# Taming Local Effects in Graph-based Spatiotemporal Forecasting

Andrea Cini*[1] Ivan Marisca*[1] Daniele Zambon[1] Cesare Alippi[12]

[1]The Swiss AI Lab IDSIA USI-SUPSI, Università della Svizzera italiana (CH) [2]Politecnico di Milano (IT) *Equal contribution

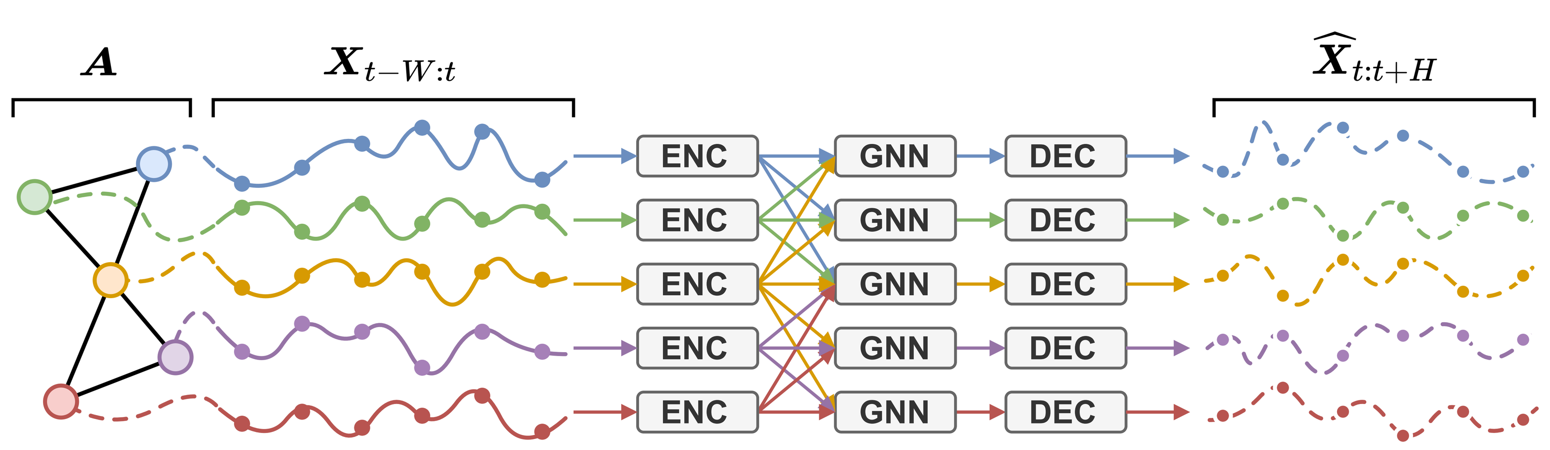
PAPER CODE

## Motivation

Spatiotemporal GNNs (**STGNNs**) are effective in forecasting **collections of time series**, but **fully global** models often underperform.

- Global models **share parameters among time series**.
- While very flexible, they might fail in accounting for the **specific dynamics of each time series.**
- Understanding the **interplay between global and local components** in graph-based predictors is critical.
- Guidelines to design **effective hybrid global-local STGNNs**, amenable to **transfer learning**, are missing.

## Graph-based Forecasting




We consider **graph-based forecasting models** composed of 3 blocks:

**ENCODER** encodes input features (e.g., with an MLP).

**(ST)GNN** propagates information along time and through the graph.

**DECODER** generates forecasts from obtained node representations.

### The Issue

Models of this kind are **global**, i.e., all parameters are shared among the time series.

| | |
|---|---|
| ☺ Fewer parameters (in total) | ☹ Struggle to model local effects |
| ☺ More data available for training | ☹ Require large model capacity... |
| ☺ Work in inductive settings | ☹ ...or impractically long windows |

Make STGNNs global-local with specialized local components!

## Main Findings

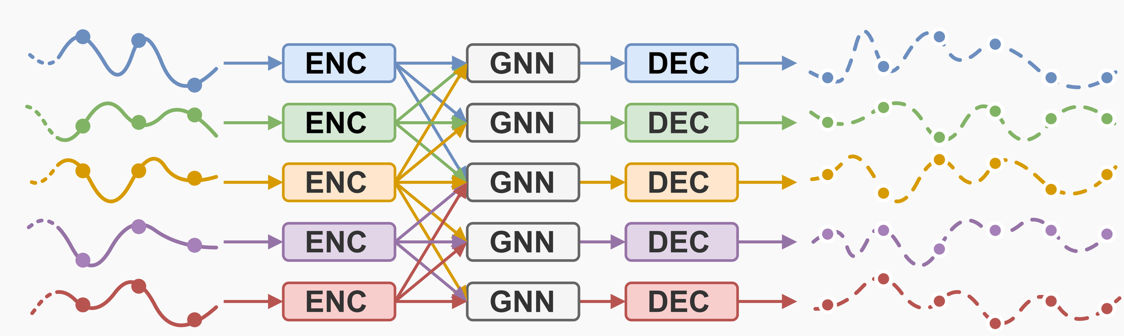
- **Local components** can be crucial for accurate predictions.
- **Node embeddings** can amortize the learning of local components.
- **Global-local** STGNNs capture local effects with contained model complexity and smaller input windows w.r.t. global approaches.
- **Node embeddings** facilitate transferability of global-local models.
- **Structuring** the embedding space is an effective regularization.

