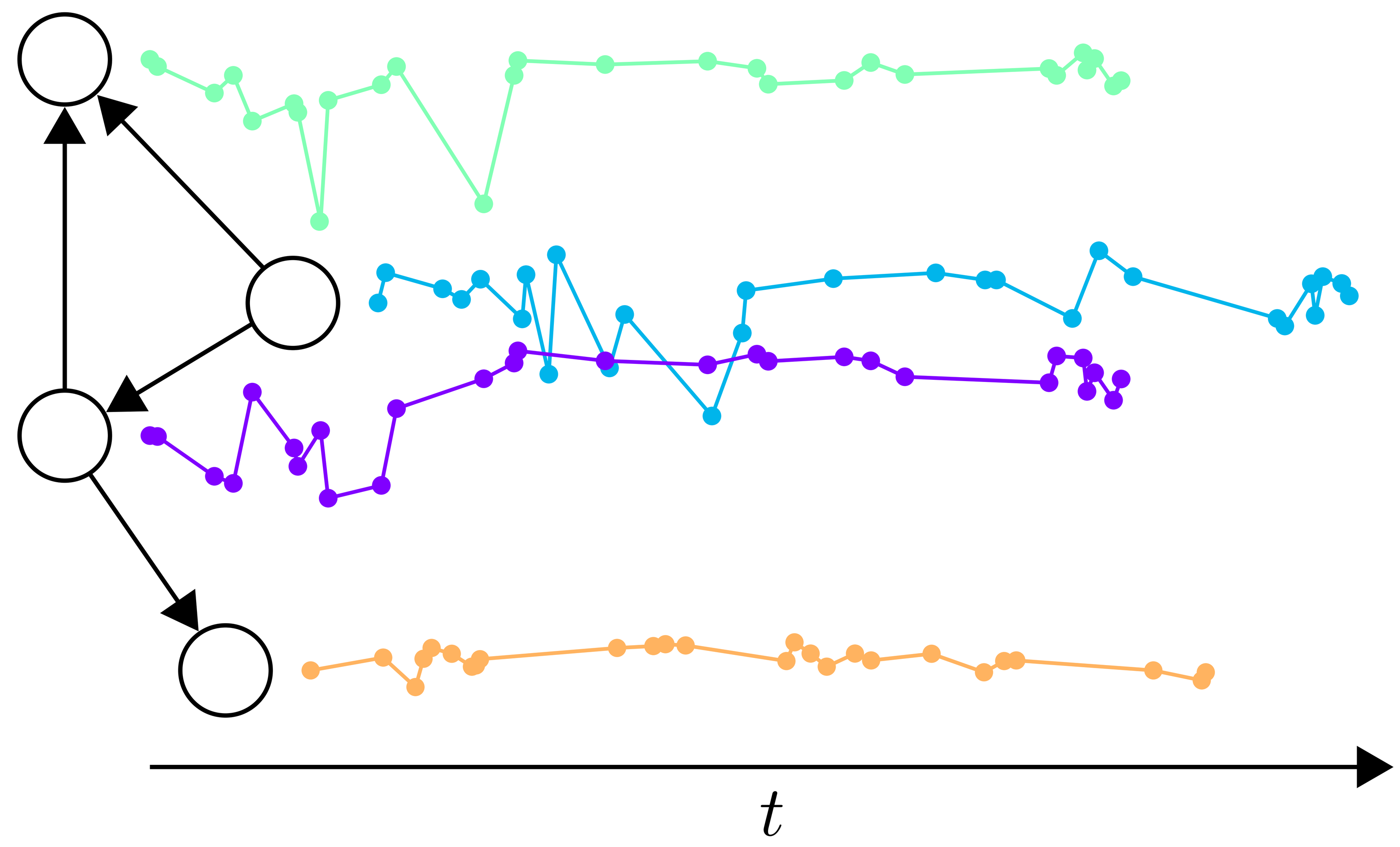
# Temporal Graph Neural Networks for Irregular Data

Joel Oskarsson[1] Per Sidén[1 2] Fredrik Lindsten[1]

