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

Missing-Data Handling

Dealing with missing data

The problem of missing data

So far, we assumed to deal with **complete sequences**.

- i.e., to have valid observations associated with each node (sensor) and time step.

However, time series collected by real-world sensor networks often have missing data, due to:

- faults, of either transient or permanent nature;
- asynchronicity among the time series;
- communication errors...

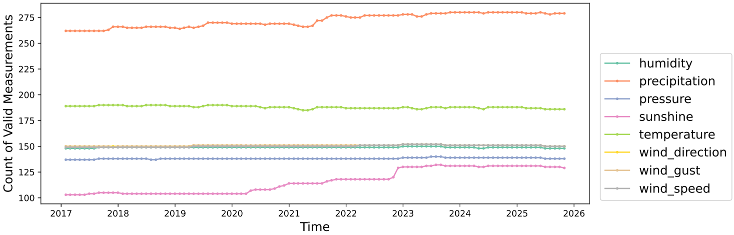
Most forecasting methods operate on complete sequences.

$\rightarrow$ We need a way to impute, i.e., *reconstruct*, missing data.

Example: Data availability in PeakWeather

	Temp.	Humidity	Precip.	Sunshine	Pressure	Wind speed	Wind gusts	Wind dir.
Rain gauges	41/38	0/0	124/141	0/0	0/0	0/0	0/0	0/0
Meteo stations	148/148	148/148	138/138	103/129	137/138	149/150	150/150	150/150
Missing values	1.94%	0.96%	3.51%	13.02%	1.39%	1.46%	1.08%	1.10%

Number of stations per variable (# in Jan 2017 / # in Oct 2025). Missing data prc only for stations with that variable.



[2] Zambon, Cattaneo, Marisca, Bhend, Nerini, and Alippi, *PeakWeather: MeteoSwiss Weather Station Measurements for Spatiotemporal Deep Learning*, Preprint 2025.

Missing data types

We can categorize missing data patterns according to the conditional distribution $p\left(\boldsymbol{m}_t^i \mid \boldsymbol{M}_{\leq t}\right)$.

- **Point missing**
  $p\left(\boldsymbol{m}_t^i = \boldsymbol{0}\right)$ is the same across nodes and time steps, i.e., RVs associated to each $\boldsymbol{m}_t^i$ are iid.

$$p\left(\boldsymbol{m}_t^i\right) = \mathcal{B}(\eta) \quad \forall\, i, t$$

- **Block missing**
  $p\left(\boldsymbol{m}_t^i = \boldsymbol{0}\right)$ is not independent from missing data at other nodes and/or time steps.

Temporal block missing $\quad p\left(\boldsymbol{m}_t^i \mid \boldsymbol{m}_{t-1}^i\right) \neq p\left(\boldsymbol{m}_t^i\right)$

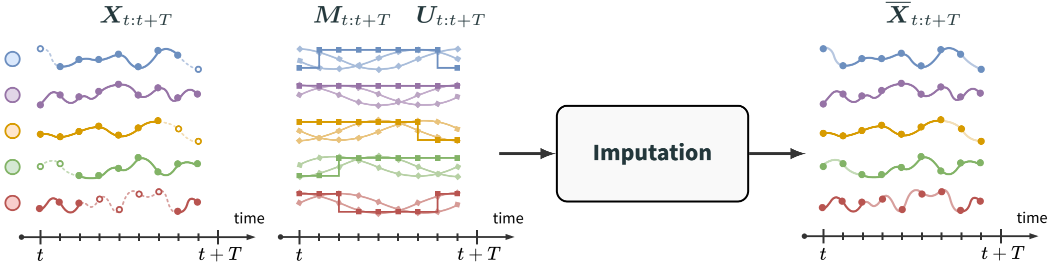
Spatial block missing $\quad p\left(\boldsymbol{m}_t^i \mid \left\{\boldsymbol{m}_t^j\right\}^{j\neq i}\right) \neq p\left(\boldsymbol{m}_t^i\right)$

Spatiotemporal block missing $\quad p\left(\boldsymbol{m}_t^i \mid \boldsymbol{m}_{t-1}^i, \left\{\boldsymbol{m}_t^j\right\}^{j\neq i}\right) \neq p\left(\boldsymbol{m}_t^i\right)$

Time series imputation

Time series imputation (TSI)

Given a window of observations $\boldsymbol{X}_{t:t+T}$, mask $\boldsymbol{M}_{t:t+T}$, and covariates $\boldsymbol{U}_{t:t+T}$, the goal is to estimate the missing observations in the sequence $\overline{\boldsymbol{X}}_{t:t+T}$.



