Neural Forecasting

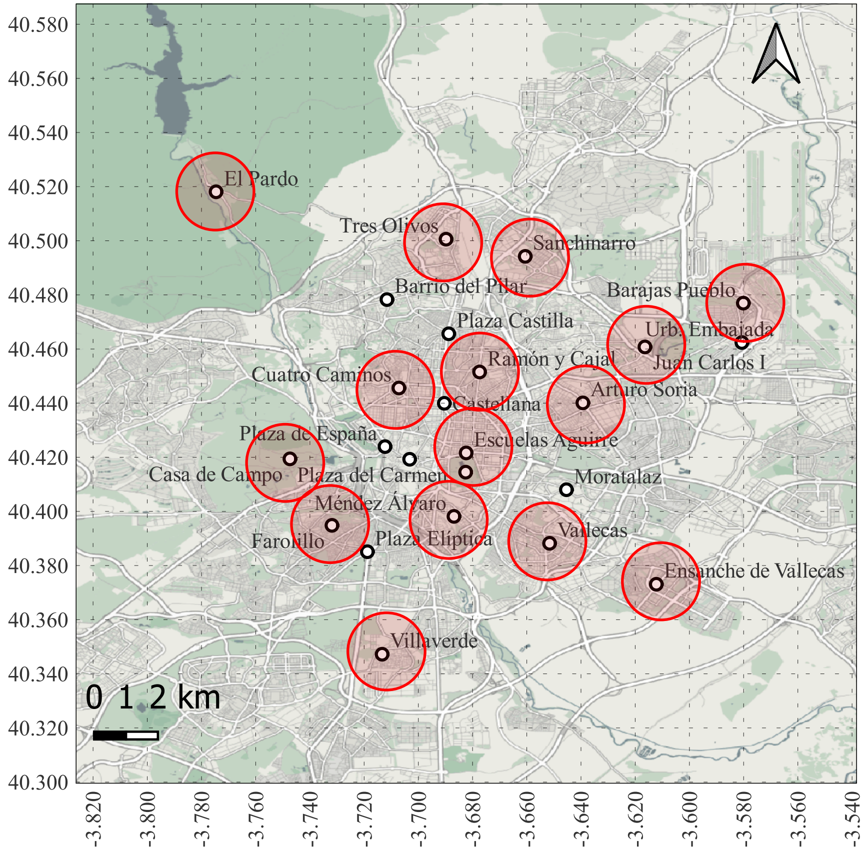
a Deep Learning approach to pollutant forecasting

Presenter: **Antonio Giganti**
antonio.giganti@polimi.it
Image and Sound Processing Lab (ISPL)
Department of Electronics, Information and Bioengineering (DEIB)
Politecnico di Milano, 20133, Milano, IT
Supervisor: Paolo Bestagini
Co-Supervisor: Sara Mandelli

Internship @ **Arianet s.r.l.**
info@aria-net.it
Via Benigno Crespi 57, 20159 Milano, IT
Supervisors:
U. Giuriato, A. D'Ausilio, C. Silibello

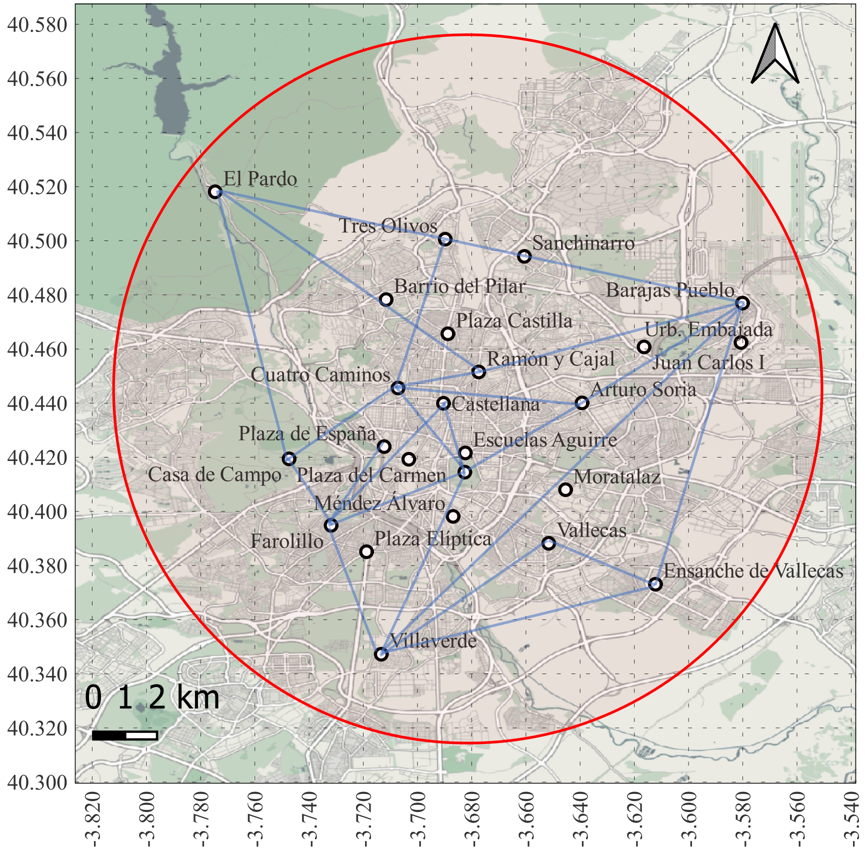
Local Model

Each monitoring station has its own model (***local***)
Training parameters are ***not shared*** among different stations



Global Model

Stations are considered part of a single model (***global***)
Training parameters are ***shared*** among different stations



= forecasting model

Covariates

A **covariate** is a time series that *may help forecast the target series*, but we are not interested in predicting. It's sometimes also called external data or exogenous variables to some extent

We could further differentiate covariates series, depending on whether they can be known in advance or not:

- **Past Covariates:** time series whose values in the forecasting horizon **are not known**. These are usually things that have to be measured, i.e. past observations

- **Future Covariates:** time series whose values in the forecasting horizon **are known**. These can for instance represent known *future holidays*, *weather forecasts,* or *weekday*

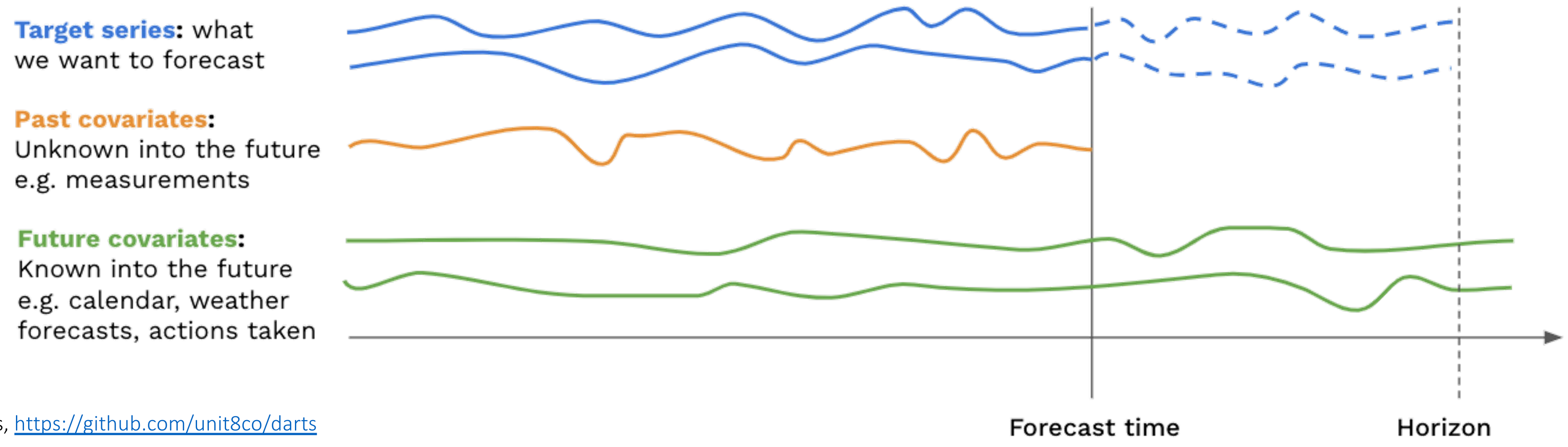


Image courtesy of Darts, https://github.com/unit8co/darts

Graph Neural Networks (GNNs) are networks for processing data that can be represented as a graph, like social network, molecular systems, a ground monitoring stations network, etc.

What is a graph?

A graph is a mathematical structures used to model (pairwise) relations between objects, i.e., *objects* (**nodes**, or vertex), *relations* (**edges**, or link). We can store information in each of these pieces of the graph

Practical applications of these architecture goes from drug discovery, physics simulations, fake news detection, traffic prediction, recommendation systems ... and time series forecasting ☺

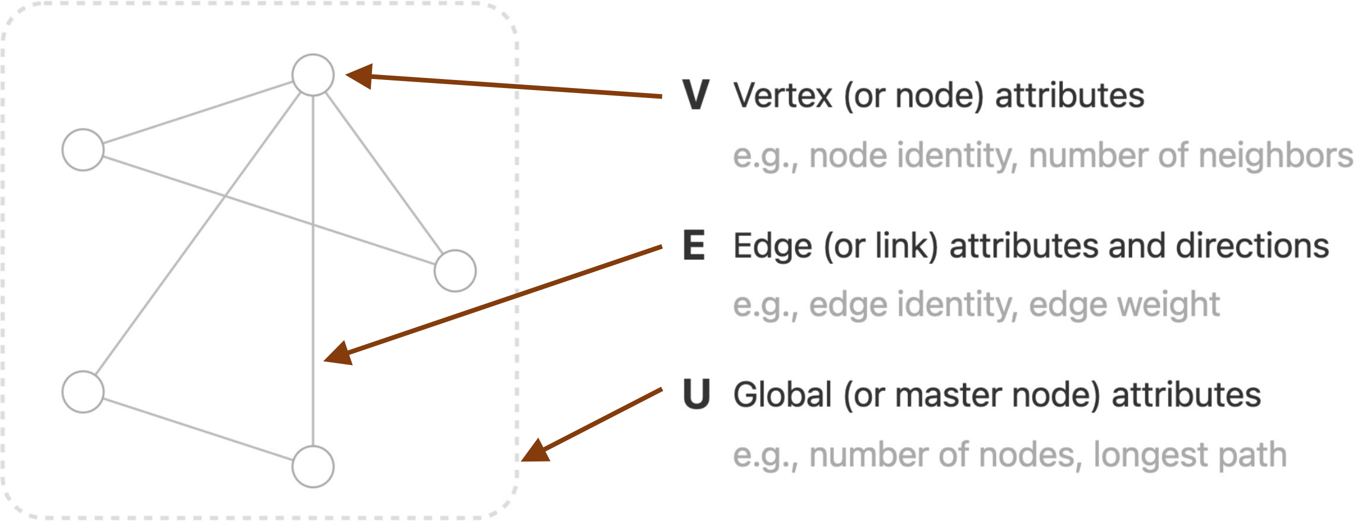


Image courtesy of an interactive introduction that can be found here: https://distill.pub/2021/gnn-intro/ Sanchez-Lengeling, et al., "A Gentle Introduction to Graph Neural Networks", Distill, 2021.
Introduction to Graph Theory: A Computer Science Perspective, https://www.youtube.com/watch?v=LFKZLXVO-Dg&ab_channel=Reducible

SpatioTemporal GNN (STGNN) are a particular type of GNN that implements mechanisms to process graph-like data that could evolve in both space and time

A representation of a network of monitoring stations is a graph where each station corresponds to a node and the functional dependencies among the stations can be seen as edges

Time-evolving observations coming from ground monitoring stations inherently contains **rich spatiotemporal dependencies**
The dependency structure of the observations can be captured, through pairwise relationships among the stations

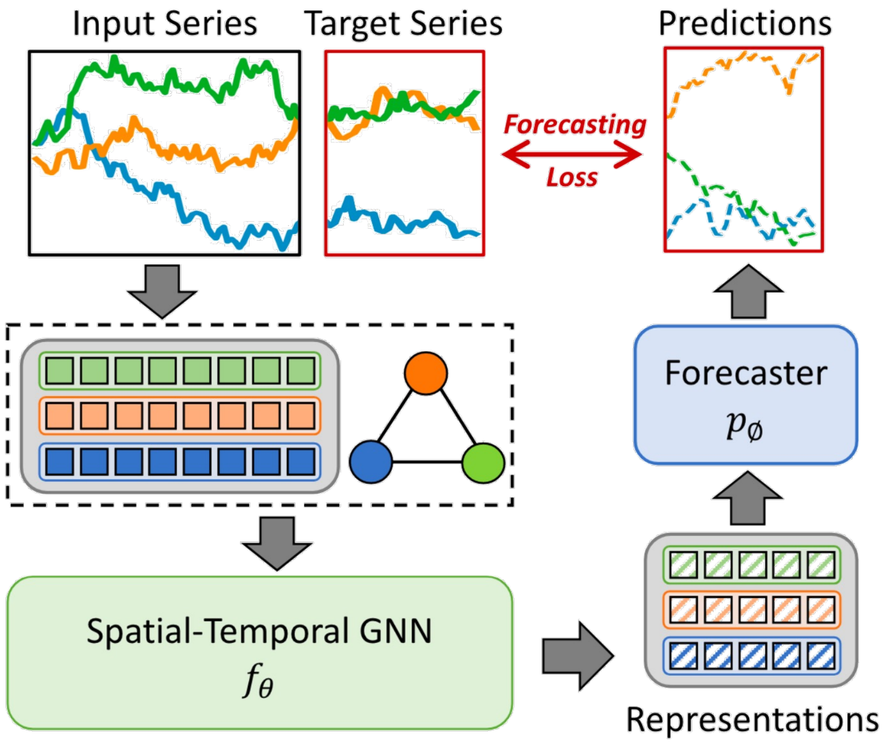


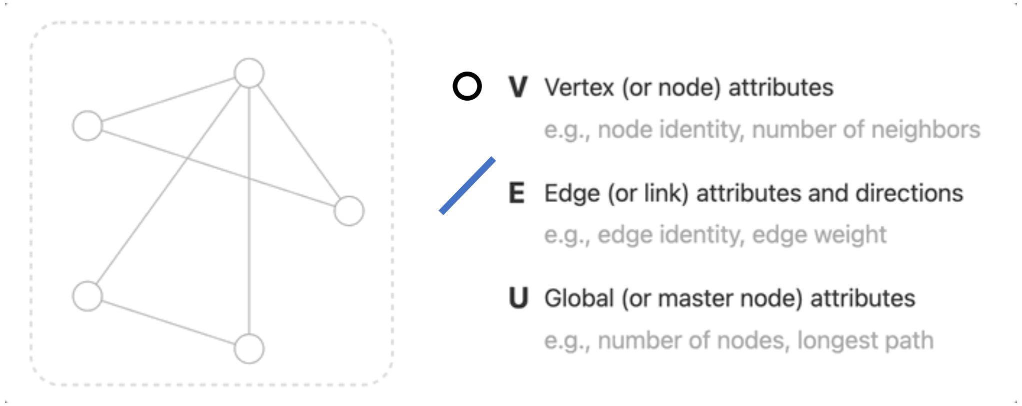
Image courtesy of Jin, M., et al., 2023. doi: 10.48550/arXiv.2307.03759

STGNN can be dissected from three aspects:

1) Modeling *spatial* (i.e., inter-variable) dependencies
2) Modeling *temporal* dependencies
3) The architectural fusion of spatial and temporal modules for time series forecasting