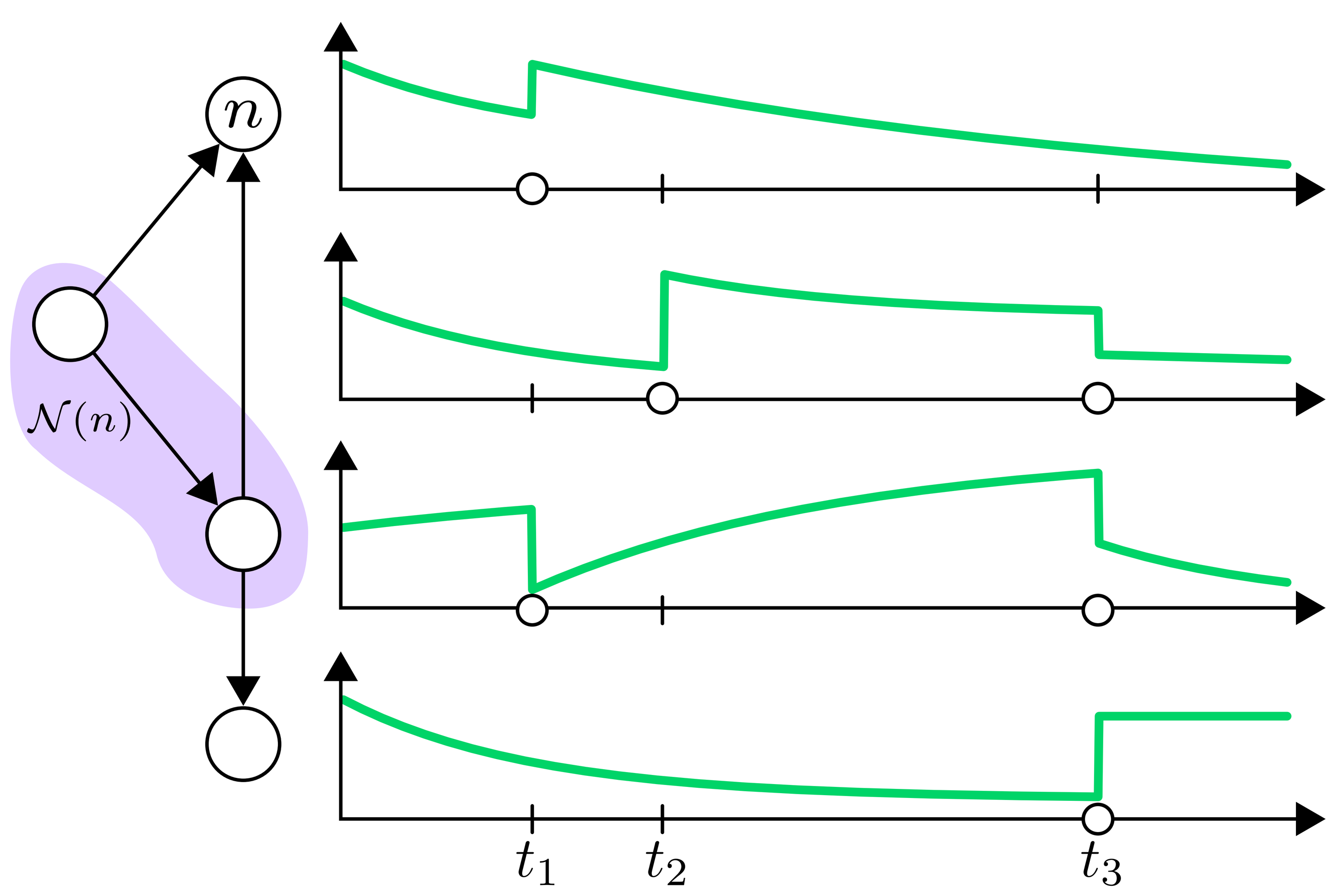
[1]Division of Statistics and Machine Learning, Department of Computer and Information Science, Linköping University, Linköping, Sweden
[2]Arriver Software AB