→ We use a **mask** $m_t^i \in \{\mathbf{0}, \mathbf{1}\}$ to distinguish between missing ($\mathbf{0}$) and valid ($\mathbf{1}$) observations.

Optimization

Parameters $\boldsymbol{\theta}$ can be learned by minimizing a loss function $\ell(\,\cdot\,,\,\cdot\,)$ on valid observations in a training set:

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{\left\| \boldsymbol{m}_t^i \odot \ell\left(\hat{\boldsymbol{x}}_t^i, \boldsymbol{x}_t^i\right) \right\|_1}{\left\| \boldsymbol{m}_t^i \right\|_1}. \qquad \leftarrow \qquad \text{e.g., } \ell = \left(\hat{\boldsymbol{x}}_t^i - \boldsymbol{x}_t^i\right)^2$$

For imputation, we mark some valid observations as missing with mask $\overline{\boldsymbol{m}}_t^i$ to obtain ground-truth labels:

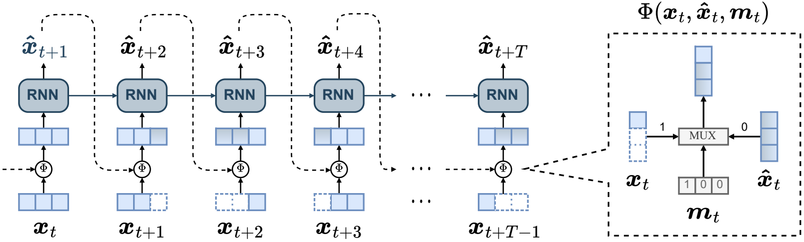
$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{\left\| \overline{\boldsymbol{m}}_t^i \odot \ell\left(\overline{\boldsymbol{x}}_t^i, \boldsymbol{x}_t^i\right) \right\|_1}{\left\| \overline{\boldsymbol{m}}_t^i \right\|_1}.$$

⚠ Data where $\overline{\boldsymbol{m}}_t^i = \mathbf{1}$ must not be used in the model to obtain the imputations.

Deep learning for TSI

Besides standard statistical methods, deep learning approaches have become a popular alternative.

- In particular, autoregressive models (e.g., RNNs).



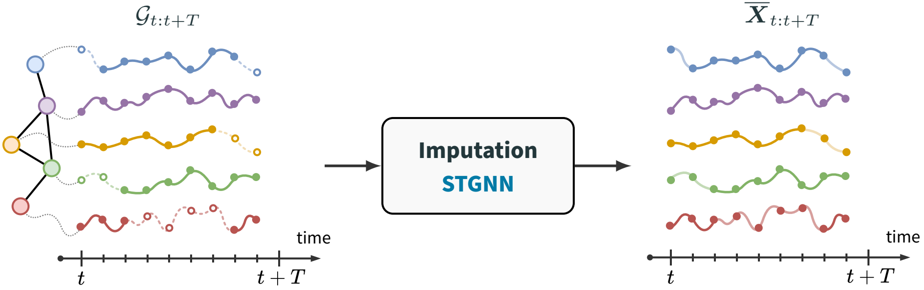
☺ Effective in exploiting past (and future, with bidirectional models) node observations.

☹ Struggle in capturing nonlinear space-time dependencies.

Time series imputation + relational inductive biases

Again, we can use the available relational information to condition the model, i.e.,

$$\boldsymbol{x}^i_{t+k} \sim p\left(\boldsymbol{x}^i_{t+k} \mid \boldsymbol{X}_{t:t+T} \odot \boldsymbol{M}_{t:t+T}, \boldsymbol{A}\right) \qquad k \in [0, T)$$



Graph Recurrent Imputation Network (GRIN)

Similarly to GCRNN for forecasting, we can integrate graph processing into the autoregressive approach for imputation [3].

