Learning to Reconstruct Missing Data from Spatiotemporal Graphs with Sparse Observations

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

[1]The Swiss AI Lab IDSIA, Università della Svizzera italiana [2]Politecnico di Milano *Equal contribution

Motivation

Missing values in **spatiotemporal** time series is an unavoidable problem when dealing with real-world applications.

- The state of the art includes **deep autoregressive methods**.
- Missing data blocks are often extended in time.
- Resulting observations are highly sparse.
- Autoregressive approaches suffer from **error compounding**.

Limits of Autoregressive Methods

SOTA autoregressive approaches model 3 different processes to **account for the available data**:

- $p\left(\boldsymbol{x}_t^i \mid \boldsymbol{X}_{<t}\right)$, for precedent observations.
- $p\left(\boldsymbol{x}_t^i \mid \boldsymbol{X}_{>t}\right)$, for subsequent observations.
- $p\left(\boldsymbol{x}_t^i \mid \{\boldsymbol{x}_t^{j \neq i}\}\right)$, for concurrent observations.

DRAWBACKS:

- Computational **overhead**.
- **Error compounding**.
- Hard to capture **global context**.

- Drawbacks are more evident in **extremely sparse settings**.

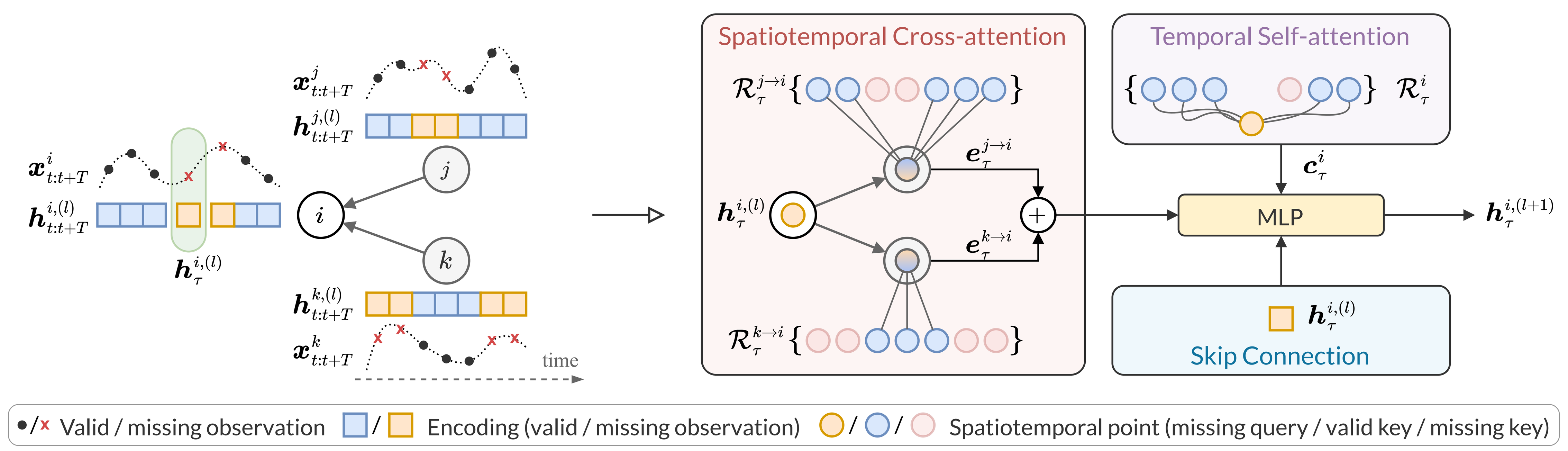
Our Different Approach

We associate **spatiotemporal coordinates** $\boldsymbol{Q}_t \in \mathbb{R}^{N \times d_q}$ to each point in time and space. And define:

- the **observed set** $\mathcal{X}_{t:t+T} = \left\{\left\langle \boldsymbol{x}_\tau^i, \boldsymbol{q}_\tau^i \right\rangle\right\}$ with valid observations.
- the **target set** $\mathcal{Y}_{t:t+T} = \left\{\boldsymbol{q}_\tau^i\right\}$ with target spatiotemporal points.

Given connectivity $\boldsymbol{A}$, for all $\boldsymbol{q}_\tau^i \in \mathcal{Y}_{t:t+T}$ we aim at modeling

$$p\left(\boldsymbol{x}_\tau^i \mid \boldsymbol{q}_\tau^i, \mathcal{X}_{t:t+T}, \boldsymbol{A}\right) \quad (1)$$



Spatiotemporal Point Inference Network (SPIN)

SPIN is an attention-based GNN imputation model f_θ such that

$$f_\theta(\boldsymbol{q}_\tau^i \mid \mathcal{X}_{t:t+T}, \boldsymbol{A}) \approx \mathbb{E}\left[p\left(\boldsymbol{x}_\tau^i \mid \boldsymbol{q}_\tau^i, \mathcal{X}_{t:t+T}, \boldsymbol{A}\right)\right] \quad (2)$$

Representations at i-th node for each τ-th time step are learned with **two main components**:

TEMPORAL SELF-ATTENTION

- **For each node** $\boldsymbol{h}_s^i$ **associated with a valid observation**, compute
$$\boldsymbol{r}_{s \to \tau}^i = \text{SelfMessage}\left(\boldsymbol{h}_s^i, \boldsymbol{h}_\tau^i\right)$$
- Then compute **self-attention scores** $\boldsymbol{\alpha}$ from $\boldsymbol{r}$ and aggregate:
$$\boldsymbol{c}_\tau^i = \sum_s \alpha_{s \to \tau}^i \cdot \boldsymbol{r}_{s \to \tau}^i$$

