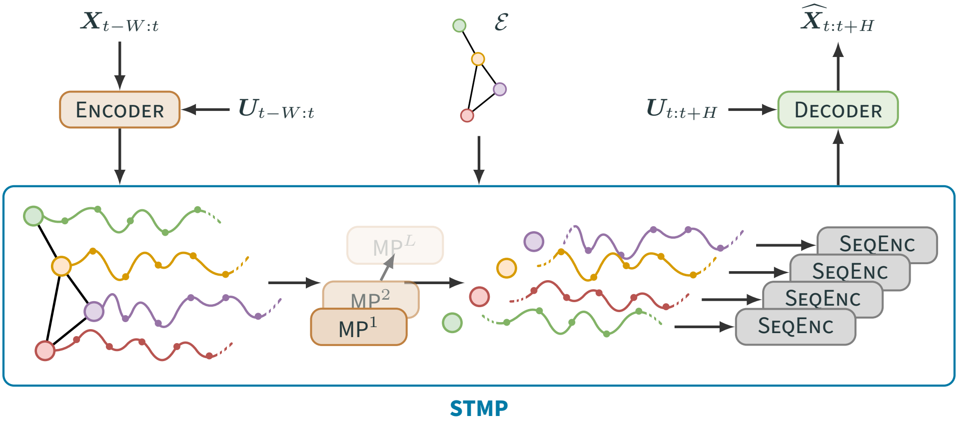
1. Observations are propagated among nodes w.r.t. each time step using a stack of MP layers.
2. Each sequence of representations is processed by a sequence encoder.

1. $\boldsymbol{H}_t^{i,l} = \mathsf{MP}^l\left(\boldsymbol{H}_t^{i,l-1}, \mathcal{E}_t\right)$ 2. $\boldsymbol{h}_t^{i,L} = \textsc{SeqEnc}\left(\boldsymbol{h}_{t-W:t}^{i,L-1}\right)$



☹ They do not have the same computational advantages of TTS models.

Full STT model



DEMO

Graph-representations and GNNs

github.com/dzambon/spatiotemporal-learning-lab

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`.