Sparse Graph Learning from Spatiotemporal Time Series

Andrea Cini[1] Daniele Zambon[1] Cesare Alippi[1,2]

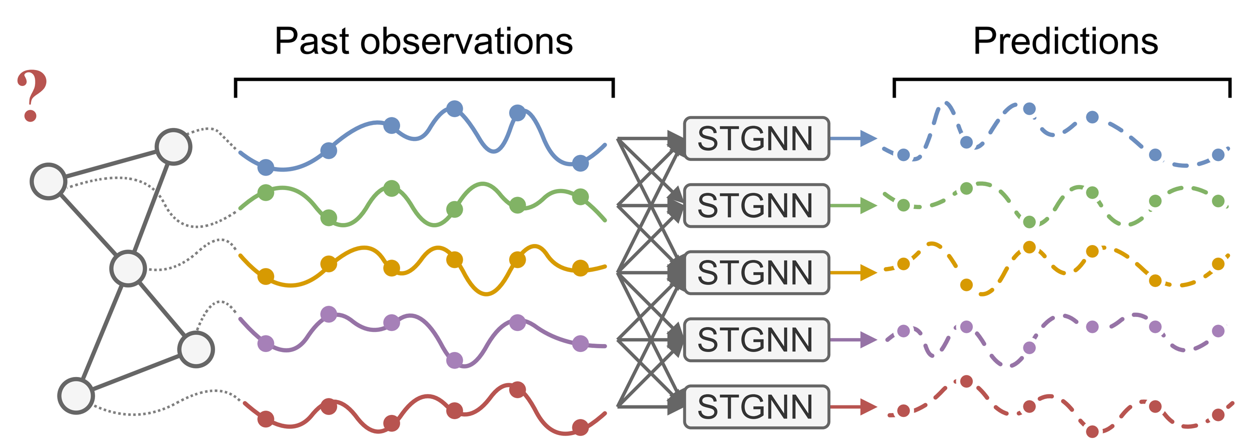
[1]The Swiss AI Lab IDSIA USI-SUPSI, Università della Svizzera italiana (CH) [2]Politecnico di Milano (IT)

Motivation

Spatiotemporal GNNs (STGNNs) are effective in forecasting **time series collections**, however, they **rely on the available relational information**.

- ☺ Relationships **localize predictions** w.r.t. related time series.
- ☹ Often, **no pre-defined** graph is accessible.
- ☹ Available **relational information can be incomplete** or misspecified.
- ? Can we **efficiently learn a graph from data**?
- ? Can we **keep computations sparse**?

Latent Graph Learning



We want to learn a graph for spatiotemporal message-passing end-to-end.

- ! The learned graph should **maximize performance at task**.
- ! We want to **keep message-passing operations sparse**.
- ⚡ We exploit a **probabilistic framework** to learn distributions over graphs:

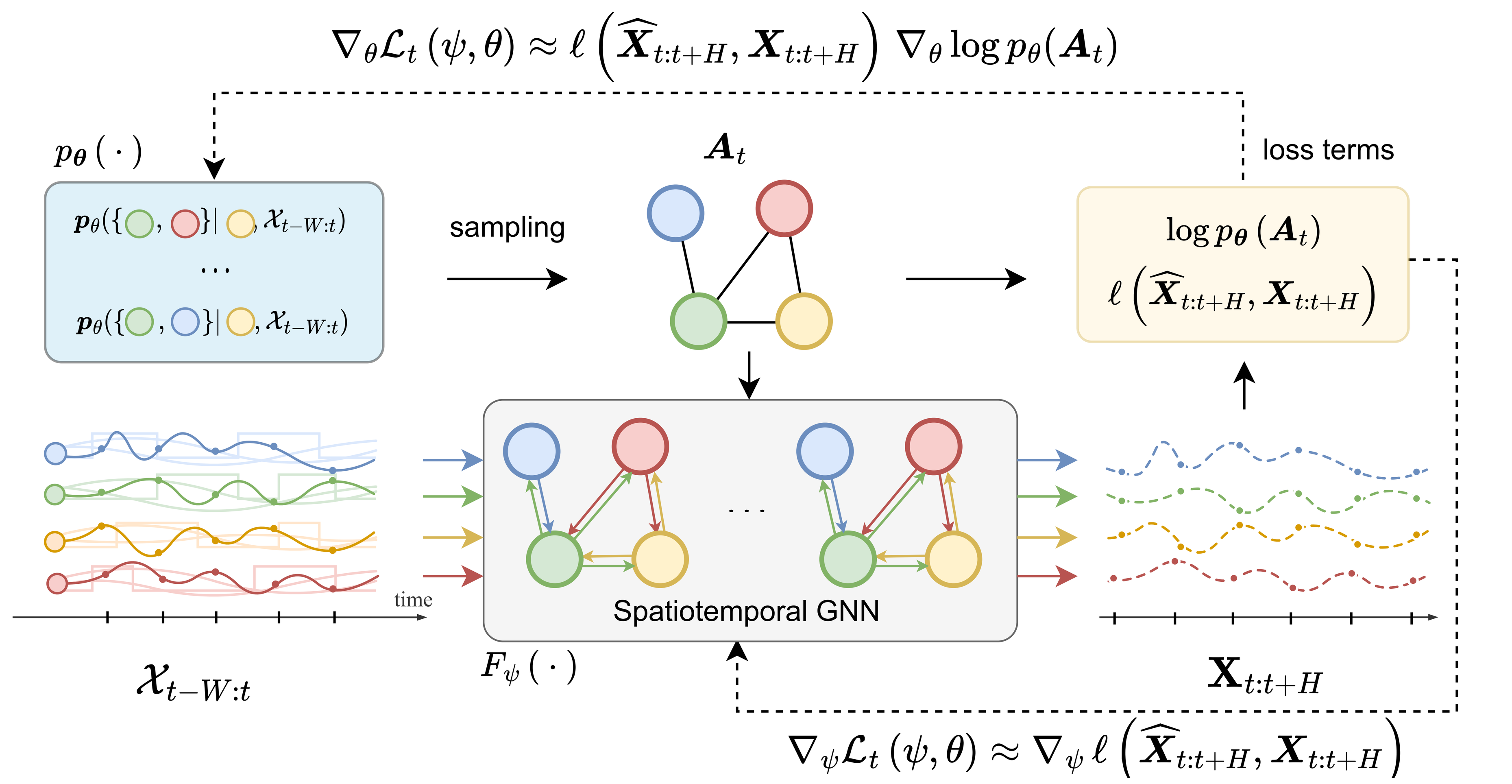
$$\widehat{\boldsymbol{A}}_t \sim p_\theta\left(\boldsymbol{A} \mid \mathcal{X}_{t-W:t}\right) \qquad \widehat{\boldsymbol{X}}_{t:t+H} = F_\psi\left(\mathcal{X}_{t-W:t}\,,\ \widehat{\boldsymbol{A}}_t\right),$$

$$\hat{\theta}, \hat{\psi} = \arg\min_{\theta,\psi} \mathcal{L}_t(\psi,\theta) = \arg\min_{\theta,\psi} \mathbb{E}_{\widehat{\boldsymbol{A}}_t \sim p_\theta}\left[\underbrace{\ell\left(\widehat{\boldsymbol{X}}_{t:t+H}, \boldsymbol{X}_{t:t+H}\right)}_{\delta_t(\widehat{\boldsymbol{A}}_t;\psi)}\right]$$

Score-based Gradient Estimators

$$\nabla_\theta \mathbb{E}_{p_\theta}\left[\delta_t(\widehat{\boldsymbol{A}}_t;\psi)\right] = \mathbb{E}_{p_\theta}\left[\delta_t(\widehat{\boldsymbol{A}}_t;\psi)\nabla_\theta \log p_\theta(\widehat{\boldsymbol{A}}_t)\right]$$
$$\approx \frac{1}{M}\sum_{i=1}^{M} \delta_t(\widehat{\boldsymbol{A}}_t^{(i)};\psi)\nabla_\theta \log p_\theta(\widehat{\boldsymbol{A}}_t^{(i)})$$

- ☺ Computationally efficient.
- ☺ Allows for sparse computations.
- ☹ High variance gradient estimates.
- ☹ High sample complexity.



Graph Samplers

Binary Edge Sampler (BES)

$$p_\theta(\boldsymbol{A}_t[i,j]=1) = $$
$$= \text{Bernoulli}(\sigma(\Phi_t[i,j]))$$

▸ BES models the probability of sampling each edge independently.

Subset Neighborhood Sampler (SNS)

$$p_\theta(\mathcal{N}(n) = S_K) =$$
$$= \sum_{\vec{S}_K \in \mathcal{P}(S_K)} \prod_{j \in \vec{S}_K} \frac{\exp(\Phi_t[n,j])}{1 - \sum_{k<j}\exp(\Phi_t[n,k])}.$$

▸ SNS samples $\mathbf{K}$ neighbors for each node, imposing sparsity ($|\mathcal{N}(\mathbf{n})| = \mathbf{K}$).