SPATIOTEMPORAL CROSS-ATTENTION

- **For each neighbor** $\boldsymbol{h}_s^j$ **associated with a valid observation**, we compute
$$\boldsymbol{r}_{s \to \tau}^{j \to i} = \text{CrossMessage}\left(\boldsymbol{h}_s^j, \boldsymbol{h}_\tau^i\right)$$
- Then compute **cross-attention scores** $\boldsymbol{\alpha}$ from $\boldsymbol{r}$ and aggregate for each neighbor:
$$\boldsymbol{e}_\tau^{j \to i} = \sum_s \alpha_{s \to \tau}^{j \to i} \cdot \boldsymbol{r}_{s \to \tau}^{j \to i}$$

Then, target representation $\boldsymbol{h}_\tau^{i,(l)}$ is updated with a final aggregation step

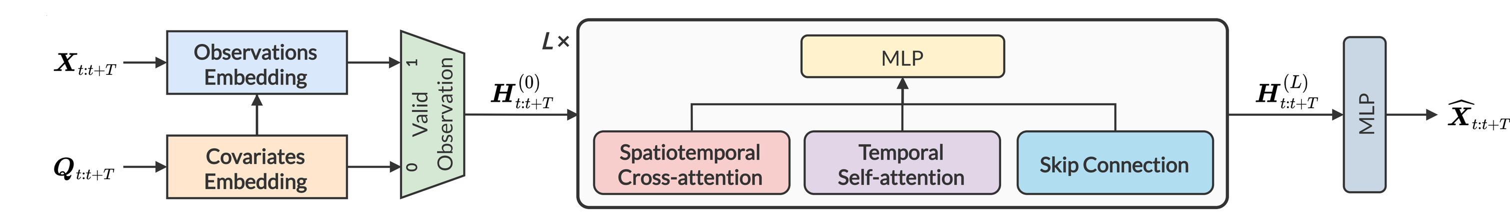
$$\boldsymbol{h}_\tau^{i,(l+1)} = \text{MLP}\left(\boldsymbol{h}_\tau^{i,(l)}, \ \boldsymbol{c}_\tau^{i,(l)}, \ \sum_{j \in \mathcal{N}(i)} \boldsymbol{e}_\tau^{j \to i,(l)}\right) \quad (3)$$

Imputations for all spatiotemporal points in $\mathcal{Y}_{t:t+T}$ are obtained as

$$\hat{\mathcal{Y}}_{t:t+T} = \{\hat{\boldsymbol{x}}_\tau^i = \text{MLP}\left(\boldsymbol{h}_\tau^{i,(L)}\right) \mid \boldsymbol{q}_\tau^i \in \mathcal{Y}_{t:t+T}\} \quad (4)$$

Covariates are **positional encodings** for observations: representation $\boldsymbol{h}_\tau^{i,(0)}$ is initialized as

$$\boldsymbol{h}_\tau^{i,(0)} = \text{MLP}\left(\boldsymbol{q}_\tau^i, \boldsymbol{x}_\tau^i\right) \ \text{if} \ \left\langle \boldsymbol{x}_\tau^i, \boldsymbol{q}_\tau^i \right\rangle \in \mathcal{X}_{t:t+T} \qquad \boldsymbol{h}_\tau^{i,(0)} = \text{MLP}\left(\boldsymbol{q}_\tau^i\right) \ \text{if} \ \boldsymbol{q}_\tau^i \in \mathcal{Y}_{t:t+T}$$



The SPIN architecture.

Hierarchical Attention

The base SPIN layer has $\mathcal{O}\left((N+E)T^2\right)$ complexity. To remove the quadratic term, we rewire attention to be **hierarchical**:

- Add K **dummy nodes** that act as hubs.
- Update the $\boldsymbol{z}^i \in \mathbb{R}^{d_z}$ hubs' representations by **querying the available nodes** $\{\boldsymbol{h}_\tau^i\}$.
- Update node encoding $\boldsymbol{h}_\tau^i$ by **querying the updated hubs** $\{\tilde{\boldsymbol{z}}_k^j\}$ for each neighbor $j \in \mathcal{N}(i)$.

This reduces complexity to $\mathcal{O}\left((N+E)KT\right)$ with $K \ll T$. We refer to this variation as **SPIN-H**.

Some Empirical Results

We report here the performance in reconstructing long blocks of consecutive missing values (from 12 to 36 time steps).

Table 1. Performance (MAE) with an increasing number of simulated failures.

	METR-LA			PEMS-BAY		
	Failure probability			Failure probability		
	5 %	10 %	15 %	5 %	10 %	15 %
BRITS	5.87 ± 0.03	7.26 ± 0.06	8.29 ± 0.07	4.14 ± 0.05	5.41 ± 0.08	5.84 ± 0.04
SAITS	4.73 ± 0.07	6.66 ± 0.05	7.27 ± 0.03	3.88 ± 0.09	7.62 ± 0.21	8.01 ± 0.11
Transformer	6.03 ± 0.04	7.19 ± 0.05	8.06 ± 0.05	3.69 ± 0.06	5.09 ± 0.05	6.02 ± 0.04
GRIN	3.05 ± 0.02	4.52 ± 0.05	5.82 ± 0.06	2.26 ± 0.03	3.45 ± 0.06	4.35 ± 0.04
SPIN	2.71 ± 0.02	3.32 ± 0.02	3.87 ± 0.05	**1.78 ± 0.03**	**2.15 ± 0.03**	**2.41 ± 0.02**
SPIN-H	**2.64 ± 0.02**	**3.17 ± 0.02**	**3.61 ± 0.04**	**1.75 ± 0.04**	**2.16 ± 0.03**	2.48 ± 0.02

Our library for neural spatiotemporal data processing:

TorchSpatiotemporal/tsl

ivan.marisca@usi.ch – andrea.cini@usi.ch