## Irregular Graph-Structured Time Series

- Consider **forecasting** for **multiple time series** related by some known **graph** structure
- How can **irregular observations** be dealt with?
  - Irregularly spaced observation times
  - Only a subset of nodes observed at each time point

## Time-Continuous Latent States

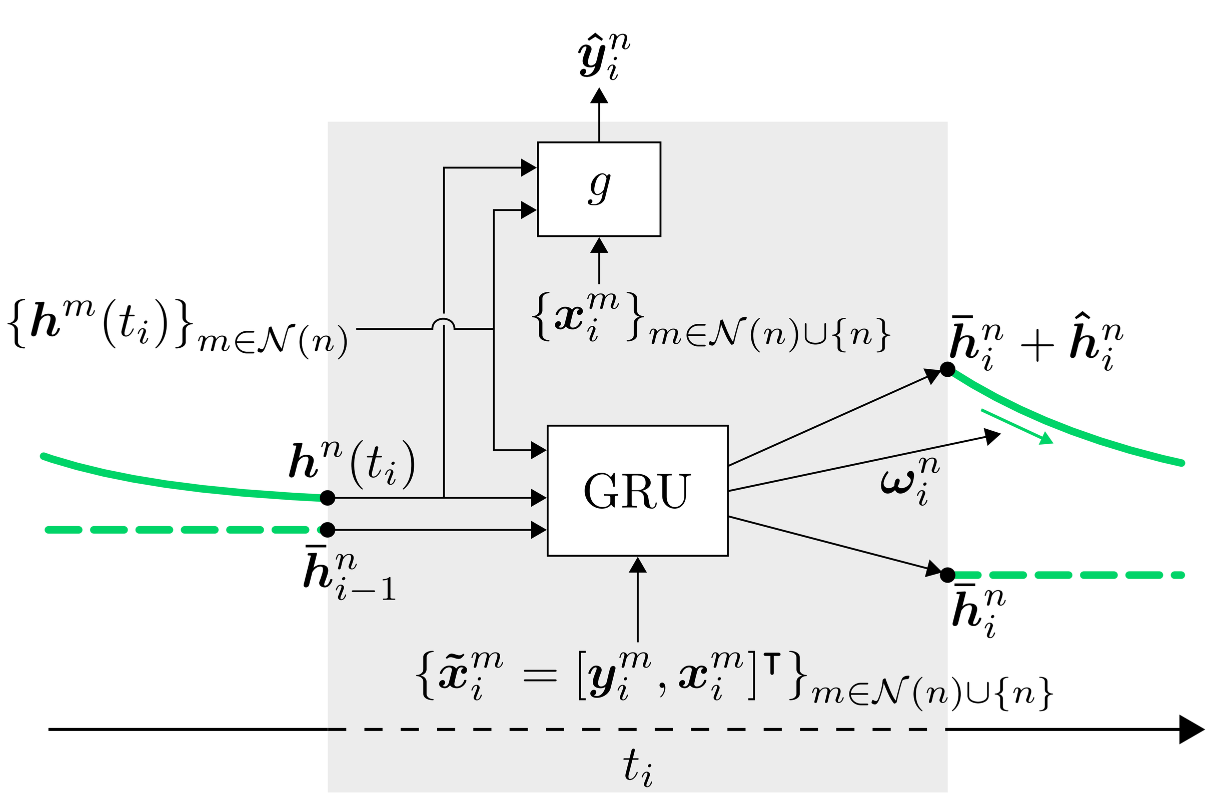



- At each node $n$ a latent state $\boldsymbol{h}^n(t) = \bar{\boldsymbol{h}}^n_i + \tilde{\boldsymbol{h}}^n(t)$ evolves over **continuous time**
- Between node observations the dynamics of $\tilde{\boldsymbol{h}}^n(t)$ are dictated by the **linear ODE**

$$d\tilde{\boldsymbol{h}}^n(t) = A\tilde{\boldsymbol{h}}^n(t)\, dt, \qquad \tilde{\boldsymbol{h}}^n(t_i) = \hat{\boldsymbol{h}}^n_i$$

  - Solution dynamics: **exponential decay** + optional **periodic** component

- When node $n$ is observed we perform a **GRU**-like update
- Observation $\boldsymbol{y}^n_i$
- Input features $\boldsymbol{x}^n_i$
- Vector $\boldsymbol{\omega}^n_i$ defines the ODE for the next time interval!

