Appendix

A Detailed experimental setup

In this appendix, we discuss in detail the experimental settings. We use the same setup of Cini et al. [4][4][5]. Indeed, Table 1 reports results from [4] whenever possible. We refer to [4] for details on these baselines.

For SPIN, we use the same hyperparameters in all datasets: $L = 4$ layers, with first $\eta = 3$ layers with masked connections; hidden size $d_h = 32$; 2 layers with hidden size 32 for every MLP; ReLU activation functions. For SPIN-H, we use similar hyperparameters, but 5 layers, with $\eta = 3$; $K = 4$ hubs per node with $d_z = 128$ units each. These hyperparameters have been selected among a small subset of options on the validation set; we expect far better performance to be achievable with further hyperparameter tuning. Depending on the dataset, the number of parameters ranges from $\approx 55K$ to $\approx 95K$ for SPIN and $\approx 540K$ to $\approx 800K$ for SPIN-H. We use Adam optimizer [61], learning rate $lr = 0.0008$ and a cosine scheduler with a warm-up of 12 steps and (partial) restarts every 100 epochs. We train our models with 300 mini-batches of 8 random samples per epoch, fixing the maximum number of epochs to 300 and using early stopping on the validation set with a patience of 40 epochs. Due to constraints on memory capacity on some of the GPUs (see the description of the hardware resources below), for SPIN-H we set the batch size to 6 and 16 in AQI and AQI-36, respectively.

To train SPIN-based models, we minimize the following loss function:

$$\mathcal{L} = \sum_{l=1}^{L} \frac{\sum_{\boldsymbol{q}_\tau^i \in \mathcal{Y}_{t:t+T}} \ell\left(\hat{\boldsymbol{x}}_\tau^{i,(l)}, \boldsymbol{x}_\tau^i\right)}{|\mathcal{Y}_{t:t+T}|}, \tag{23}$$

where $\ell\left(\,\cdot\,,\,\cdot\,\right)$ is the absolute error and $\hat{\boldsymbol{x}}_\tau^{i,(l)}$ is l-th layer imputation for the i-th node at time step τ. Note that, to provide more supervision to the architecture, the loss is computed and backpropagated w.r.t. representations learned at each layer, not only at the last one. The error is computed only on data not seen by the model at each forward pass. For this reason, we randomly remove p ratio of the input data for each mini-batch sample, with p sampled uniformly from $[0.2, 0.5, 0.8]$, and use them to compute the loss. We never use data masked for evaluation to train any model.

For the spatiotemporal Transformer baseline, we use the same training strategy and a similar hyperparameters configuration of SPIN-H: $L = 5$ layers; 4 attention heads; hidden size and feed-forward size of 64 and 128 units, respectively. For SAITS, we use the code provided by the authors[6]. Hyperparameters for SAITS have been selected on the validation set with a random search by using hyperparameter ranges from the original paper.

All the models were developed in Python [62] using PyTorch [57], PyG [63] and Torch Spatiotemporal [58]. We use Neptune[7] [64] for experiments tracking. The code to reproduce the experiments of the paper is available as supplementary material. All the experiments have been run in a cluster using GPU-enabled nodes with different hardware setups. Running times of SPIN-H training on a node equipped with a 12GB NVIDIA Titan V GPU range from 4 to 14 hours (depending on the dataset). For SPIN we used a node with 40GB NVIDIA A100 GPU, with running times ranging from 4 to 26 hours.

B Datasets

In this appendix, we provide details on datasets and preprocessing used for the experiments. We use temporal windows of $T = 24$ steps for all datasets except AQI-36, for which we set $T = 36$. For traffic datasets, we split the data sequentially as 70% for training, 10% for validation, and 20% for testing. For air quality datasets, following Yi et al. [15], we consider as the test set the months of March, June, September, and December and we use valid observation $\boldsymbol{x}_\tau^i$ as ground-truth if the

---

[4] `https://github.com/Graph-Machine-Learning-Group/grin`

[5] `https://github.com/TorchSpatiotemporal/tsl`

[6] `https://github.com/WenjieDu/SAITS`

[7] `https://neptune.ai/`

Table 4: Ablation study to assess the contribution of the single components in the spatiotemporal attention block. Performance averaged over 5 independent runs.