Improving Sample Efficiency

⚡ A **variance-reduced** estimator for each sampler (BES, SNS) based on control variates:

$$\nabla_\theta \mathbb{E}_{p_\theta}\left[\delta_t(\widehat{\boldsymbol{A}}_t;\psi)\right] \approx \left(\delta_t(\widehat{\boldsymbol{A}}_t;\psi) - \delta_t(\boldsymbol{A}_t^\mu;\psi)\right)\nabla_\theta \log p_\theta(\widehat{\boldsymbol{A}}_t).$$

We theoretically show that the **Fréchet mean of the graph distribution** is a sensible choice for $\boldsymbol{A}_t^\mu$ (see Prep. 1–4).

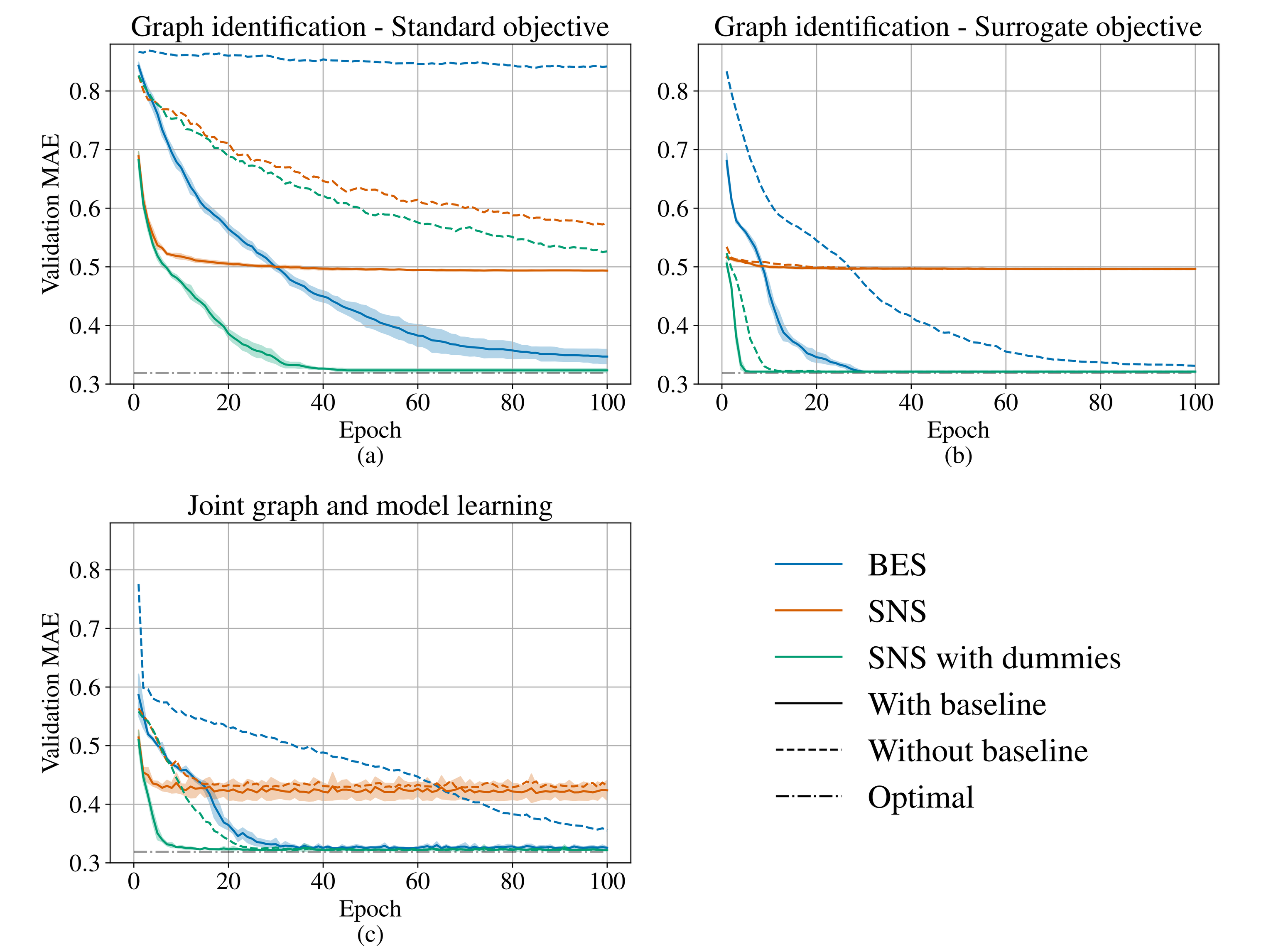
⚡ A **surrogate loss** to reweight the contribution of each error:

$$\nabla_\theta \mathcal{L}_t(\psi,\theta) \approx \mathbb{E}_{p_\theta}\left[\lambda\delta_t(\widehat{\boldsymbol{A}}_t;\psi)\nabla_\theta \log p_\theta(\widehat{\boldsymbol{A}}_t) + \sum_{i=1}^{N}\delta_t^i(\widehat{\boldsymbol{A}}_t;\psi)\nabla_\theta \log p_\theta(\widehat{\boldsymbol{A}}_t[i,:])\right],$$

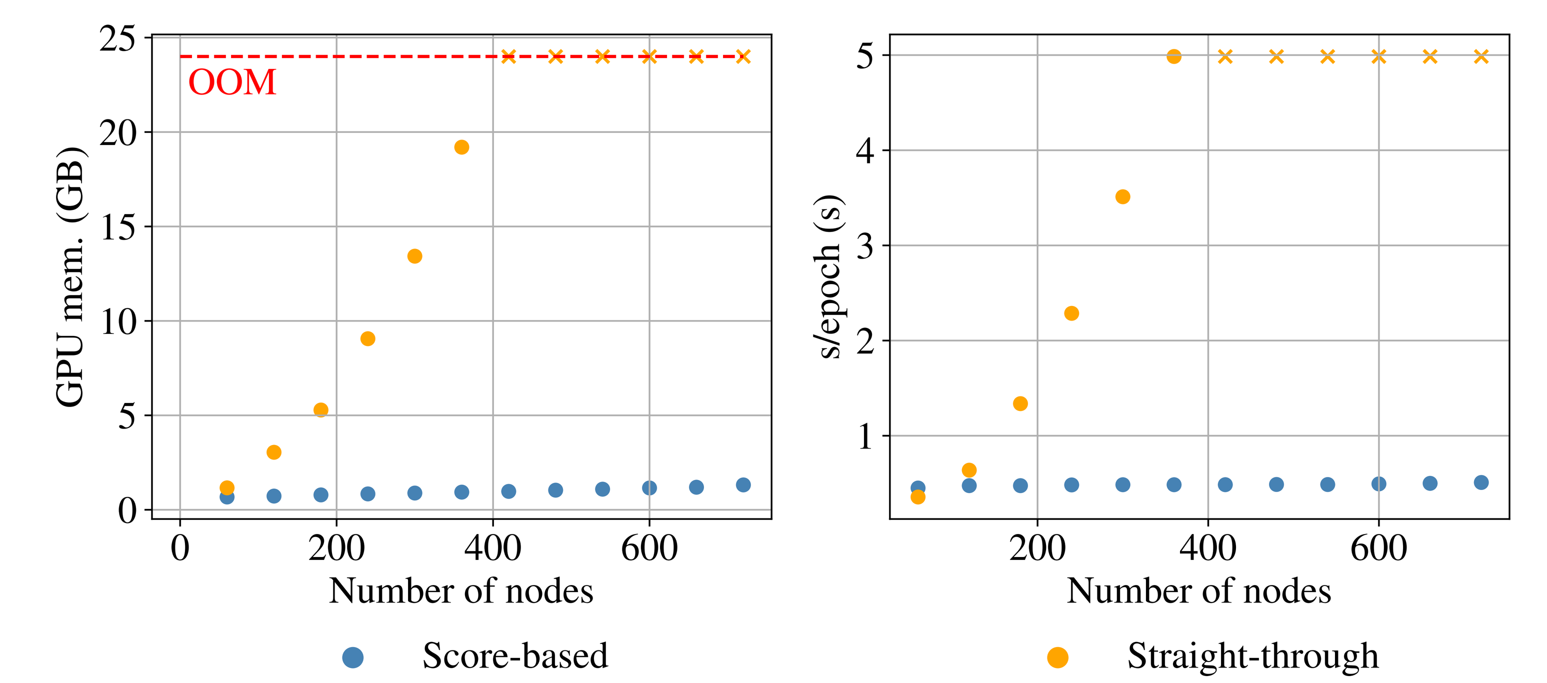
with λ trading of bias and variance (see Prep. 5 and Sec. 7.1). The surrogate objective **consistently reduces sample complexity**.

Some Empirical Results

Model training



Computational scalability



* Experiments on synthetic data.

andreacini/sparse-graph-learning

TorchSpatiotemporal/tsl