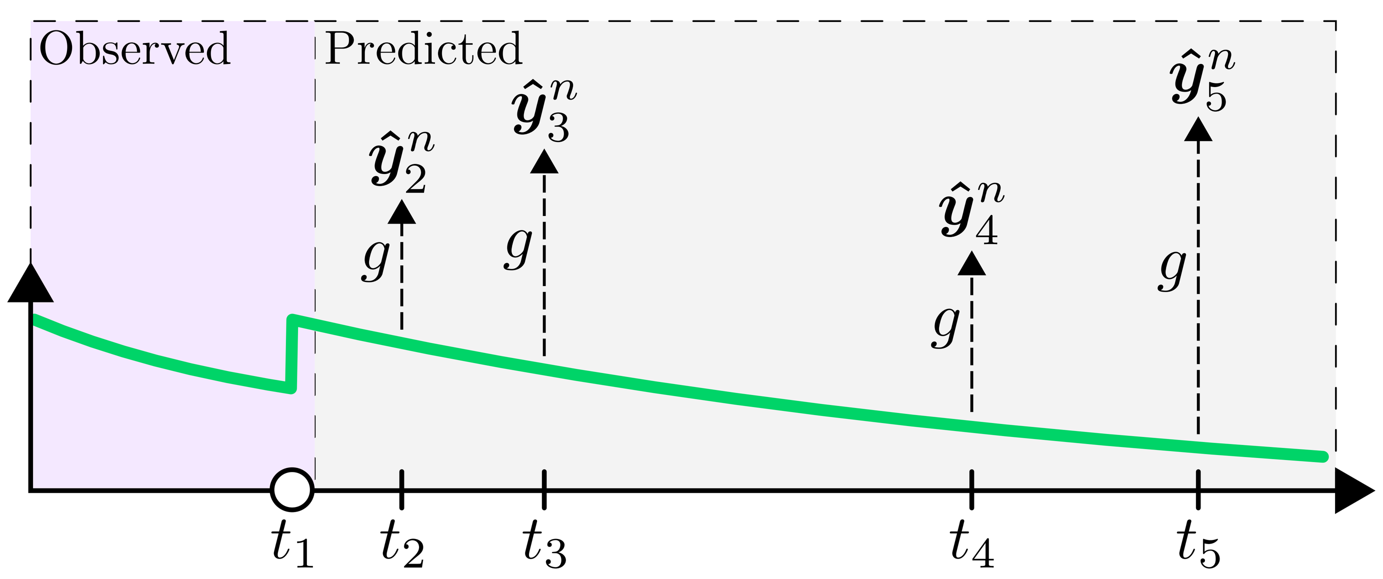



$$\hat{\boldsymbol{h}}^n_i, \bar{\boldsymbol{h}}^n_i, \boldsymbol{\omega}^n_i = \text{GRU}(\boldsymbol{h}^n(t_i), \bar{\boldsymbol{h}}^n_{i-1}, \boldsymbol{x}^n_i, \boldsymbol{y}^n_i)$$

- GRU cell extended with **Graph Neural Network (GNN)** components to capture dependencies in the graph neighborhood $\mathcal{N}(n)$

## Making Predictions

- Forecasting for **arbitrary future time points** by applying predictive model $g$ to unrolled latent states!