## Taming local effects in STGNNs

We study the impact of introducing local components in global STGNNs.

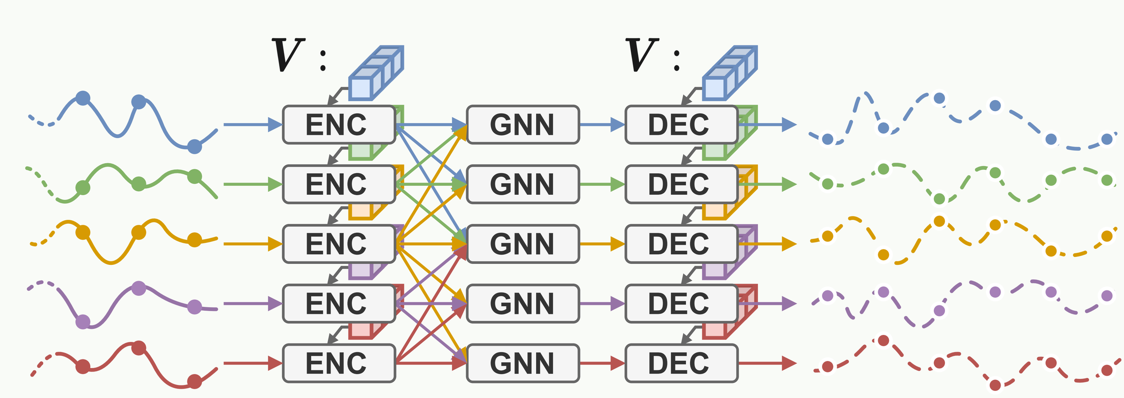
Option 1
### Local blocks
(with node-specific weights)

☺ Can deal with local effects
☹ Flexibility is compromised
☹ Model complexity increases

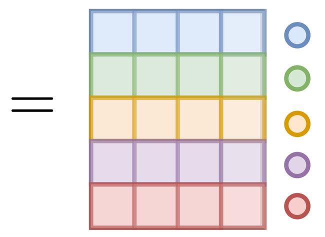



Option 2
### Node embeddings

☺ Amortize the cost of learning local components
☺ Ease transfer learning




Node embeddings are a table of learnable parameter vectors, each associated with a node.

$V =$ 

## Structuring the Embedding Space

Regularizing the embedding space improves transferability and interpretability.

- Variational: the smoother embedding space enables interpolation.

$$\mathcal{L}_t^{\mathrm{r}} \doteq \mathbb{E}_{\boldsymbol{Q}\sim \boldsymbol{M}}\left[\left\|\widehat{\boldsymbol{X}}_{t:t+H} - \boldsymbol{X}_{t:t+H}\right\|_2^2\right] + \beta D_{\mathrm{KL}}(M|\mathcal{N}(\mathbf{0},\mathbb{I}))$$

- Clustering: clusters in the latent space improve interpretability.

$$\mathcal{L}_{reg} \doteq \mathbb{E}_{\boldsymbol{M}}\left[\|\boldsymbol{V} - \boldsymbol{M}\boldsymbol{C}\|_2\right], \quad p(\boldsymbol{M}_{ij} = 1) = \frac{e^{\boldsymbol{S}_{ij}/\tau}}{\sum e^{\boldsymbol{S}_{ik}/\tau}}$$