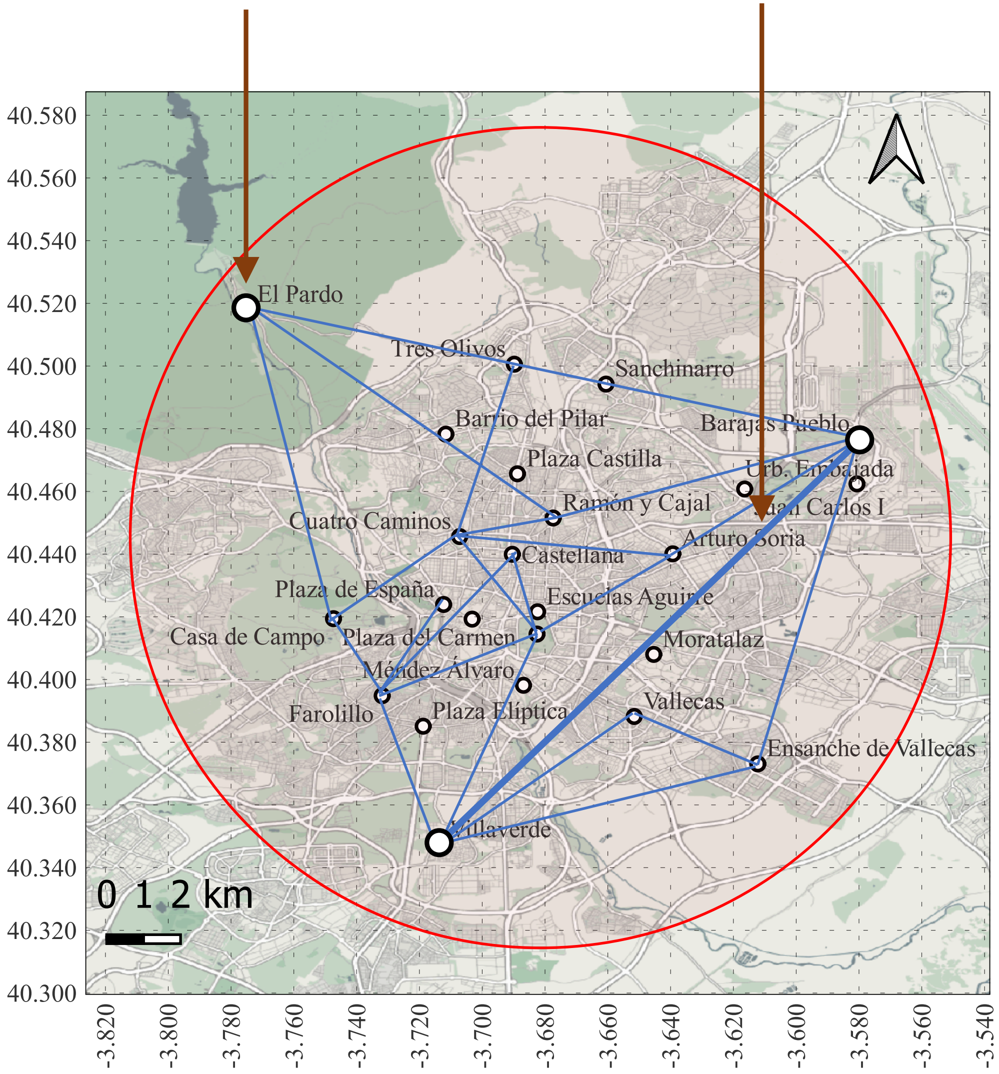
Global Model – SpatioTemporal GNN (STGNN)

A representation of a network of monitoring stations is a graph where each station corresponds to a node and the functional dependencies among the stations can be seen as edges



Stations = Graph's **Nodes**

Stations' **Relationships** = Graph's **Edges**

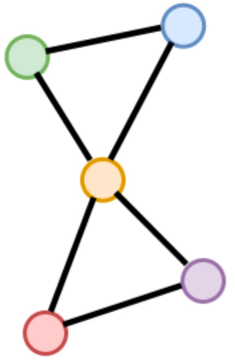


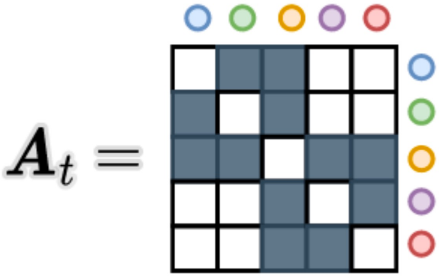
A representation of a network of monitoring stations is a graph where each station corresponds to a node and the functional dependencies among the stations can be seen as edges

Fixed Stations' Network !
Time-evolving attributes !

Nodes = stations

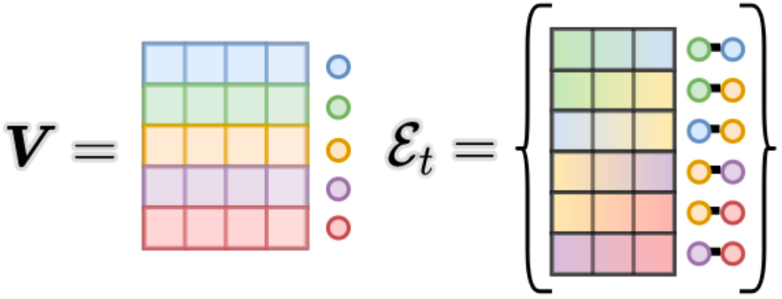
Edges = functional dependencies btw stations





A is called **Adjacency Matrix**

Nodes/Edges attributes = time series/relations



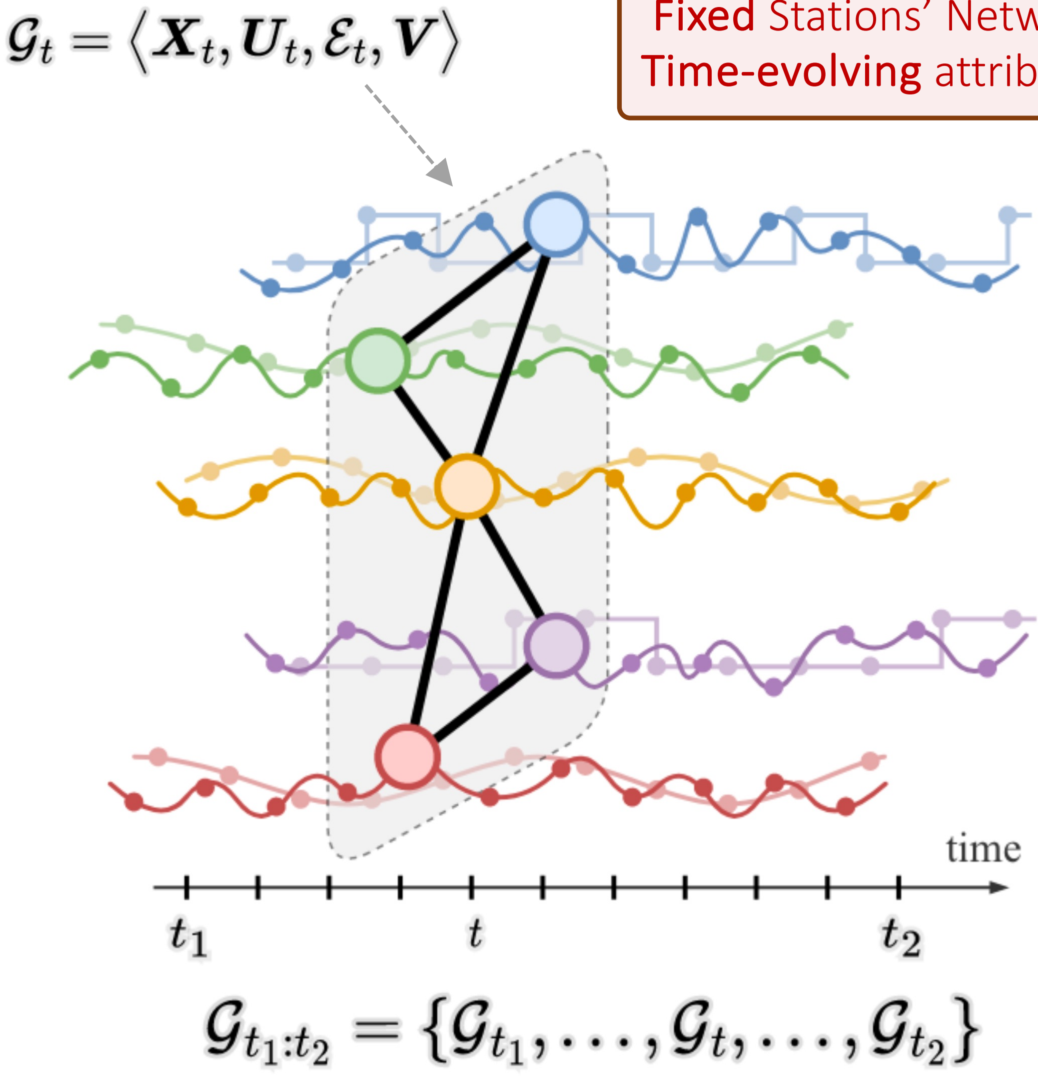
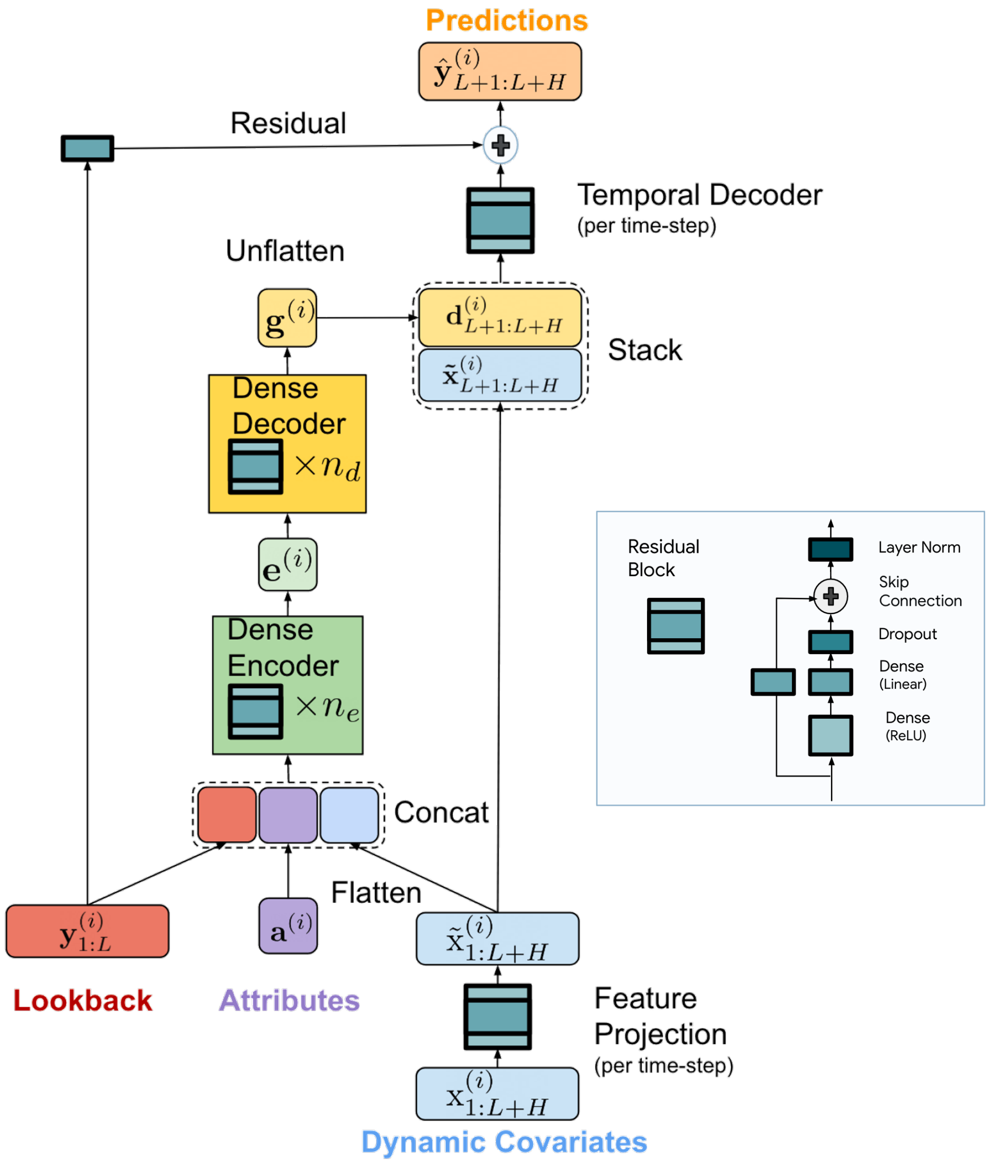


Image courtesy of Cini, A., et al., 2023. arxiv.org/abs/2310.15978

- Forecasting architecture proposed by Google Research in 2023
- Very lightweight and computational efficient
- Exploit the knowledge from past and future covariates

Simplified Working Principle

1) *TS mapping*: covariates are mapped (per time-step) to a lower dimensional space using a feature projection block
2) *TS encoding*: a concatenation of the past target $\mathbf{y}$ (lookback) along with the projected covariates, form the encoding $\mathbf{e}$
3) TS decoding: the decoder maps the previous encoding to a vector (per time-step) in the forecasting horizon, forming $\mathbf{g}$
4) *Final prediction*: a temporal decoder combines this decoded vector (per time-step) with the projected covariates of that time-step in the horizon to form the final forecast $\mathbf{y}$ hat



[1] A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, and R. Yu, “Long-term Forecasting with TiDE: Time-series Dense Encoder.” arXiv, Aug. 08, 2023. Available: http://arxiv.org/abs/2304.08424

Global-Local (Hybrid) Model

The basic idea is to **combine information** obtained using a single (**global**) model (GNN) – where the parameters are shared among different stations – with ones obtained from multiple single (**local**) models (NN) – where the parameters are related to the single model, precluding the parameter sharing between stations

These models can be denoted as Global-Local (or Hybrid) STGNN since they implement several mechanisms to combine **global** (*graph-level*) components with **local** (*station-level*) components

In this way, we can:

1) Exploit relational dependencies together flexibly and efficiently, thanks to a graph representation → *Globality*
2) Obtain accurate prediction specialized for each node, i.e., the stations of our monitoring network → *Locality*

Introducing **local** components related to a specific station's time series explicitly accounts for **node-level** effects that would not be efficiently captured by the sole **fully global models**.