- GNN as predictive model $g$

$$\hat{\boldsymbol{y}}^n_j = \text{GNN}^g\left([\boldsymbol{h}^n(t_j), \boldsymbol{x}^n_j], \{[\boldsymbol{h}^m(t_j), \boldsymbol{x}^m_j]\}_{m\in\mathcal{N}(n)}\right)$$

- Training with **custom loss function** $\mathcal{L}_{\text{MSE}}$: MSE with time-continuous weighting function

## Experiments with our TGNN4I Model

- We experiment with versions of our TGNN4I model with **static**, **exponential** and **periodic** dynamics
- For non-graph baselines we consider versions that model:
  - All nodes jointly as single high-dimensional time series (joint)
  - The time series in each node independently (node)
- Custom loss $\mathcal{L}_{\text{MSE}}$ used also as evaluation metric

### 🚗 METR-LA Traffic Data

- Highway network graph with 207 nodes
- Traffic speed forecasting
- Subsampled to 25%–100% node observations

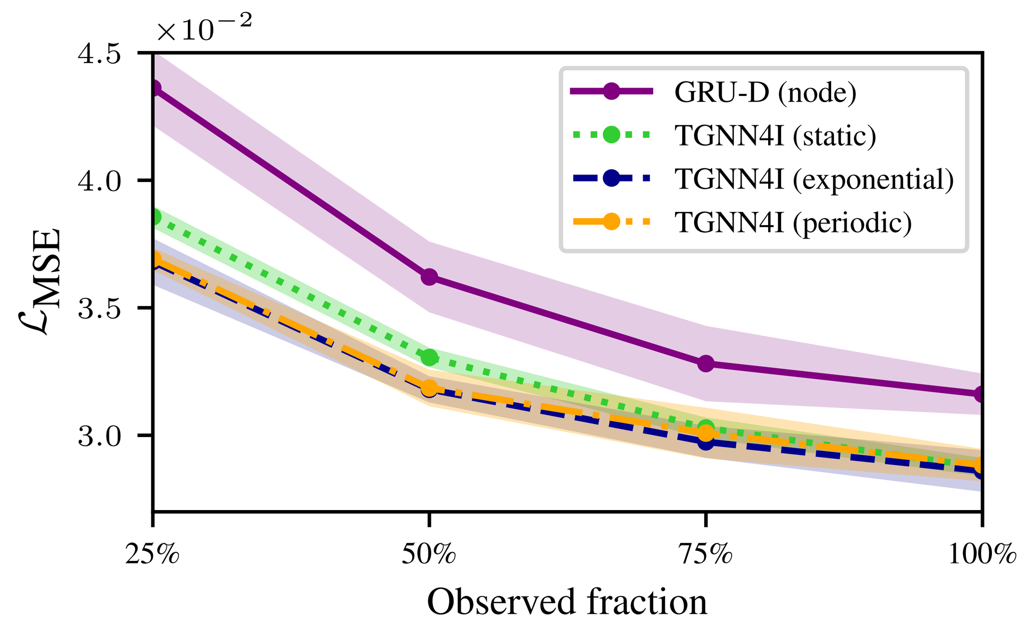



Figure: $\mathcal{L}_{\text{MSE}}$ on METR-LA with 25%–100% observations

### USHCN Climate Data

- Spatial graph of 1123 nodes
- Daily temperature data

### Synthetic Periodic Data

- Random DAG with 20 nodes
- Periodic signals with graph dependencies

Table: $\mathcal{L}_{\text{MSE}}$ $(\times10^2)$ on USHCN and Periodic data

| | USHCN | Periodic |
|---|---|---|
| Predict Previous | 16.88 | 27.52 |
| GRU-D [1] (joint) | 8.03±0.23 | – |
| GRU-D [1] (node) | 13.12±0.03 | 3.13±0.06 |
| Transformer (joint) | 7.36±0.41 | 23.19±0.38 |
| Transformer (node) | 15.68±0.32 | 15.39±0.05 |
| LG-ODE [2] | - | 16.61±0.23 |
| TGNN4I (static) | 6.97±0.05 | 15.12±0.05 |
| TGNN4I (exponential) | **6.72±0.04** | 2.91±0.17 |
| TGNN4I (periodic) | **6.72±0.05** | **1.95±0.11** |

## More Information

Paper

Code

**Code, Link to Paper:**
`github.com/joeloskarsson/tgnn4i`

**Correspondence to:**
`joel.oskarsson@liu.se`

**References**

[1] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, 2018.

[2] Z. Huang, Y. Sun, and W. Wang. Learning continuous system dynamics from irregularly-sampled partial observations. In *Advances in Neural Information Processing Systems*, volume 33, 2020.