In these approaches, the distribution $p\left(\boldsymbol{x}_t^i \mid \boldsymbol{X}_{0:\infty} \odot \boldsymbol{M}_{0:\infty}\right)$ is modeled into three separate steps:

Information from previous observations.	Information from subsequent observations.	Information from related concurrent observations.
$p\left(\boldsymbol{x}_t^i \mid \boldsymbol{X}_{<t} \odot \boldsymbol{M}_{<t}\right)$	$p\left(\boldsymbol{x}_t^i \mid \boldsymbol{X}_{>t} \odot \boldsymbol{M}_{>t}\right)$	$p\left(\boldsymbol{x}_t^i \mid \left\{\boldsymbol{x}_t^j \odot \boldsymbol{m}_t^j\right\}^{j \neq i}\right)$

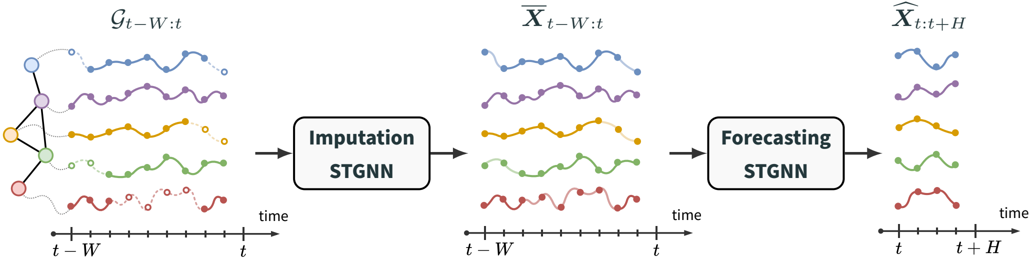
Typically modeled by bidirectional autoregressive models.

Enabled by message passing.

[3] Cini, Marisca, and Alippi, "Filling the G_ap_s: Multivariate Time Series Imputation by Graph Neural Networks", ICLR 2022.

Imputation *before* forecasting

TSI can be used as a preprocessing step for a downstream task, e.g., forecasting.

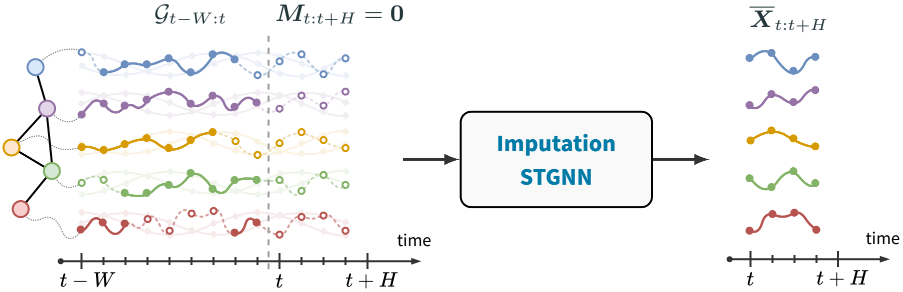


☹ Often necessary to use standard forecasting methods with irregular time series.

☹ Might introduce biases due to errors in estimated values.

Imputation *in place of* forecasting

Imputation methods can also be adapted to perform forecasting.



- ☹ It is a workaround (this is not their purpose).
- ☹ Might perform poorly due to the absence of values in the forecasting horizon.

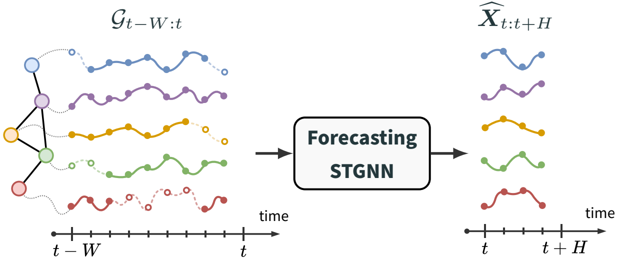
Forecasting from partial observations

A more direct approach: avoid the reconstruction step!