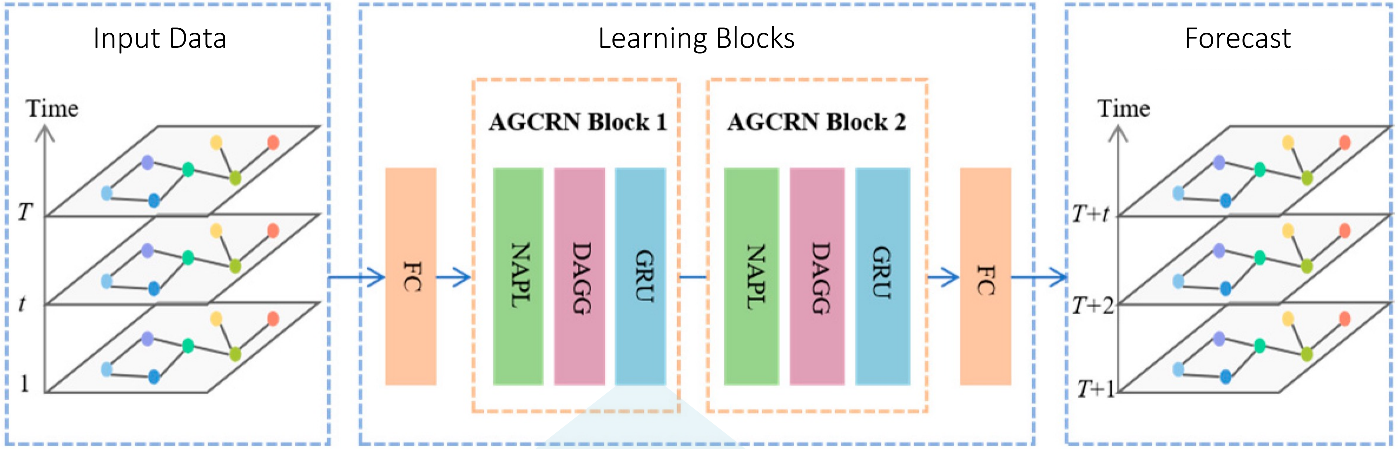
For example, node-level (station-related) effects could be the range, dynamics, and trend, strictly related to a specific station

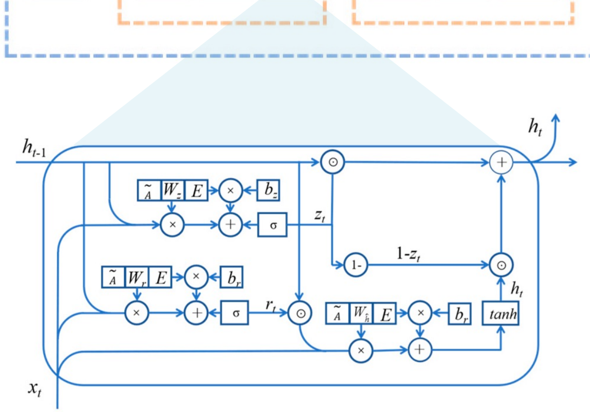
These **node-level effects** can be learned directly from data and form the so-called *learnable* node embeddings, providing a means to condition representations at each station with respect to the peculiarities of its time series

- One of the most recent hybrid STGNN is the ***A**daptive **G**raph **C**onvolutional **R**ecurrent **N**etwork* (AGCRN) [1]
- Proposed for traffic forecasting
- Learns the hidden graph topology (***A*** matrix) and combines global and local characteristics in an end-to-end fashion

Simplified Working Principle

1) *<u>Input:</u>* series of temporal consecutive graph
2) *<u>Fully Connected (FC):</u>* input mapping
3) *<u>Node Adaptive Parameter Learning (NAPL):</u>* captures specific spatial and temporal characteristics of each node
4) *<u>Data Adaptive Graph Generation (DAGG):</u>* learns the underlying graph topology from data with a node-level similarity function
5) *<u>Gated Recurrent Unit (GRU):</u>* models the temporal dynamics incorporating both spatial and temporal information
6) *<u>Fully Connected (FC):</u>* output mapping
7) *<u>Output :</u>* the prediction, thus series of temporal consecutive graph





Images courtesy of Gao, X., et al.
https://doi.org/10.3390/app132011369

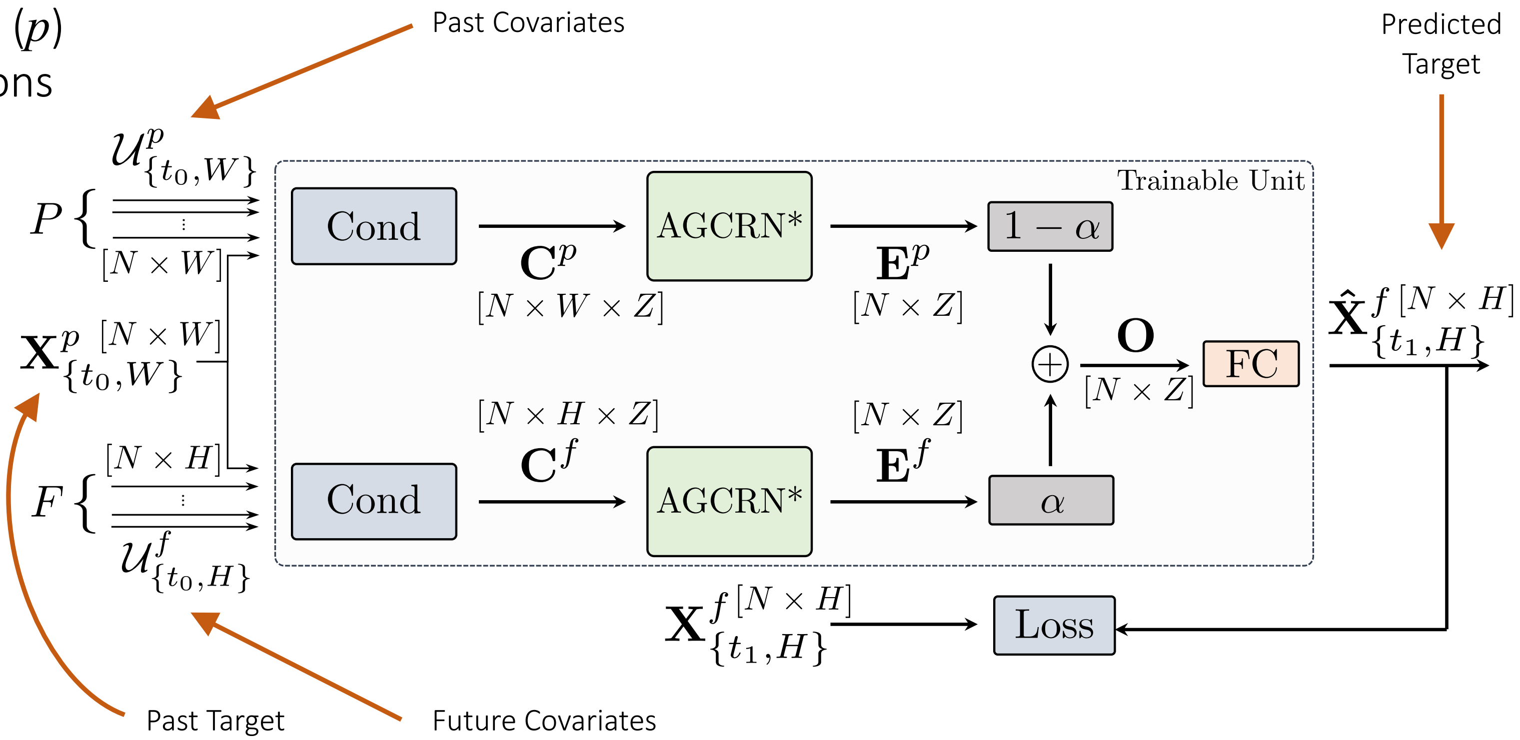
[1] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, “Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting,” Neural Information Processing Systems (NeurIPS), 2020.

Exploiting future covariates in STGNN – MAGCRN [1]

- Future covariates, like weather forecast or calendar events, could be used to improve the forecast
- Most of the STGNNs **do not directly allow for future covariates exploitation !**
- We propose the Modified AGCRN (**MAGCRN**) [1] that leverages the ability of conditioning the forecast not only on past but also on future **information**, at the same time take all the advantages of the AGCRN explained before

Simplified Working Principle

1) *Input:* series of target time series $\mathbf{X}$, and past (p) and future covariates (f) from multiple stations
2) *Conditioning:* the input time series $\mathbf{X}$ is conditioned on past and future covariates separately, forming $\mathbf{C}$
3) *Embedding estimation:* the 2 conditioning signals $\mathbf{C}$ are used to extract 2 different spatiotemporal embeddings, $\mathbf{E}$
4) *Past&Future fusion:* a combination of the 2 embeddings $\mathbf{E}$ forms the final embedding, $\mathbf{O}$
5) *Fully Connected (FC):* output mapping
6) *Output:* forecasting of the target signal $\mathbf{X}$ in multiple stations, *all at once*



[1] A. Giganti, S. Mandelli, P. Bestagini, U. Giuriato, A. D'Ausilio et al., "Back To The Future: GNN-based NO2 Forecasting Via Future Covariates". International Geoscience and Remote Sensing Symposium (IGARSS) 2024.

Exploiting future covariates in STGNN – MAGCRN [1] – Building Blocks

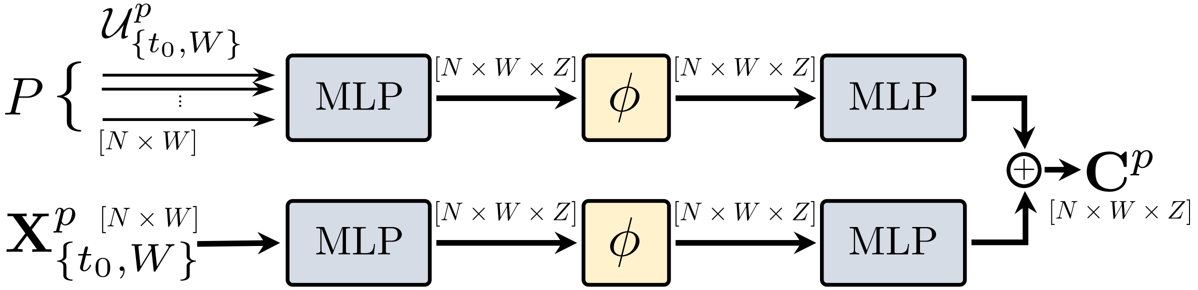
Conditioning

- The fully-learnable conditioning module perform a conditioning of the current observed time-series window, on both past (or future) covariates separately

- In general, given 2 generic data matrices, $\mathbf{K}_1, \mathbf{K}_2$, the module computes the conditioning as

$$\mathrm{Cond}(\mathbf{K}_1, \mathbf{K}_2) = \mathrm{MLP}(\phi(\mathrm{MLP}(\mathbf{K}_1))) + \mathrm{MLP}(\phi(\mathrm{MLP}(\mathbf{K}_2)))$$

- For a specific case of past covariate conditioning, we have:



Past&Future Fusion

- The Past&Future embedding fusion simply combine the 2 hidden spatiotemporal representation of the 2 branches, i.e.,

$$\mathbf{O} = (1-\alpha)\mathbf{E}^p + \alpha\mathbf{E}^f$$

- The α parameter balances the contribution between the past and the future representation of the output

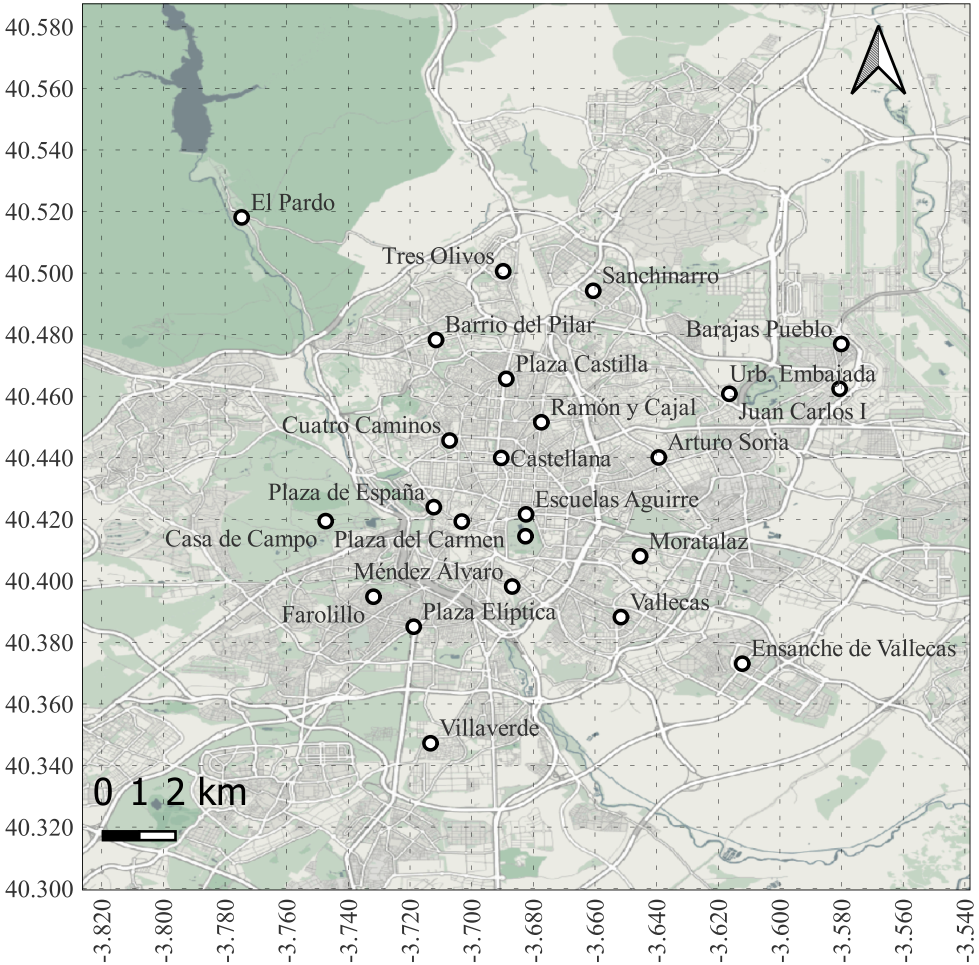
[1] A. Giganti, S. Mandelli, P. Bestagini, U. Giuriato, A. D'Ausilio et al., "Back To The Future: GNN-based NO2 Forecasting Via Future Covariates". International Geoscience and Remote Sensing Symposium (IGARSS) 2024.

Use Case

📈 NO_2 Forecasting 📉

Dataset

We conduct experiments on a recently released real-world dataset composed of **hourly-sampled** air quality, meteorological and traffic data from **24** different ground monitoring **stations** in the city of Madrid (Spain), from **January to June 2019** [1]



The released data include:

- *Air quality data* — NO_2 concentration [$\mu g/m^3$]
- *Meteorological data* — wind speed [*m/s*], wind direction [*rad*], temperature [$^\circ C$], relative humidity [*%*], barometric pressure [*mb*], solar irradiance [W/m^2]
- *Traffic data* — intensity [*vehicles/h*], occupancy time [*%*], load (degree of congestion) [*%*], average traffic speed [*km/h*]

Target:	NO_2 concentration
Past Covariates:	Traffic Data + Calendar
Future Covariates:	Meteorological Data + Calendar

[1] D. Iskandaryan, F. Ramos, and S. Trilles, “Graph Neural Network for Air Quality Prediction: A Case Study in Madrid,” IEEE Access, vol. 11, pp. 2729–2742, 2023, doi: 10.1109/ACCESS.2023.3234214.

Network Baselines

We select 3 STGNN that are the state-of-the-art for time series forecasting:

- ***Graph WaveNet*** [1], which integrates diffusion graph convolutions with dilated convolutions to capture spatial and temporal dependencies more effectively
- ***Gated GNN*** *(GGNN)* [2], that integrates gated mechanisms and residual connections for effective information propagation along the graph
- ***AGCRN*** [3], the reference architecture which our method is built upon and explained before

Although very recent, none of these STGNNs directly supports **future covariates**, contrarily to our proposed MAGCRN

In addition, we focus on STGNNs that directly **learn the adjacency matrix** (*latent graph*) of the graph nodes (*nodes relationships*), since predefined matrices usually discard additional spurious correlation between them, i.e., our monitoring stations

[1] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph wavenet for deep spatial-temporal graph modeling,” International Joint Conference on Artificial Intelligence (IJCAI) 2019, pp. 1907–1913.
[2] V. G. Satorras, S. S. Rangapuram, and T. Januschowski, “Multivariate Time Series Forecasting with Latent Graph Inference.” arXiv, 2022. doi: 10.48550/arXiv.2203.03423.
[3] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, “Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting,” Neural Information Processing Systems (NeurIPS), 2020.