	METR-LA (P)		AQI-36	
	MAE	MRE (%)	MAE	MRE (%)
SPIN	**1.90** ± **0.01**	**3.29** ± **0.01**	11.77 ± 0.54	16.56 ± 0.76
SPIN-H	1.96 ± 0.03	3.40 ± 0.05	**10.89** ± **0.27**	**15.32** ± **0.38**
Without cross-attention	2.18 ± 0.01	3.78 ± 0.01	15.47 ± 0.22	21.77 ± 0.31
Without self-attention	2.21 ± 0.08	3.82 ± 0.14	13.76 ± 0.30	19.37 ± 0.42
Transformer	2.16 ± 0.00	3.74 ± 0.01	11.98 ± 0.53	16.87 ± 0.75

value is missing at the same hour and day in the following month. For data preprocessing we use the same approach of Cini et al. [4], by normalizing data across the feature dimension (graph-wise for graph-based models) to zero mean and unit variance.

In line with [3, 4], we obtain the adjacency matrix from the node pairwise geographical distances using a thresholded Gaussian kernel [65]

$$a^{i,j} = \begin{cases} \exp\left(-\frac{\mathrm{dist}(i,j)^2}{\gamma}\right) & \mathrm{dist}\,(i,j) \leq \delta \\ 0 & \text{otherwise} \end{cases}, \tag{24}$$

where $\mathrm{dist}\,(\,\cdot\,,\,\cdot\,)$ is the geographical distance operator, γ is a shape parameter and δ is the threshold.

C Virtual sensing

The spatiotemporal cross-attention mechanism allows SPIN variants to exploit valid observations at neighboring nodes to impute missing values at spatial locations where no sensor is available, thus performing *virtual sensing* (also referred to as *kriging* [66]). We assess the performance of SPIN-H in the virtual sensing task by completely masking out observations at two nodes during training and then using the trained model to infer the entire sequences at the masked nodes. For this experiment, we keep the same architecture hyperparameters and consider the dataset AQI-36, masking out the two nodes with the highest (station no. 1015) and lowest (no. 1032) number of neighbors, reproducing the experimental settings of [4]. Figure 3 shows the performance of SPIN-H in reconstructing a temporal window of 36 time steps for the two virtual sensors at test time. Results qualitatively show that our method is able to reconstruct observations for sensors with no available information other than their location in the network.

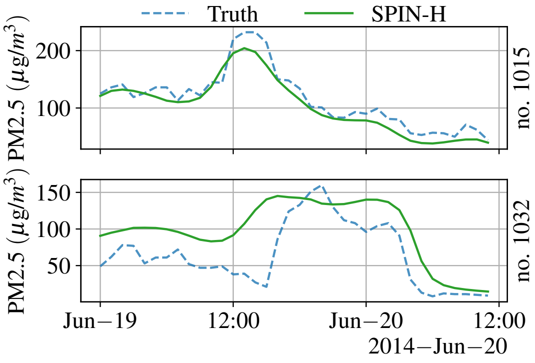

Figure 3: Visualization of completely missing sequences reconstructed by SPIN-H at nodes masked out during training. Imputations averaged over 5 independent runs.

D Ablation study

Table 4 shows the results of an ablation study on METR-LA (Point missing) and AQI-36. Here, we evaluate the performance in terms of mean absolute error (MAE) and mean relative error (MRE). We consider two different versions of SPIN-H in which we remove the spatiotemporal cross-attention and the temporal self-attention components, respectively. We also report the performance of SPIN, SPIN-H and the Transformer for reference. Results clearly show that both components contribute positively to imputation accuracy. We also point out that in METR-LA (P) observations are masked out uniformly at random while the mask in AQI-36 reflects the empirical distribution of missing data in the dataset.