→ Design forecasting architecture to directly deal with irregular observations.

Benefits

- ☺ Learn how to leverage only valid observations specifically for the task at hand.
- ☺ Avoid the computational burden of imputing missing values.



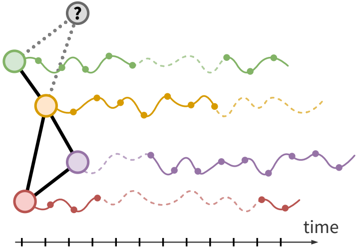
[4] Zhang *et al.*, "Graph-guided network for irregularly sampled multivariate time series", ICLR 2022.
[5] Zhong *et al.*, "Heterogeneous spatio-temporal graph convolution network for traffic forecasting with missing values", IEEE ICDCS 2021.
[6] Marisca, Alippi, and Bianchi, "Graph-based Forecasting with Missing Data through Spatiotemporal Downsampling", ICML 2024.
[7] Gravina, Zambon, Bacciu, and Alippi, "Temporal Graph ODEs for Irregularly-Sampled Time Series" 2024.

Virtual sensing

The practice of estimating unmeasured states using models and existing observations.

The power of graphs:

- ☺ The relational processing allows us to condition estimates on data close in *space*.
- ☺ The inductive property of MP allows us to handle new nodes and edges.
- ☺ Useful in applications where sensing has a cost.



[8] Wu *et al.*, "Inductive Graph Neural Networks for Spatiotemporal Kriging", AAAI 2021.
[9] De Felice, Cini, Zambon, Gusev, *et al.*, "Graph-Based Virtual Sensing from Sparse and Partial Multivariate Observations", ICLR 2024.

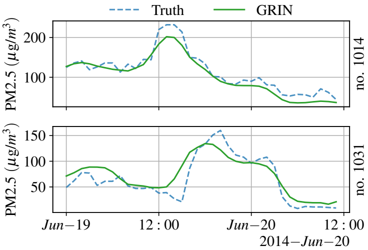
Graph imputation for virtual sensing

Add a fictitious node with **no data** and let the model infer the corresponding time series.

Clearly, several assumptions are needed

- high degree of homogeneity of sensors,
- capability to reconstruct from observations at neighboring sensors,
- and many more...

Two virtual sensors for air quality. (from [3])



[3] Cini *et al.*, “Filling the G_ap_s: Multivariate Time Series Imputation by Graph Neural Networks”, ICLR 2022.
[10] Marisca *et al.*, “Learning to Reconstruct Missing Data from Spatiotemporal Graphs with Sparse Observations”, NeurIPS 2022.

Module

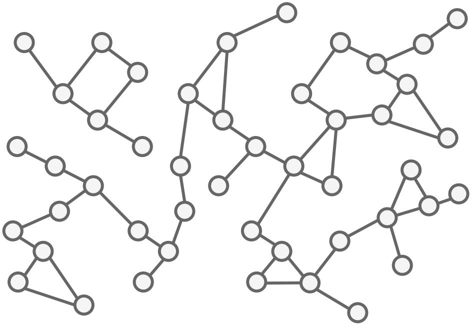
Scalability

Scalability

☺ The scalability feature

Graph-based processing allows us to

- ☺ learn a single inductive (global) model...
- ☺ ...while conditioning on related time series in a **sparse** fashion.
- ☺ The cost of this operation reduces from $\mathcal{O}(N^2)$ to $\mathcal{O}(|\mathcal{E}_t|)$



☹ The scalability issue

Spatiotemporal data span – as the name suggests – **two dimensions**:

- the spatial dimension – the number of time series.
- the time dimension – the number of time steps per time series.

In the real world, dealing with high-frequency, large-scale time series data is quite common.

– E.g., smart cities, environmental monitoring, finance

☹ A large amount of data needs to be **processed at once**.

☹ In particular, to account for **long-range** spatiotemporal dependencies.