Results

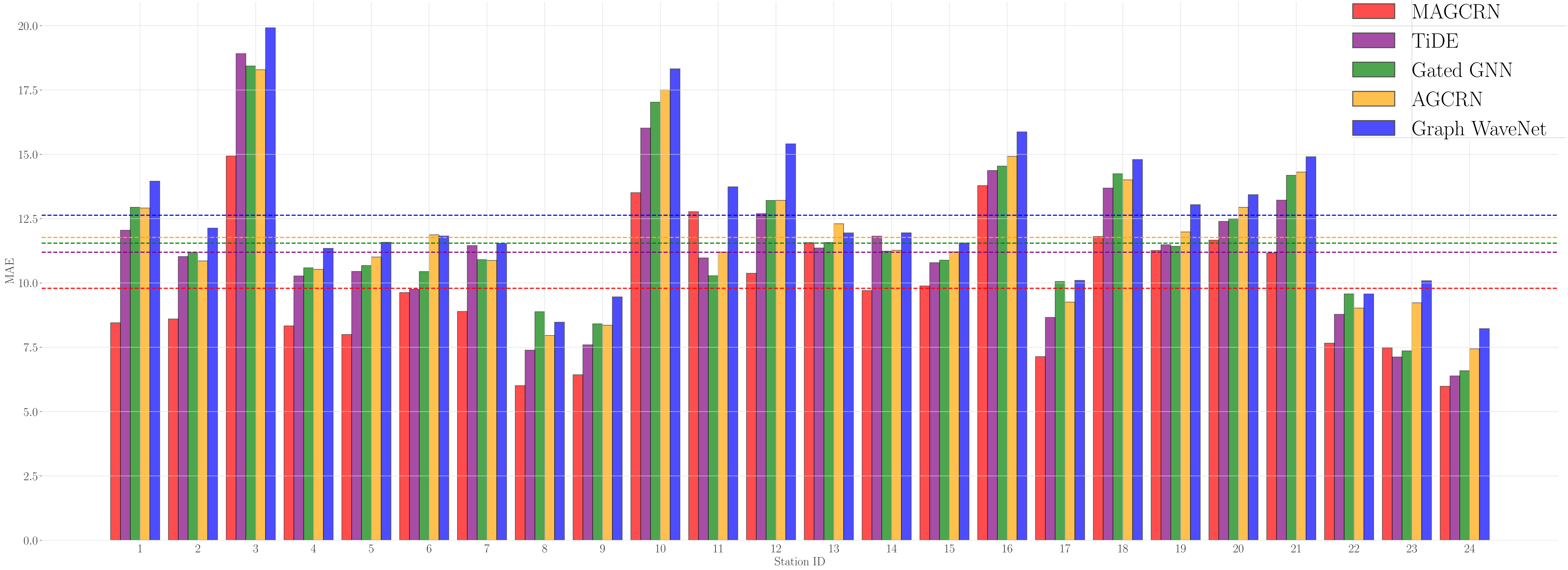
... actually, some of the them ☺

Results – Global – Graph Neural Network

Target: NO_2 concentration
Past Covariates: Traffic Data + Calendar
Future Covariates: Meteorological Data + Calendar

Network Comparison

- We compare the performance of multiple GNN-based (*global*) forecasting architecture and the previous TiDE (*local*) model
- We compute the **MAE** value **for each station** in the observation network



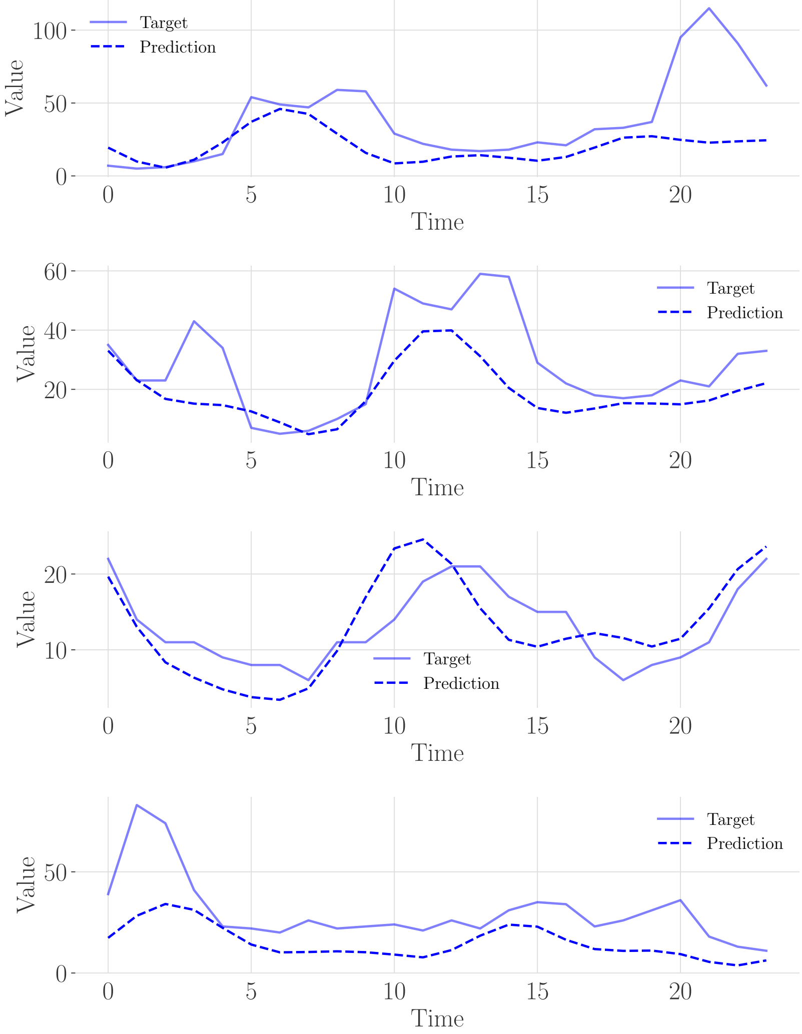
- Our proposed MAGCNR, almost all the time outperforms all the considered state-of-the-art DL-based forecasting models (red bars). The dashed (-----) lines denote the mean MAE values among all the stations, for each model

Results – Global – Graph Neural Network

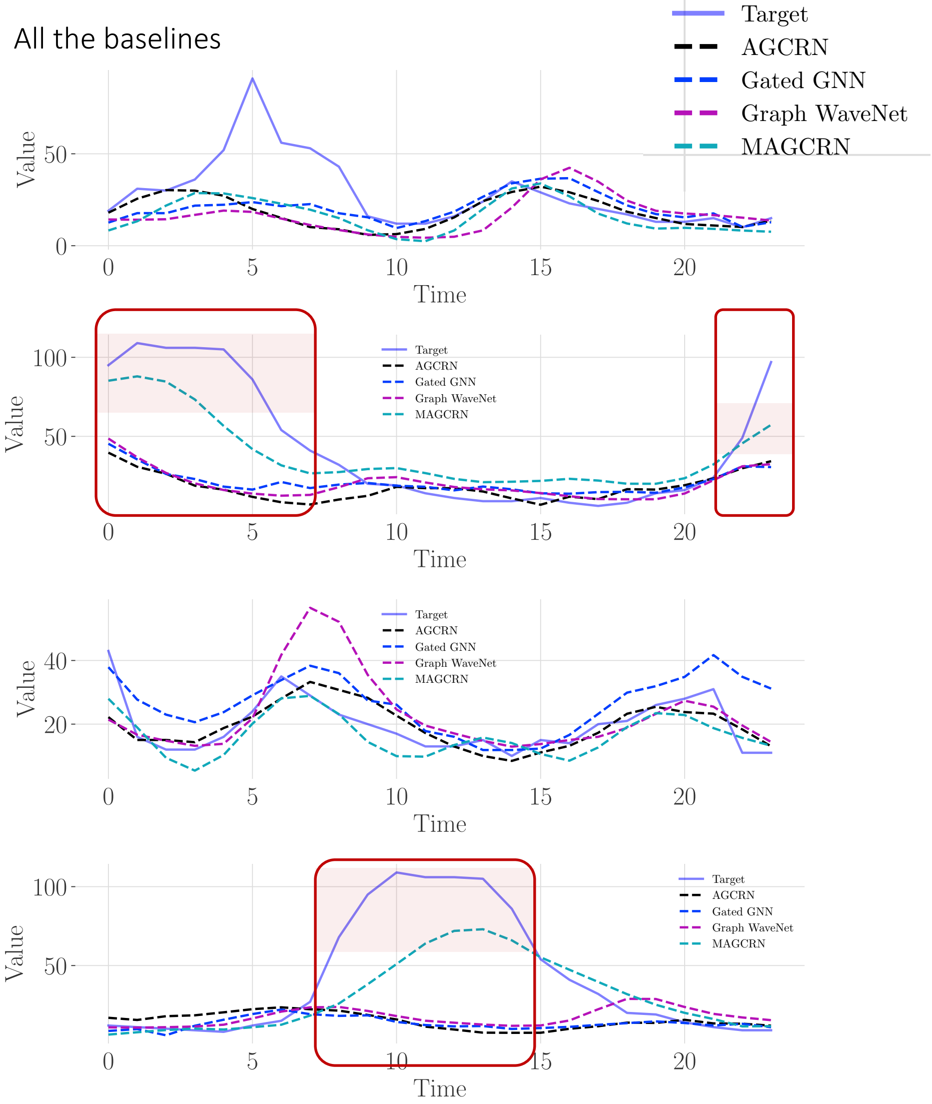
Target: NO_2 concentration
Past Covariates: Traffic Data + Calendar
Future Covariates: Meteorological Data + Calendar

Prediction Examples

MAGCRN only



All the baselines



Examples of predictions from station 1 (right) – MAGCRN only – and 20 (left) – all the selected baselines. The time is in hours, the value in the predicted NO2 concentration at the station
Notice how – **red** boxes, right figure – MAGCRN is the only one that tries to recover concentration maxima, and the others fail to predict it

Results – Global – Graph Neural Network

Target: NO_2 concentration
Past Covariates: Traffic Data + Calendar
Future Covariates: Meteorological Data + Calendar

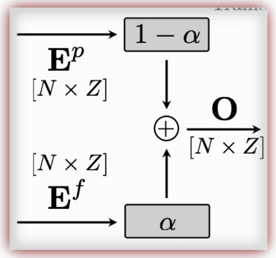
Parameter Sensitivity

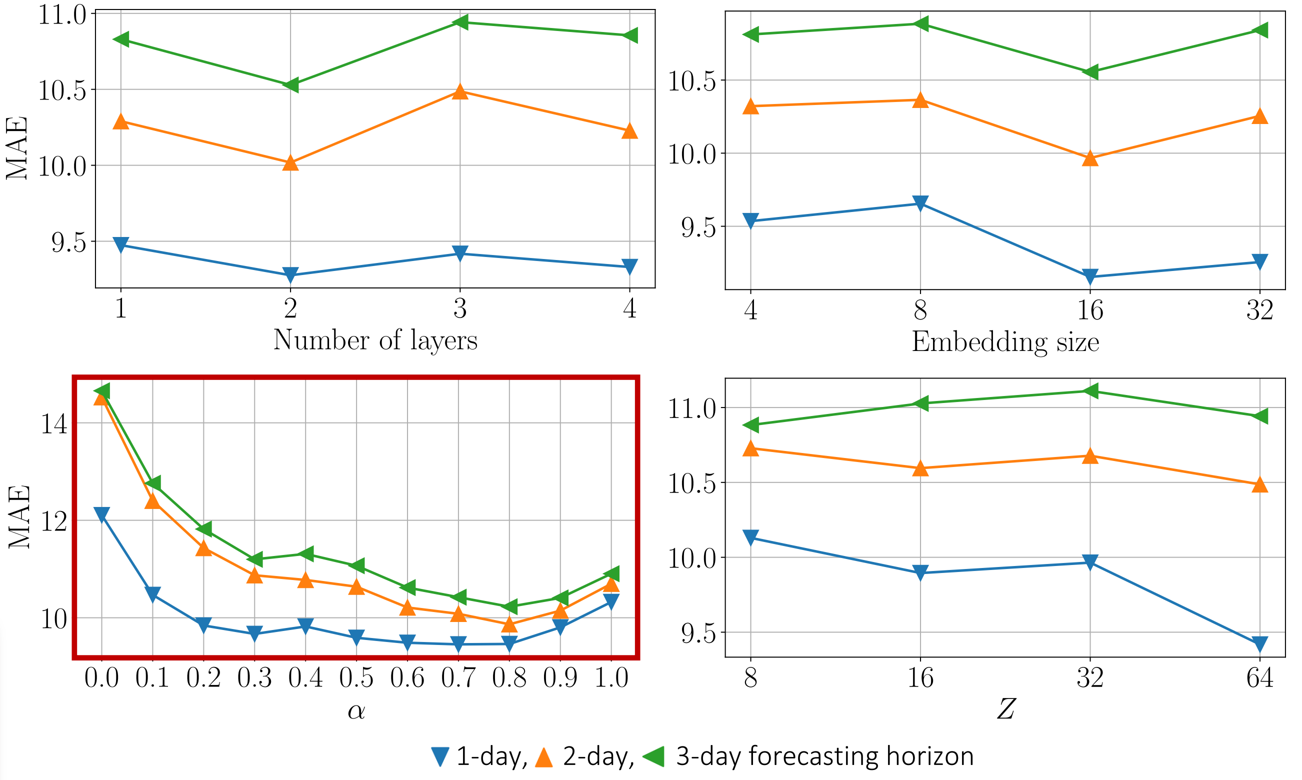
We explored the role of (from right to left):

1) the number of layers of the AGCRN module
2) the size of the node's (station's) embeddings
3) the α factor used to balance the action between past and future conditioning
4) the size of the hidden representation, Z

We report the achieved MAE as a function of the analyzed parameters, for different forecasting horizons*

*Each time we vary a parameter, we set the other to its default value
Embedding size=10, Num. Layers=3, Z=64 and α =0.5





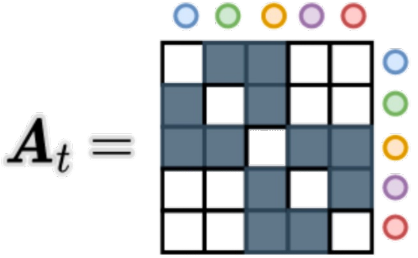
- ➢ From the figure referred to α, it is possible to notice a particular behavior, consistent among all the forecasting horizons: the **future** weather conditions are more important (*higher* α) for the final the forecasting task, compared to the **past** traffic conditions (*lower* α), leading to a **lower** ⬇⬇⬇ MAE
- ➢ For the forecasting horizons that are distant to the one adopted in training (2-day and 3-day), the future conditioning becomes even more crucial to achieve acceptable results

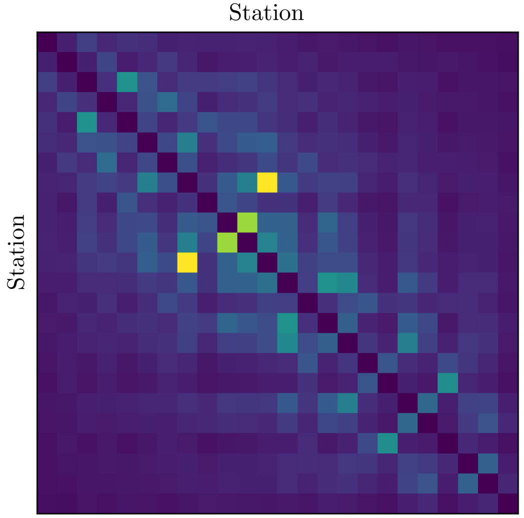
Results – Global – Graph Neural Network

Target: NO_2 concentration
Past Covariates: Traffic Data + Calendar
Future Covariates: Meteorological Data + Calendar

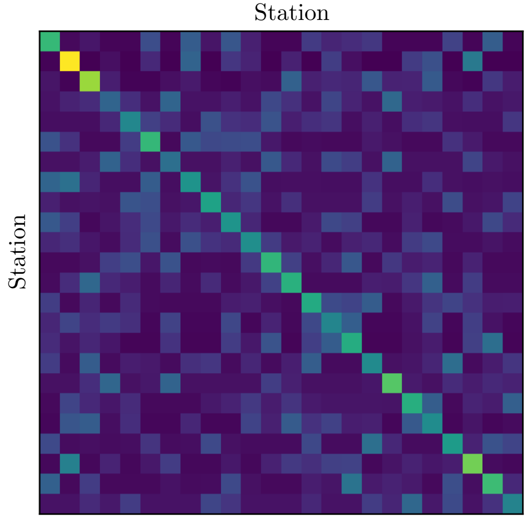
Latent Graph Learning

- Most of the GNNs for forecasting, used a **predefined** graph topology – defined by a weighted **adjacency matrix** (*A*) – thus **imposing *a priori*** station-to-station **relationships information**
- Usually - in the Air Quality forecasting context - the ***A*** values are the inverse of the *geographical distances* between stations
- This **prevents the possibility to consider additional correlations** that could exists between stations
- For this reason, we let the network **learn these additional correlation directly from the data**
- In addition, having 2 different branches (Past&Future conditioning of the input), we have 2 different **adjacency matrix** (*A*)

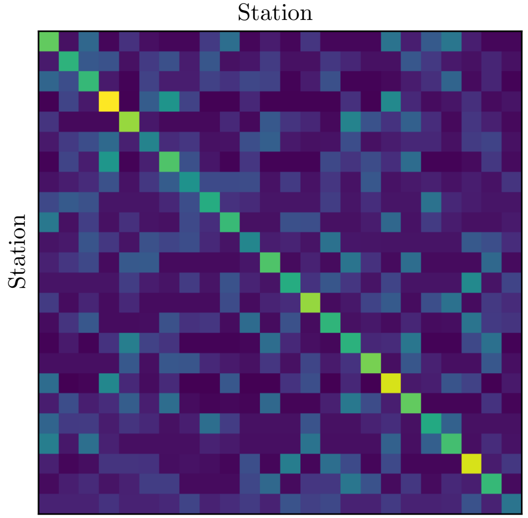




A-priori A from stations' **geographical distances**



Learned ***A*** from **Past** Covariates



Learned ***A*** from **Future** Covariates

- We can see that past and future covariates conditioning, leads to **different stations correlations**, validating the approach of considering at the same time, both conditioning in the forecasting process

Future Works

- Asses the model performance using a more rigorous forecasting metrics, e.g. *False Alarms, Missed Alarms, Good Above Threshold, Good Below Threshold, Index of Agreement* etc.