## Some Empirical Results

| Models | MetrLA | PemsBAY | CER-E | AQI | MetrLA | PemsBAY | CER-E | AQI |
|---|---|---|---|---|---|---|---|---|
| Reference arch. | Global models | | | | + Embeddings | | | |
| RNN | 3.54±.00 | 1.77±.00 | 456.98±0.61 | 14.02±.04 | **3.15**±.03 | **1.59**±.00 | **421.50**±1.78 | **13.73**±.04 |
| T&S-IMP | 3.35±.01 | 1.70±.01 | 443.85±0.99 | 12.87±.02 | **3.10**±.01 | **1.59**±.00 | **417.71**±1.28 | **12.48**±.03 |
| TTS-IMP | 3.34±.01 | 1.72±.00 | 439.13±0.51 | 12.74±.02 | **3.08**±.01 | **1.58**±.00 | **412.44**±2.80 | **12.33**±.02 |
| T&S-AMP | 3.22±.02 | 1.65±.00 | N/A | N/A | **3.07**±.02 | **1.59**±.00 | N/A | N/A |
| TTS-AMP | 3.24±.01 | 1.66±.00 | 431.33±0.68 | 12.30±.02 | **3.06**±.01 | **1.58**±.01 | **412.95**±1.28 | **12.15**±.02 |
| Baseline arch. | Original | | | | + Embeddings | | | |
| DCRNN | 3.22±.01 | 1.64±.00 | 428.36±1.23 | 12.96±.03 | **3.07**±.02 | **1.60**±.00 | **412.87**±1.51 | **12.53**±.02 |
| GraphWaveNet | 3.05±.03 | **1.56**±.01 | **397.17**±0.67 | 12.08±.11 | **2.99**±.02 | 1.58±.00 | 401.15±1.49 | **11.81**±.04 |
| AGCRN | 3.16±.01 | **1.61**±.00 | 444.80±1.25 | 13.33±.02 | **3.14**±.00 | 1.62±.00 | **436.84**±2.06 | **13.28**±.03 |

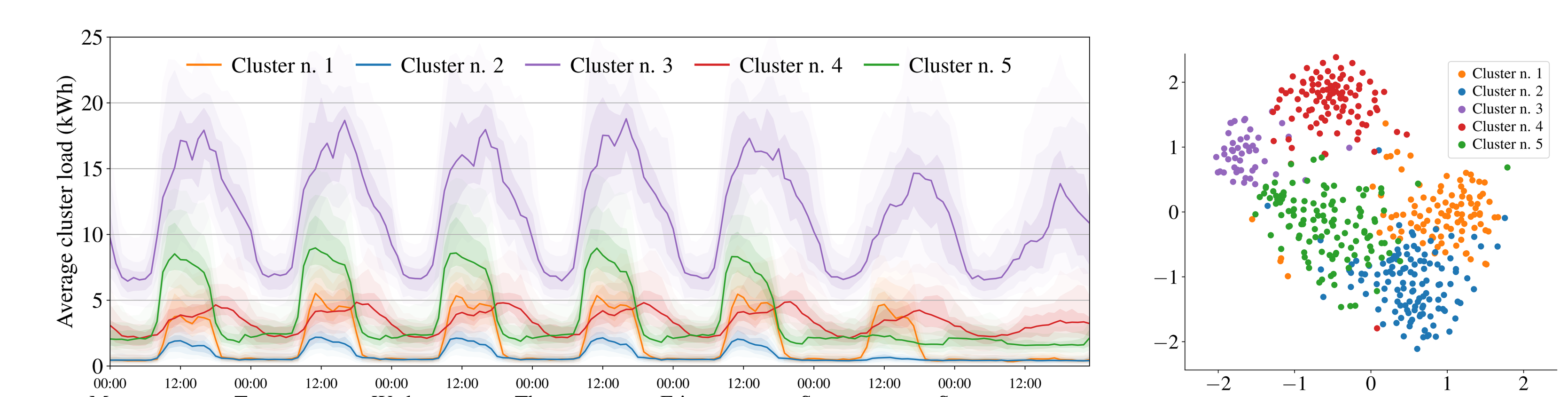
Table 1. Performance (MAE) on benchmark datasets.

## Transfer Learning

| | TTS-IMP | PEMS03 | PEMS04 | PEMS07 | PEMS08 |
|---|---|---|---|---|---|
| Fine-tuning | Global | 15.30±.03 | 21.59±.11 | 23.82±.03 | 15.90±.07 |
| | Embeddings | 14.64±.05 | 20.27±.11 | **22.23**±.08 | **15.45**±.06 |
| | – Variational | **14.56**±.03 | 20.19±.05 | 22.43±.02 | **15.41**±.06 |
| | – Clustering | **14.60**±.02 | **19.91**±.11 | **22.16**±.07 | **15.41**±.06 |
| | Zero-shot | 18.20±.09 | 23.88±.08 | 32.76±.69 | 20.41±.07 |

Table 2. Performance (MAE) on traffic datasets.

## Clusterized embedding space (CER dataset)




(a) Average cluster load by day of the week. (b) Node embeddings.


**Andrea**


**Ivan**


**Daniele**


**Cesare**

Check out our library for STGNNs!
TorchSpatiotemporal/tsl



andrea.cini@usi.ch · ivan.marisca@usi.ch · daniele.zambon@usi.ch gmlg.ch @GMLG_Lugano