Computational complexity of STGNNs

W: length of time series · N: number of nodes · $|\mathcal{E}_t|$: number of edges · L: number of MP layers

The computational complexity of T&S models is given by:

- node-wise temporal processing – $\mathcal{O}(WN)$;
- L MP layers **for each time step** – $\mathcal{O}(WL|\mathcal{E}_t|)$.

$$\rightarrow \mathcal{O}(W(N + L|\mathcal{E}_t|))$$

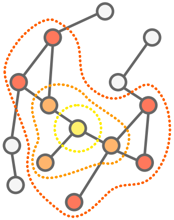
A first step toward improving scalability is represented by TTS models, which perform:

- node-wise temporal processing – $\mathcal{O}(WN)$;
- L MP layers **at the last time step** – $\mathcal{O}(L|\mathcal{E}_t|)$.

$$\rightarrow \mathcal{O}(WN + L|\mathcal{E}_t|)$$

STT models, instead, do not have computational advantages over T&S models.

Graph subsampling



Computations can be reduced by training on subgraphs of the full network.

- sampling the K**-th order neighborhood** of a subset of nodes;
- **rewiring** the graph to reduce the total number of edges.

Mostly adapted from methods developed in **static graph processing** (e.g., [12], [13]).

☹ Subsampling might break long-range spatiotemporal dependencies.

☹ The learning signal may be noisy.

[11] Gandhi *et al.*, "Spatio-Temporal Multi-graph Networks for Demand Forecasting in Online Marketplaces", ECML-PKDD 2021.
[12] Hamilton *et al.*, "Inductive representation learning on large graphs", NeurIPS 2017.
[13] Rong *et al.*, "DropEdge: Towards Deep Graph Convolutional Networks on Node Classification", ICLR 2020.

Pre-computation

Pre-processing methods (e.g., [14]) enable scalability to large graphs by:

- precomputing a representation for each node's neighborhood **ahead of training**;
- processing the obtained node representations as if they were **i.i.d. samples**.

An extension to spatiotemporal data is given by SGP [16], which acts in 2 steps:

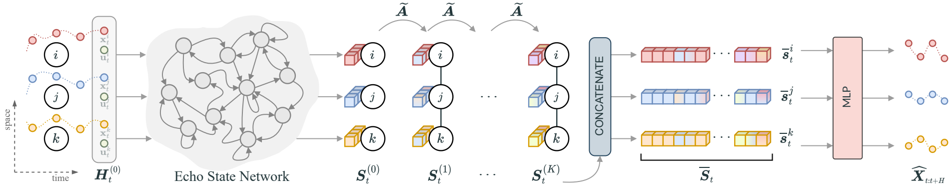
1. obtain a temporal encoding at each time step with a deep echo state network[1];
2. propagate such encodings through the graph using powers of a graph shift operator.

[14] Frasca *et al.*, "SIGN: Scalable inception graph neural networks" 2020.
[15] Zambon *et al.*, "Graph Random Neural Features for Distance-Preserving Graph Representations", ICML 2020.
[16] Cini *et al.*, "Scalable Spatiotemporal Graph Neural Networks", AAAI 2023.
[17] Liu *et al.*, "Do we really need graph neural networks for traffic forecasting?" Preprint 2023.

[1] A randomized recurrent neural networks

SGP: Scalable Graph Predictor [16]

Extracted representations can be sampled uniformly across time and space during training.



- ☺ The cost of a training step is independent of W, N and $|\mathcal{E}_t|$.
- ☺ Performance matches state of the art.
- ☹ More storage space is required – the number of extracted features is $\gg d_x$.
- ☹ More reliant on hyperparameter selection than end-to-end approaches.

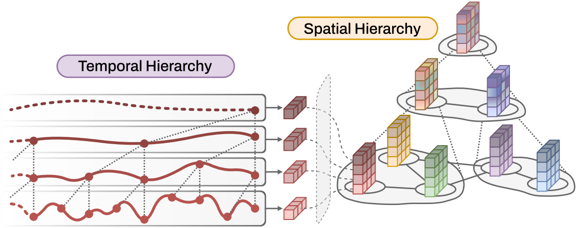
[16] Cini, Marisca, Bianchi, and Alippi, “Scalable Spatiotemporal Graph Neural Networks”, AAAI 2023.

Hierarchical processing

We can reduce computational complexity by using coarser-grained representations of the input.

In space, this can be achieved through graph pooling [18].