- Test the model on a longer, sparse and topologically varied dataset e.g. on a *regional*, even *national* scale

- Better handle extreme events, e.g. concentration peaks

- Proper integration of future covariates directly into the GNN’s *message-passing* process

- Explore dynamic graph structures, e.g. a *time-dependent* adjacency matrix (one for each time step)

Publications

Part of this work was accepted for publication at the *2024 IEEE International Geoscience and Remote Sensing Symposium (IGARSS24)*

https://www.2024.ieeeigarss.org/

IEEE IGARSS 2024 | July 7-12 ATHENS GREECE

made with:

PyTorch

Torch Spatiotemporal

PyTorch Lightning

Darts

A. Giganti, S. Mandelli, P. Bestagini, U. Giuriato, A. D'Ausilio et al.,
Back To The Future: *GNN-based NO2 Forecasting Via Future Covariates.*
2024 IEEE International Geoscience and Remote Sensing Symposium (IGARSS24)

BACK TO THE FUTURE

BACK TO THE FUTURE: GNN-BASED NO_2 FORECASTING VIA FUTURE COVARIATES

Antonio Giganti[1], *Sara Mandelli*[1], *Paolo Bestagini*[1],
Umberto Giuriato[2], *Alessandro D'Ausilio*[2], *Marco Marcon*[1], *Stefano Tubaro*[1]

[1]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano - Milan, Italy
[2]ARIANET srl, Via Benigno Crespi 52, Milan, Italy

ABSTRACT

Due to the latest environmental concerns in keeping at bay contaminants emissions in urban areas, air pollution forecasting has been rising the forefront of all researchers around the world. When predicting pollutant concentrations, it is common to include the effects of environmental factors that influence these concentrations within an extended period, like traffic, meteorological conditions and geographical information. Most of the existing approaches exploit this information as *past* covariates, i.e., past exogenous variables that affected the pollutant but were not affected by it. In this paper, we present a novel forecasting methodology to predict NO_2 concentration via both past and *future* covariates. Future covariates are represented by weather forecasts and future calendar events, which are already known at prediction time. In particular, we deal with air quality observations in a city-wide network of ground monitoring stations, modeling the data structure and estimating the predictions with a Spatiotemporal Graph Neural Network (STGNN). We propose a conditioning block that embeds past and future covariates into the current observations. After extracting meaningful spatiotemporal representations, these are fused together and projected into the forecasting horizon to generate the final prediction. To the best of our knowledge, it is the first time that future covariates are included in time series predictions in a structured way. Remarkably, we find that conditioning on future weather information has a greater impact than considering past traffic conditions. We release our code implementation at https://github.com/polimi-ispl/MAGCRN.

Index Terms— Air Quality Forecasting, Spatiotemporal Data, Graph Neural Network, Graph-based Forecasting, Urban Computing

1. INTRODUCTION

In the landscape of 21st-century environmental concerns, air pollution forecasting has risen to the forefront, capturing widespread attention at a global scale. Addressing this challenge, numerous researchers have delved into the exploration of effective solutions, endeavoring to precisely forecast air pollutant concentrations through a variety of methods. Among these approaches, Deep Learning (DL) methods currently hold the predominant interest, marking significant advancements in the efforts to forecasting air quality parameters and mitigating the impact of air pollution [1]–[3].

Pollutant concentrations exhibit intricate correlations in both temporal and spatial domains, and these correlations dynamically evolve over time [4]–[6]. Recently, DL-based approaches have demonstrated considerable success in pollutant forecasting by capturing nonlinear temporal and spatial patterns very effectively [2], [7].

In the last few years, there has been a surge in the popularity of applying Spatiotemporal Graph Neural Networks (STGNNs) to pollutants forecasting [8]–[14] since they can process the data structure by modeling it as a graph. Indeed, the underlying assumption is to incorporate the data structure as an inductive bias [15]–[17]. Time-evolving observations coming from ground monitoring stations inherently contains rich spatiotemporal structure and spatiotemporal dynamics. The dependency structure of the observations can be captured through pairwise relationships among the stations. These representations form a graph where each station corresponds to a node and the functional dependencies among the stations can be seen as edges.

Most of the existing approaches capture spatial dependencies on a fixed graph structure, assuming that the underlying relationship between entities is fixed and pre-determined [12], [18]–[20]. However, the explicit graph structure may not necessarily reflect the true dependency between measurements and existing relationships may be missing due to incomplete data connections [21], [22]. Recently, graph learning methods have been proposed [23] that allow to learn the graph structure directly from data. These techniques are promising, pushing forward the forecasting abilities [21], [24], [25].

In the context of air pollutant forecasting like NO_2, PM_{10} and $PM_{2.5}$, it is reasonable to assume that including auxiliary information helps the prediction performance. This is motivated by the fact that air quality is often influenced by exogenous variables like traffic and weather conditions. For example, traffic-related emissions have been one of the top contributors to air pollution in many cities around the world. It has been proved that these emissions can deteriorate ambient air quality on a large spatial scale, especially during the morning and evening rush hours in urban regions [4], [8], [18]. In addition, studies have shown that air pollutants vary under different meteorological conditions [5]. Indeed, the temperature affects the atmospheric and ventilation conditions; humidity and precipitation can change the deposition characteristics of particulate matter; wind speed promotes the diffusion and spread of pollutants [26].

All this information could be included in the air quality forecasting as covariates, i.e., exogenous variables that affect the pollutant to predict, but are not affected by it. For instance, past covariates are represented by traffic conditions, meteorological factors and geographical information. Future covariates are all the future-related information that are known in advance, like weather forecasts and calendar events (seasons, days of a week, time of a day) [4].

Past covariates have been successfully exploited in [11]–[14], [18], [19]. To the best of our knowledge, very few STGNN-based

This work was supported by the Italian Ministry of University and Research (MUR) and the European Union (EU) under the PON/REACT project.

POLITECNICO MILANO 1863

ARIANET MODELLISTICA ATMOSFERICA

suez

ISPL Image and Sound Processing Lab

Thanks for your attention !

Any questions ?

arXiv

A. Giganti, et al.,
Back To The Future:
GNN-based NO2 Forecasting
Via Future Covariates.
IGARSS'24

Presenter: **Antonio Giganti**
antonio.giganti@polimi.it
Image and Sound Processing Lab (ISPL)
Department of Electronics, Information and Bioengineering (DEIB)
Politecnico di Milano, 20133, Milano, IT
Supervisor: Paolo Bestagini
Co-Supervisor: Sara Mandelli

PON RICERCA E INNOVAZIONE 2014 - 2020

UNIONE EUROPEA
Fondo Sociale Europeo

MUR

Internship @ **Arianet s.r.l.**
info@aria-net.it
Via Benigno Crespi 57, 20159 Milano, IT
Supervisors:
U. Giuriato, A. D'Ausilio, C. Silibello

Further Readings

[1] K. Benidis et al., “Deep Learning for Time Series Forecasting: Tutorial and Literature Survey,” ACM Comput. Surv., vol. 55, no. 6, pp. 1–36, Jul. 2023, doi: 10.1145/3533382

[2] Z. Shao et al., “Exploring Progress in Multivariate Time Series Forecasting: Comprehensive Benchmarking and Heterogeneity Analysis.” arXiv, Oct. 09, 2023. http://arxiv.org/abs/2310.06119

[3] B. Zhang et al., “Deep learning for air pollutant concentration prediction: A review,” Atmospheric Environment, vol. 290, p. 119347, Dec. 2022, doi: 10.1016/j.atmosenv.2022.119347

[4] A. Longa et al., “Graph Neural Networks for temporal graphs: State of the art, open challenges, and opportunities.” arXiv, May 07, 2023. Available: http://arxiv.org/abs/2302.01018

[5] M. Jin et al., “A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection.” arXiv, Jul. 07, 2023. doi: 10.48550/arXiv.2307.03759

[6] A. Cini, D. Zambon, and C. Alippi, “Sparse Graph Learning from Spatiotemporal Time Series,” 2023

[7] A. Cini, I. Marisca, D. Zambon, and C. Alippi, “Graph Deep Learning for Time Series Forecasting.” arXiv, Oct. 24, 2023. Available: http://arxiv.org/abs/2310.15978

[8] P. Montero-Manso and R. J. Hyndman, “Principles and algorithms for forecasting groups of time series: Locality and globality,” doi: 10.1016/j.ijforecast.2021.03.004

[9] A. Zeng, M. Chen, L. Zhang, and Q. Xu, “Are Transformers Effective for Time Series Forecasting?,” AAAI, vol. 37, no. 9, pp. 11121–11128, Jun. 2023, doi: 10.1609/aaai.v37i9.26317

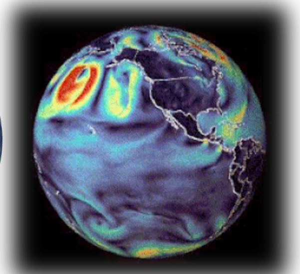
[10] J.-M. Bertrand, et al., “Technical note: Improving the European air quality forecast of the CAMS using machine learning techniques,” 2023, doi: 10.5194/acp-23-5317-2023

[11] “Air quality | Copernicus.” https://atmosphere.copernicus.eu/air-quality

[12] H. Petetin et al., “Model output statistics (MOS) applied to CAMS O3 forecasts: trade-offs between continuous and categorical skill scores,” 2022, doi: 10.5194/acp-22-11603-2022

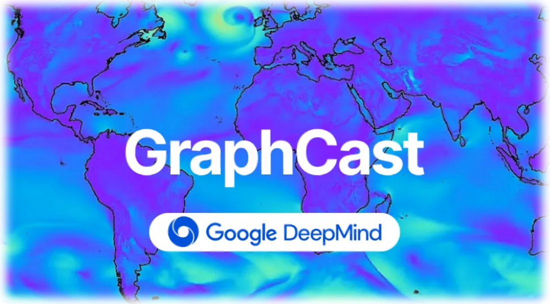
Appendix

Why Deep Learning for Time Series (TS) Forecasting ?

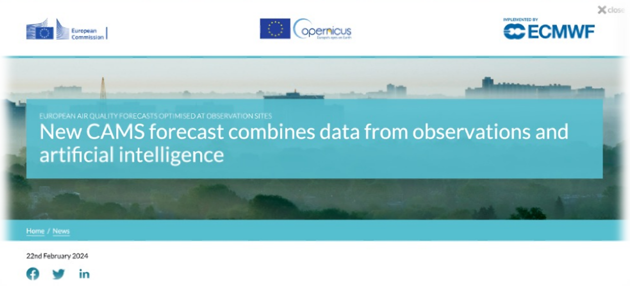


- **Learning from Data**
  - Classical Methods: *Rule-based*, rely on explicit models/physics law/assumptions
  - Deep Learning: *Data-driven*, learns patterns and relationships automatically

- **Interpretability**
  - Classical Methods: Transparent, easier to understand but may oversimplify complex systems
  - Deep Learning: Sometimes considered a "black box," challenging to interpret

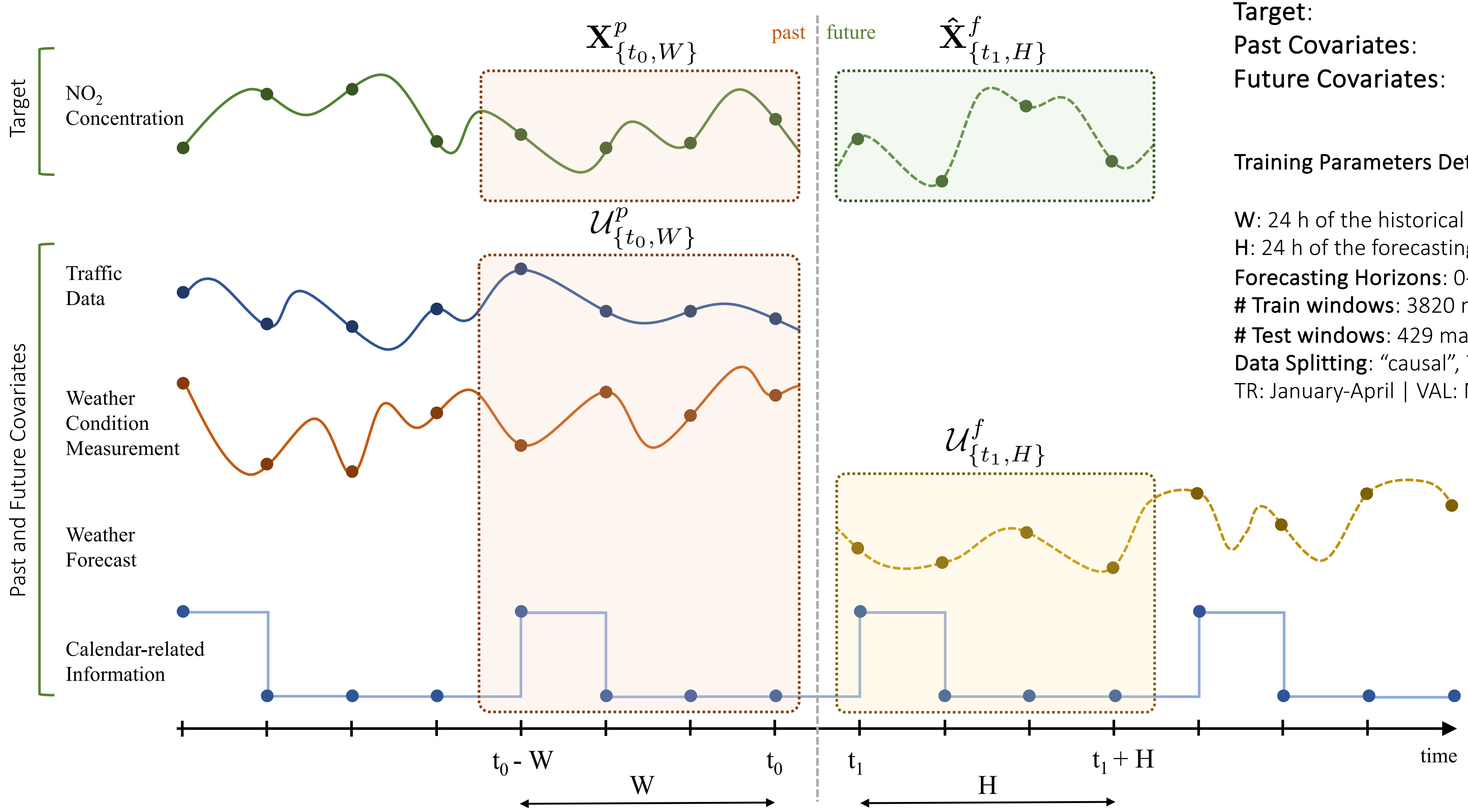


... and some big companies (like Microsoft, Google, NVIDIA, Copernicus) are investing on it [1-5]