- ☺ Reduced number of operations to reach the same receptive field.
- ☹ Introduce bottlenecks in information propagation.



[18] Grattarola, Zambon, Bianchi, and Alippi, "Understanding Pooling in Graph Neural Networks" 2024.

References i

[1] 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.

[2] D. Zambon, M. Cattaneo, I. Marisca, J. Bhend, D. Nerini, and C. Alippi, ***Peakweather: Meteoswiss weather station measurements for spatiotemporal deep learning,*** 2025. arXiv: 2506.13652 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2506.13652.

[3] A. Cini, I. Marisca, and C. Alippi, **"Filling the g_ap_s: Multivariate time series imputation by graph neural networks,"** in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=kOu3-S3wJ7.

[4] X. Zhang, M. Zeman, T. Tsiligkaridis, and M. Zitnik, **"Graph-guided network for irregularly sampled multivariate time series,"**, 2022.

[5] W. Zhong, Q. Suo, X. Jia, A. Zhang, and L. Su, **"Heterogeneous spatio-temporal graph convolution network for traffic forecasting with missing values,"** in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, IEEE, 2021, pp. 707–717.

References ii

[6] I. Marisca, C. Alippi, and F. M. Bianchi, **"Graph-based forecasting with missing data through spatiotemporal downsampling,"** in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235, PMLR, 2024, pp. 34 846–34 865.

[7] A. Gravina, D. Zambon, D. Bacciu, and C. Alippi, **"Temporal graph ODEs for irregularly-sampled time series,"** in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, International Joint Conferences on Artificial Intelligence Organization, 2024. DOI: `10.24963/ijcai.2024/445`. [Online]. Available: `https://doi.org/10.24963/ijcai.2024/445`.

[8] Y. Wu, D. Zhuang, A. Labbe, and L. Sun, **"Inductive Graph Neural Networks for Spatiotemporal Kriging,"** in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 4478–4485.

[9] G. De Felice, A. Cini, D. Zambon, V. Gusev, and C. Alippi, **"Graph-based Virtual Sensing from Sparse and Partial Multivariate Observations,"** in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: `https://openreview.net/forum?id=CAqdG2dy5s`.

[10] 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 iii

[11] A. Gandhi, Aakanksha, S. Kaveri, and V. Chaoji, **"Spatio-temporal multi-graph networks for demand forecasting in online marketplaces,"** in *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part IV*, 2021, pp. 187–203, ISBN: 978-3-030-86513-9. DOI: 10.1007/978-3-030-86514-6_12. [Online]. Available: https://doi.org/10.1007/978-3-030-86514-6_12.

[12] W. Hamilton, Z. Ying, and J. Leskovec, **"Inductive representation learning on large graphs,"** *Advances in neural information processing systems*, vol. 30, 2017.

[13] Y. Rong, W. Huang, T. Xu, and J. Huang, **"Dropedge: Towards deep graph convolutional networks on node classification,"** in *International Conference on Learning Representations*, 2020.

[14] F. Frasca, E. Rossi, D. Eynard, B. Chamberlain, M. Bronstein, and F. Monti, **"SIGN: Scalable inception graph neural networks,"** *arXiv preprint arXiv:2004.11198*, 2020.

[15] D. Zambon, C. Alippi, and L. Livi, **"Graph Random Neural Features for Distance-Preserving Graph Representations,"** in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, H. D. III and A. Singh, Eds., ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 13–18 Jul 2020, pp. 10 968–10 977. [Online]. Available: http://proceedings.mlr.press/v119/zambon20a.html.

References iv

[16] 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.

[17] X. Liu, Y. Liang, C. Huang, *et al.*, **"Do we really need graph neural networks for traffic forecasting?"** *arXiv preprint arXiv:2301.12603*, 2023.

[18] D. Grattarola, D. Zambon, F. M. Bianchi, and C. Alippi, **"Understanding pooling in graph neural networks,"** *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 2, pp. 2708–2718, 2024.

[19] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, **"Learning laplacian matrix in smooth graph signal representations,"** *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.

[20] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, **"Connecting the dots: Identifying network structure via graph signal processing,"** *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 16–43, 2019.