[1] J. Pathak et al., “FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators.” arXiv, Feb. 22, 2022.
[2] R. Lam et al., “GraphCast: Learning skillful medium-range global weather forecasting.” arXiv, Aug. 04, 2023.
[3] T. Nguyen, J. Brandstetter, A. Kapoor, J. K. Gupta, and A. Grover, “ClimaX: A foundation model for weather and climate.” arXiv, Jul. 10, 2023.
[4] M. Andrychowicz *et al.*, “Deep Learning for Day Forecasts from Sparse Observations.” arXiv, Jul. 06, 2023.
[5] “New CAMS forecast combines data from observations and artificial intelligence | Copernicus.” https://atmosphere.copernicus.eu/new-cams-forecast-combines-data-observations-and-artificial-intelligence

Training Details – Local Model



Target: NO_2 concentration
Past Covariates: Traffic Data + Calendar
Future Covariates: Meteorological Data + Calendar

Training Parameters Details

W: 24 h of the historical window [*past*]
H: 24 h of the forecasting window [*future*]
Forecasting Horizons: 0-24, 25-48, 49-72 h
 # Train windows: 3820 max
 # Test windows: 429 max
Data Splitting: "causal", 70/20/10 % for TR/VAL/TS
TR: January-April | VAL: May | TS: June

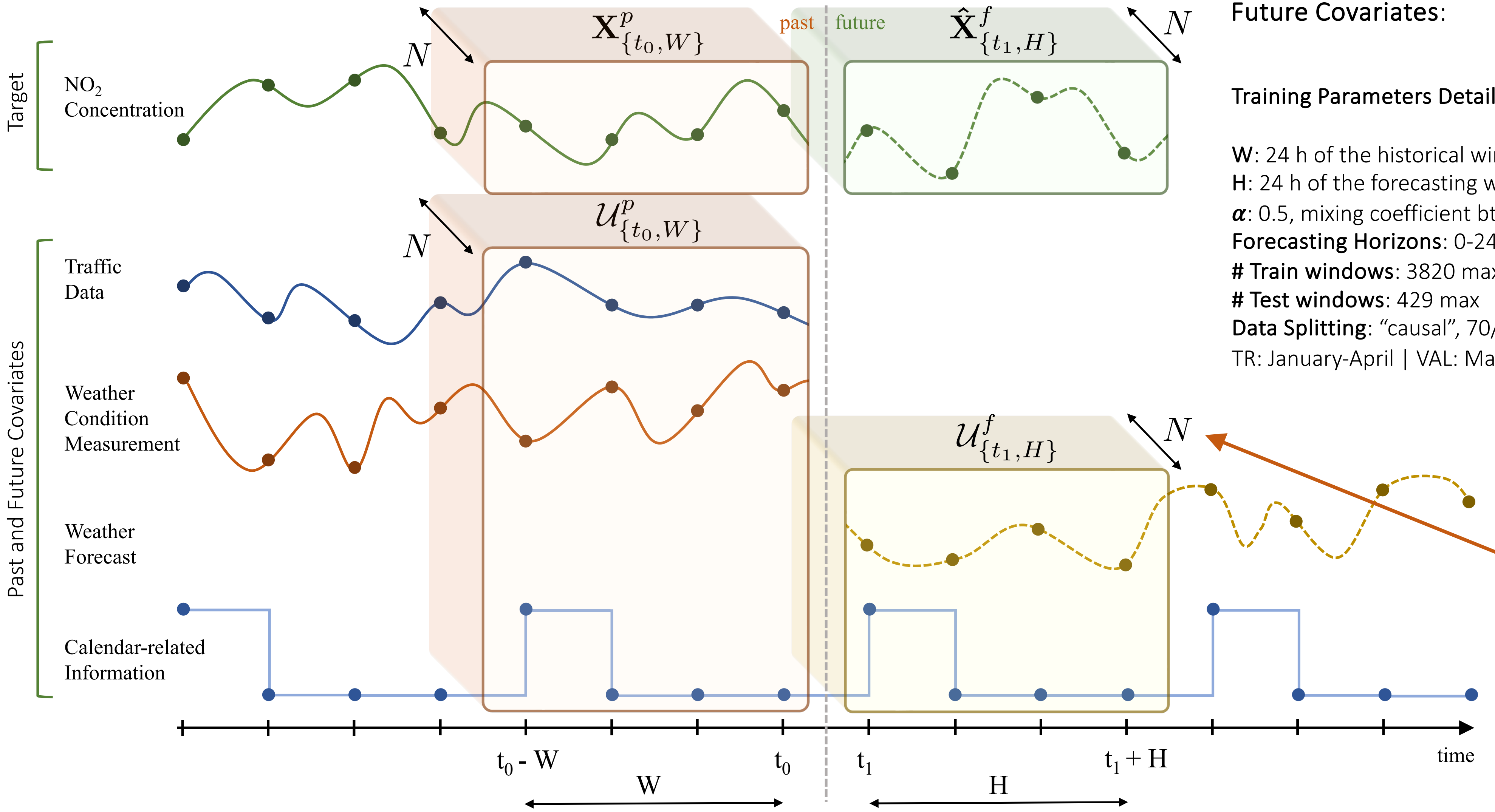
To train and assess the performance of our models we use the **M**ean **A**bsolute **E**rror (**MAE**) as

Predicted Target
True Target

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Sample Size

Training Details – Global Model



Target: NO_2 concentration
Past Covariates: Traffic Data + Calendar
Future Covariates: Meteorological Data + Calendar

Training Parameters Details

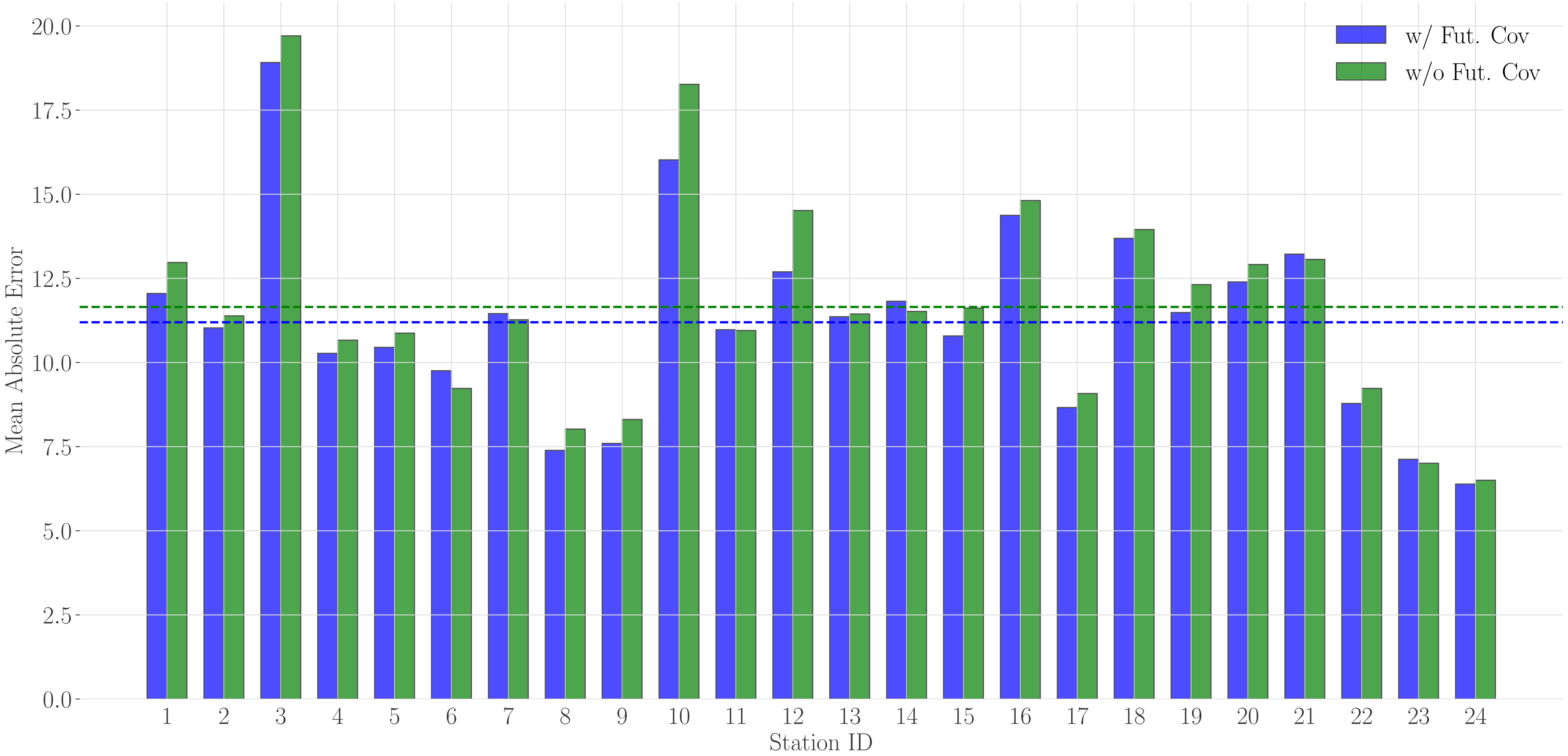
W: 24 h of the historical window [*past*]
H: 24 h of the forecasting window [*future*]
α: 0.5, mixing coefficient btw *past&future*
Forecasting Horizons: 0-24, 25-48, 49-72 h
 # Train windows: 3820 max
 # Test windows: 429 max
Data Splitting: "causal", 70/20/10 % for TR/VAL/TS
TR: January-April | VAL: May | TS: June

N time-series as the number of station in our monitoring network

Results – Local – TiDE

Future Conditioning Studies

- We compare the performance in conditioning the prediction with Past or Past&Future covariates
- **REMEMBER: we train 1 model for each monitoring station !**

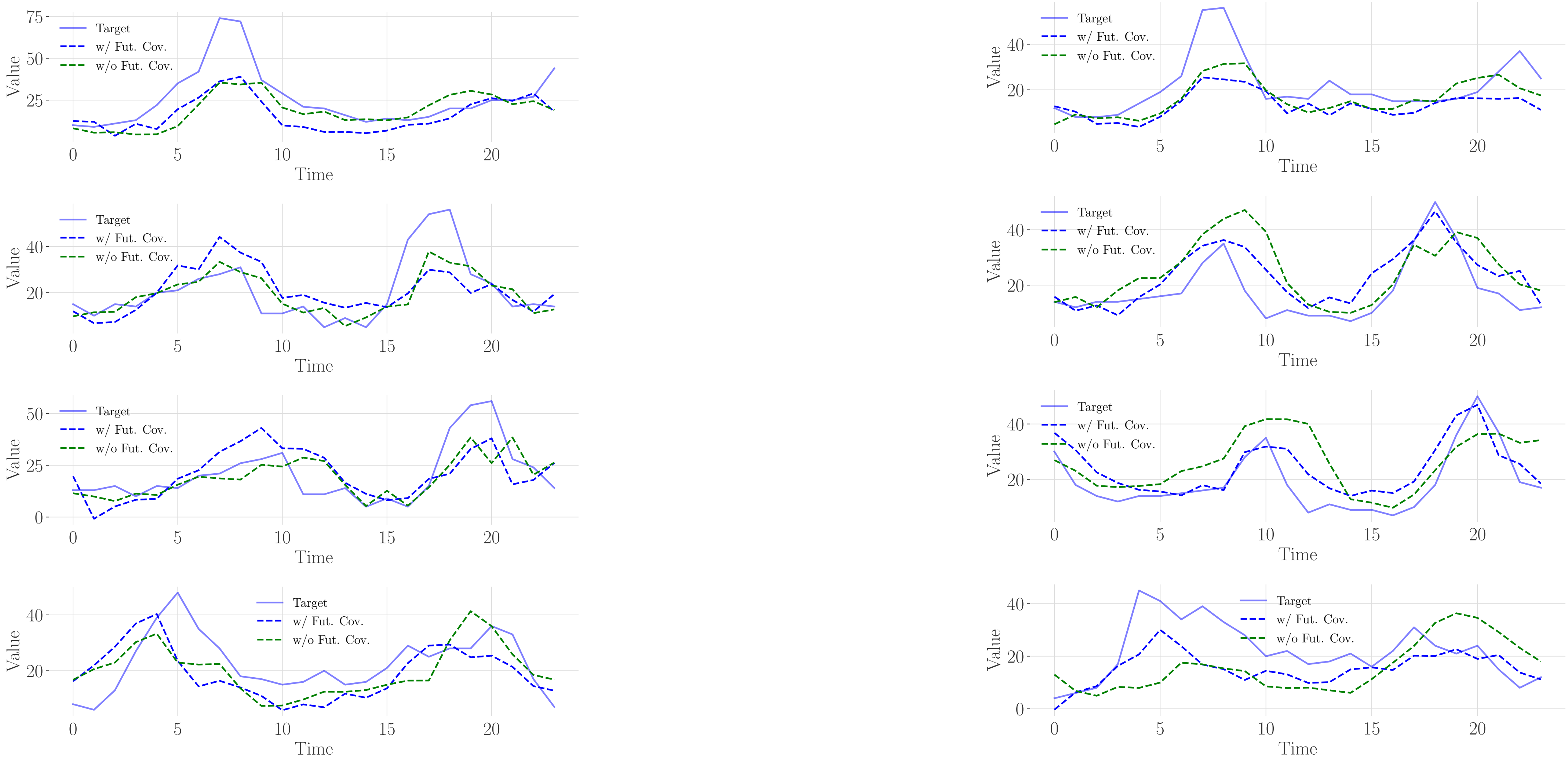


- As we expected , most of the time, conditioning the prediction with future covariates (blue bars), i.e., using the weather forecast, brings the best results in terms of forecasting accuracy

Results – Local – TiDE

Target: NO_2 concentration
Past Covariates: Traffic Data + Calendar
Future Covariates: Meteorological Data + Calendar

Future Conditioning Studies



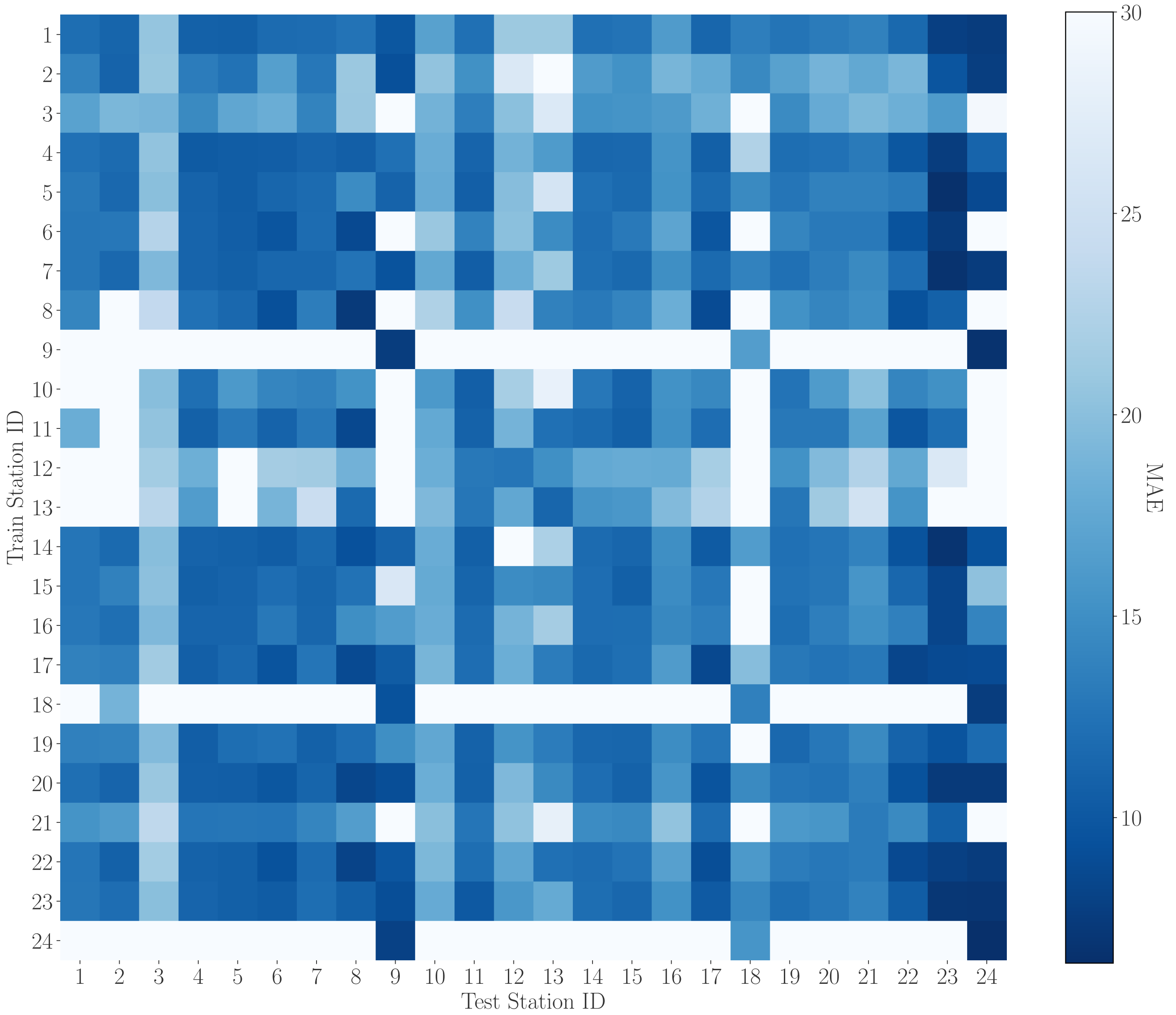
Examples of predictions from station 1 (left) and 22 (right) – thus 2 different training – referred to same time-span. The time is in hours, the value is the predicted NO_2 concentration at the station

Results – Local – TiDE

Target: NO_2 concentration
Past Covariates: Traffic Data + Calendar
Future Covariates: Meteorological Data + Calendar

Cross Station Generalization

- We asses the performance in using a model trained on a specific station, i.e., A, to predict NO_2 concentration on station B
- This could be seen as a generalization study, to spot cluster of stations that have the same prediction behavior
- As we expected, most of the time, it is not possible to use a model trained on a specific station, as a surrogate for different multiple station, since the MAE is not acceptable

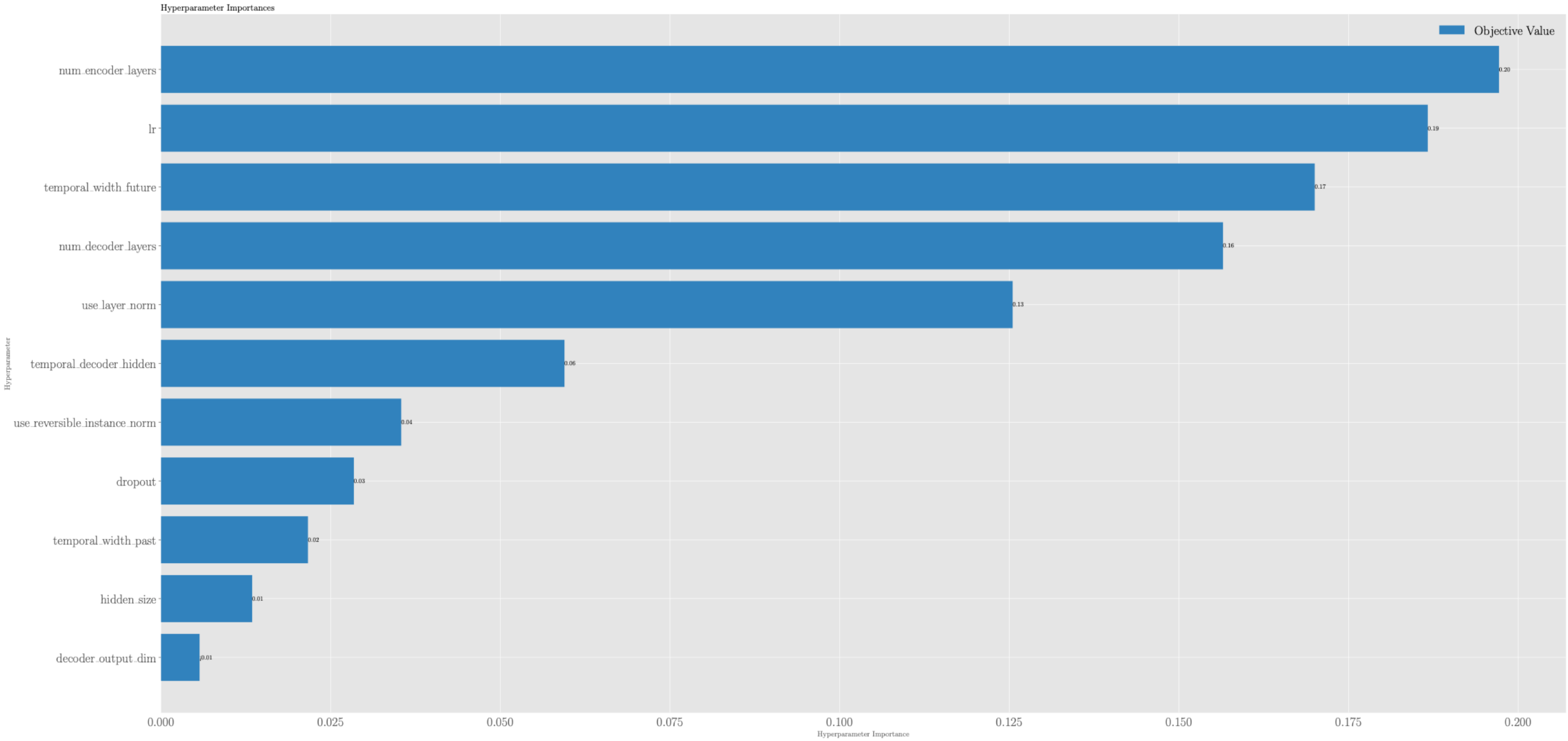


Results – Local – TiDE

Target: NO_2 concentration
Past Covariates: Traffic Data + Calendar
Future Covariates: Meteorological Data + Calendar

Hyperparameter Search

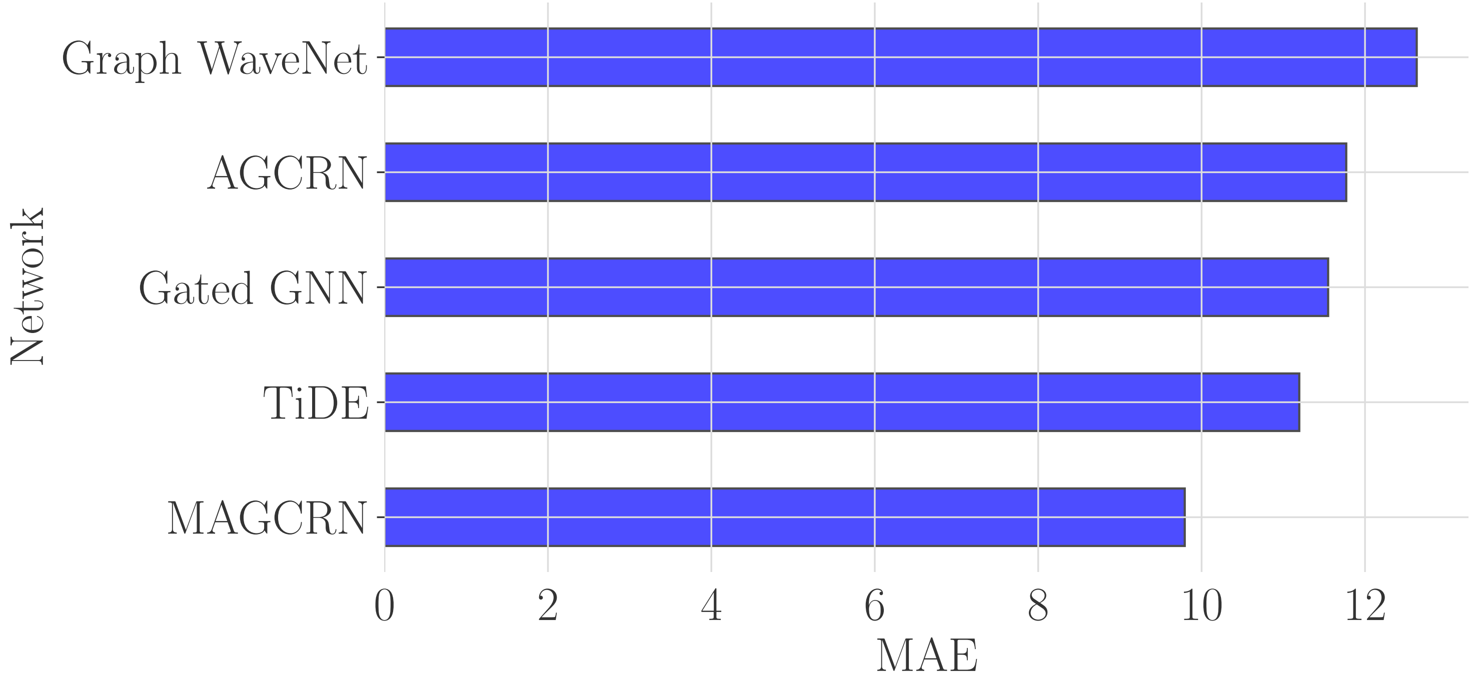
- Since TiDE is a local model, it is preferable to carry out study for each station (model), or a subset of them
- For the sake of brevity, here we only report the parameter importance study, but we have plenty of results, from convergence study, to objective value rank, and so on



Target: NO_2 concentration
Past Covariates: Traffic Data + Calendar
Future Covariates: Meteorological Data + Calendar

Network Comparison

- We compare the performance of multiple GNN-based (*global*) forecasting architecture and the previous TiDE (*local*) model
- The **MAE** value is the **mean value among all the stations** of the observation network



- As we can see, our MAGCRN outperforms all the selected state-of-the-art DL-based forecasting models

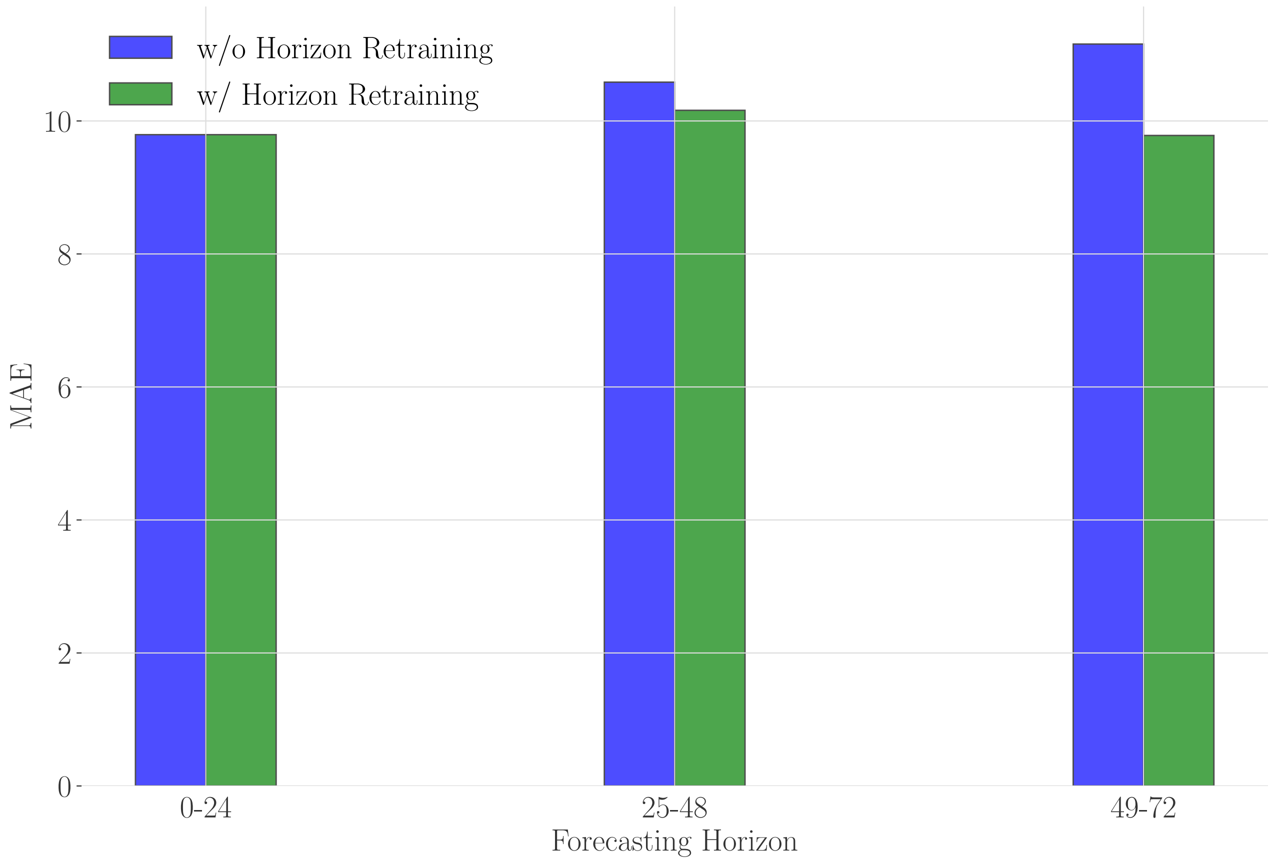
Results – Global – Graph Neural Network

Forecasting Horizons Effectiveness

- Until now, we considered the performance for only one forecasting horizon, i.e., predicting the 0-24h (1-day ahead)
- Here we report the results of the MAGCRN in forecasting NO_2 1-day (**0-24h**), 2-day (**25-48h**) and 3-day (**49-72h**) ahead

- In blue we report the MAE of the MAGCRN trained in predicting 1-day ahead NO_2 by using the data referred to previous day (1-day before). Then, with the same training we infer the 2 and 3-day ahead NO_2
- We can notice that more distant forecasting horizons bring to an appreciable decrease in performance

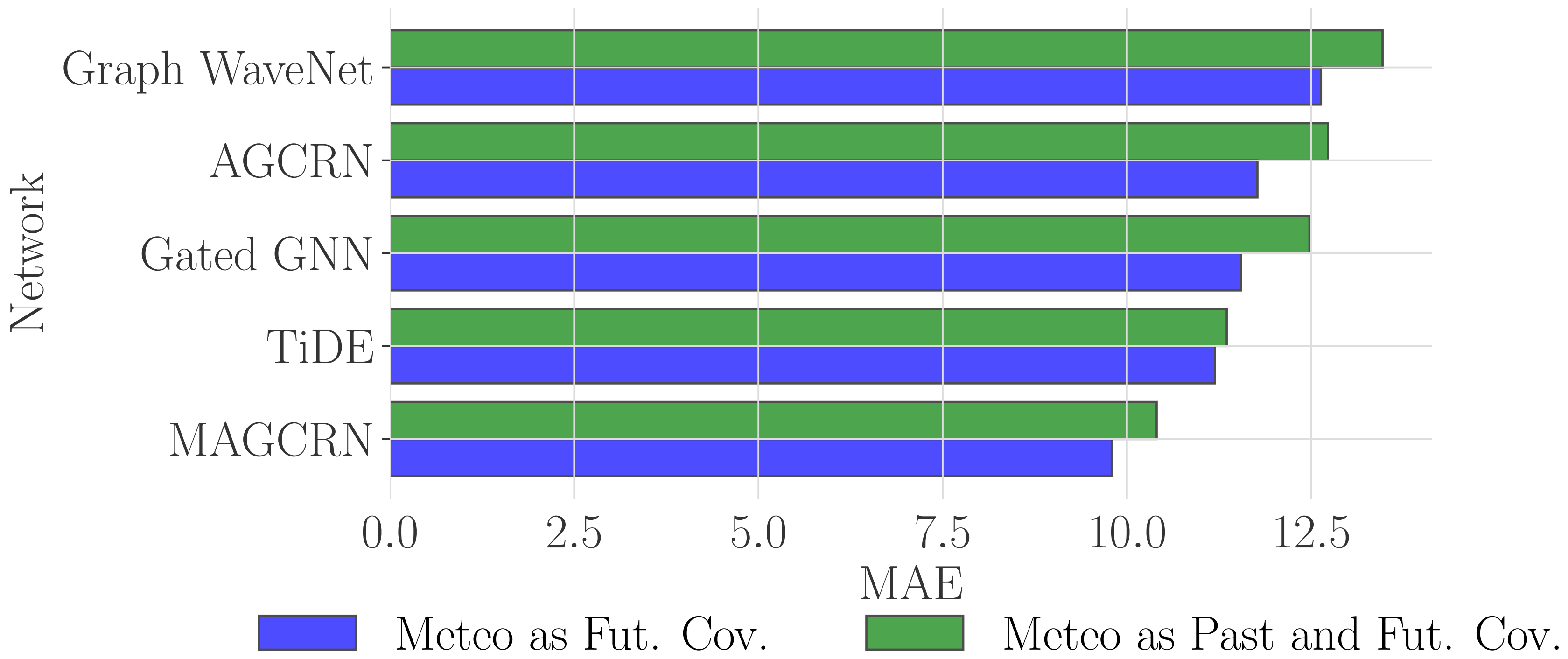
- In green we report the MAE of **3 different** MAGCRN trainings, each one specialized in predicting (from left to right) 1, 2 or 3 day ahead values respectively, by using the values referred only to previous day (1-day before).
- As we expected, having 3 different models, one for each forecasting horizon, benefits the performance (near-equal MAE among all the horizons), at the expense of higher computational cost (3 different trainings)



Results – Global – Graph Neural Network

Complexity Consideration

- For all the experiments, we considered the **meteorological data only as a future covariates** (blue bars)



- Using the meteorological variables also as a past covariates (green bars) compromises the performance of all the considered models
- We believe that the models' ability to consider additional information is limited, and more information does not always lead to a better result

Results – Model's Resource Consumption

Training Time

- TiDE model (local): GPU: ≈ 10 min. * CPU: ≈ 30 min. *
- MAGCRN model (global): GPU: ≈ 15 min. CPU: ≈ 3 h

* for single station

Memory Consumption

- TiDE model (local): ≈ 1 GiB *
- MAGCRN model (global): ≈ 8 GiB

Reference Workstation Specification

CPU: Intel Xeon Gold 6246, 48 Cores @ 3.3 GHz
GPU: Titan RTX (4608 CUDAs @ 1350MHz, 24 GiB)
RAM: 252 GiB

Results – Global – Graph Neural Network

Target: NO_2 concentration
Past Covariates: Traffic Data + Calendar
Future Covariates: Meteorological Data + Calendar

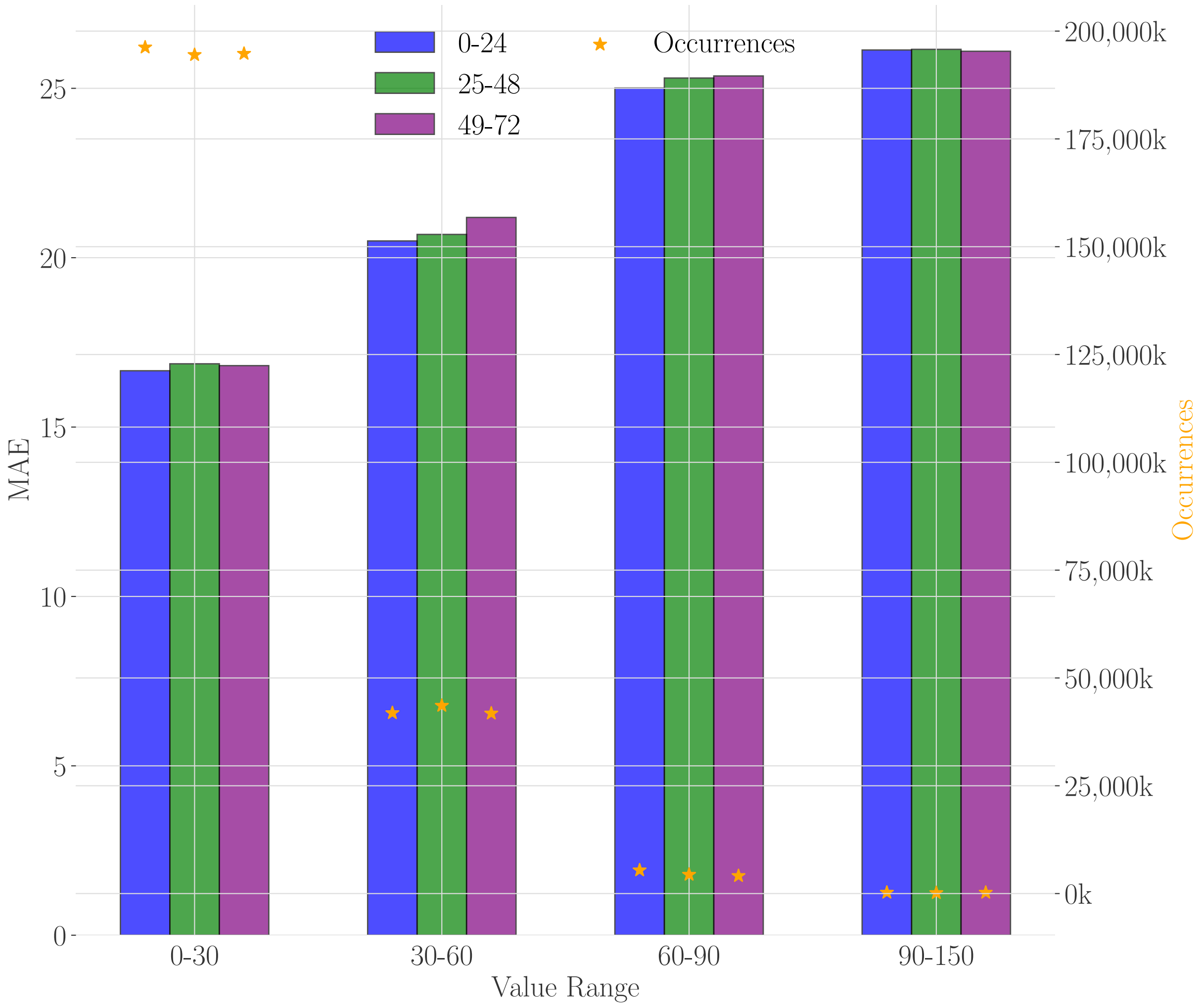
Range Effectiveness

- We study the effectiveness of our MAGCRN on **4** non-overlapping contiguous NO_2 concentration **ranges**

- The blue, green and violet bars denotes the **3** different **forecasting horizons**

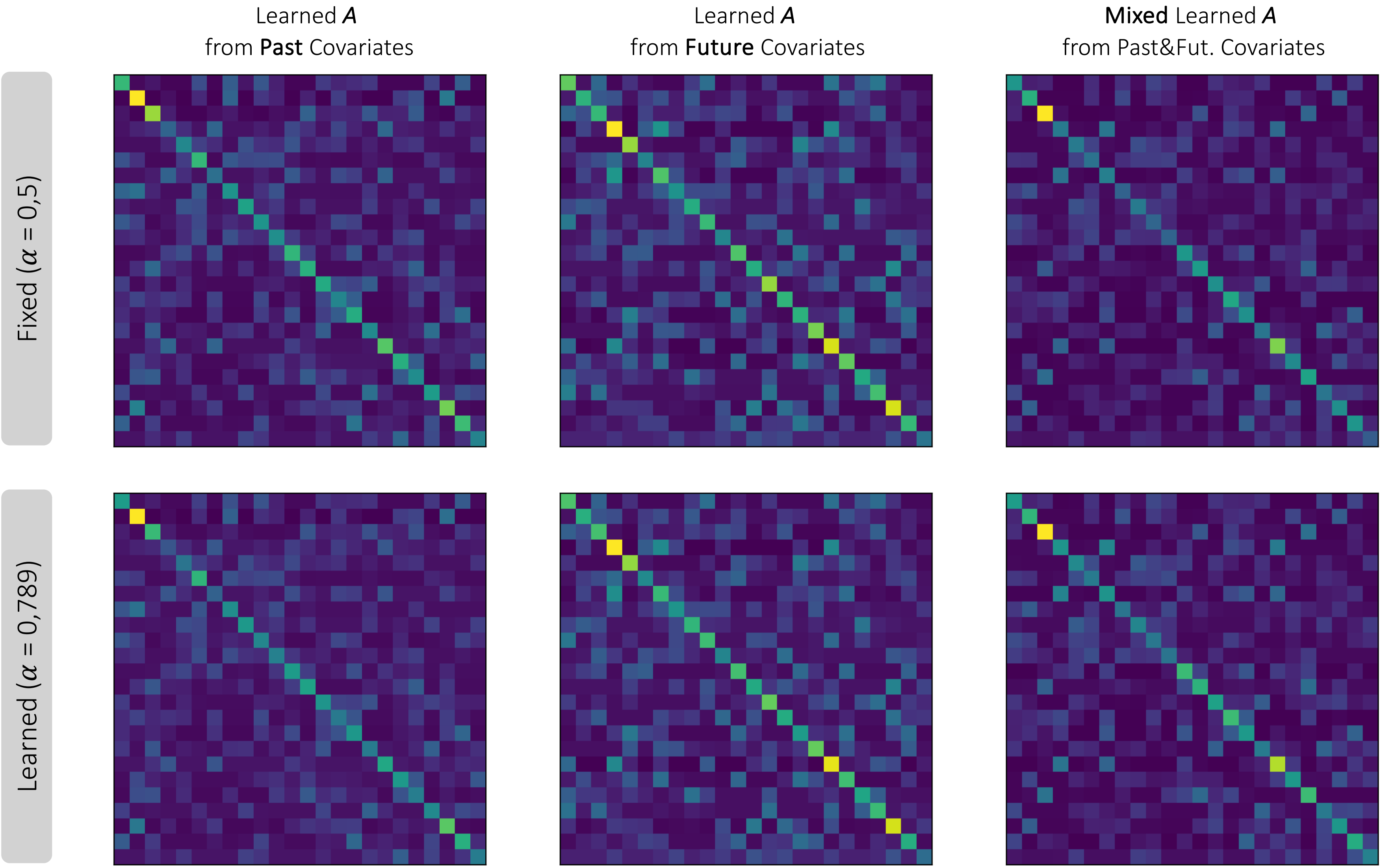
- The stars denotes the number of values (occurrences) within the considered range, and are different for each forecasting horizon, i.e., higher values are only a few

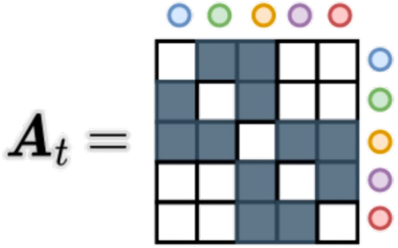
- We can observe a performance **drop** when the NO_2 concentration values are **high**, since we do not have sufficient training samples of this dynamic, i.e., as you can see from the number of occurrences

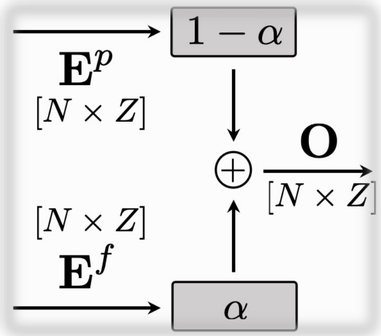


Results – Global – Graph Neural Network

Learning α







Training Details – Design Choice

Target: NO_2 concentration
Past Covariates: Traffic Data + Calendar
Future Covariates: Meteorological Data + Calendar

Window Pairs Construction

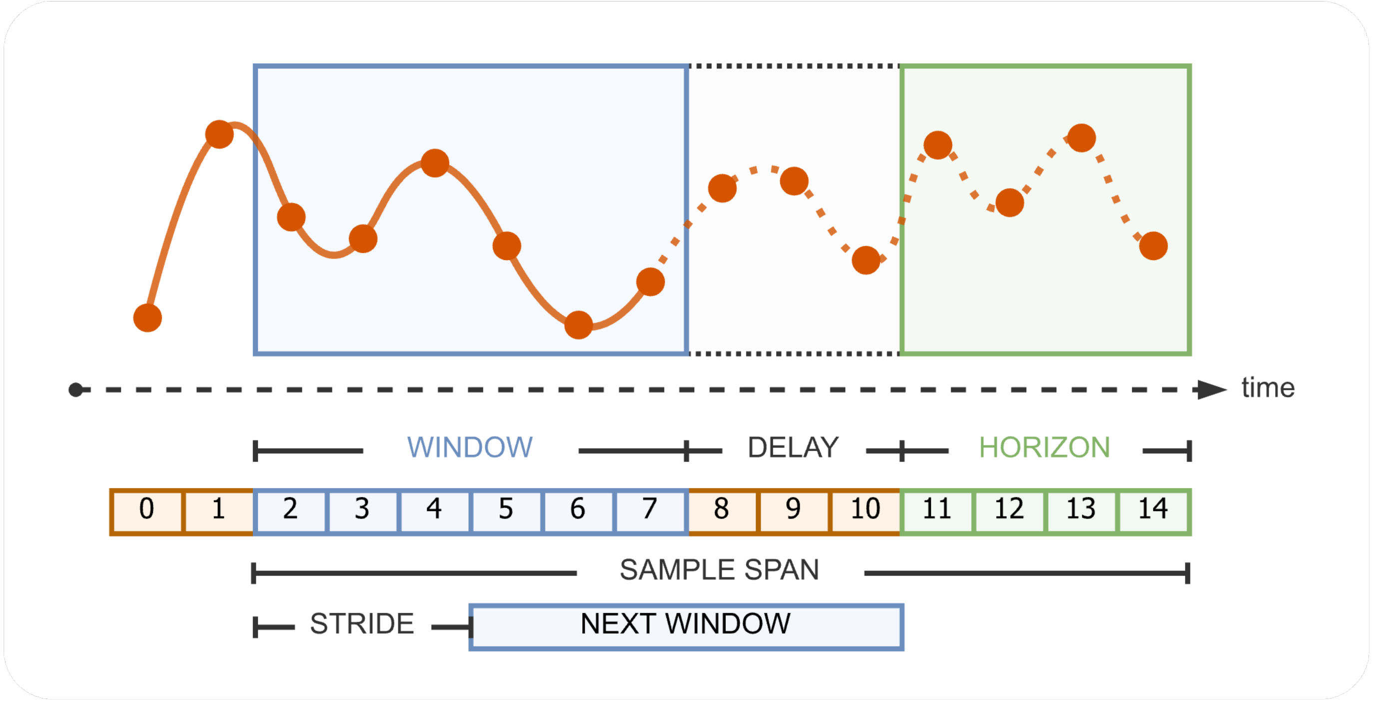


Image courtesy of Torch Spatiotemporal https://github.com/TorchSpatiotemporal/tsl