Filling the G_ap_s: Multivariate time series imputation by Graph Neural Networks

ICLR 2022 - Tenth International Conference on Learning Representations

Andrea Cini[*1], Ivan Marisca[*1], Cesare Alippi[1,2]

[1] The Swiss AI Lab IDSIA, Università della Svizzera italiana, Lugano, CH

[2] Politecnico di Milano, Milano, IT

[*] Equal contribution

The problem of missing data

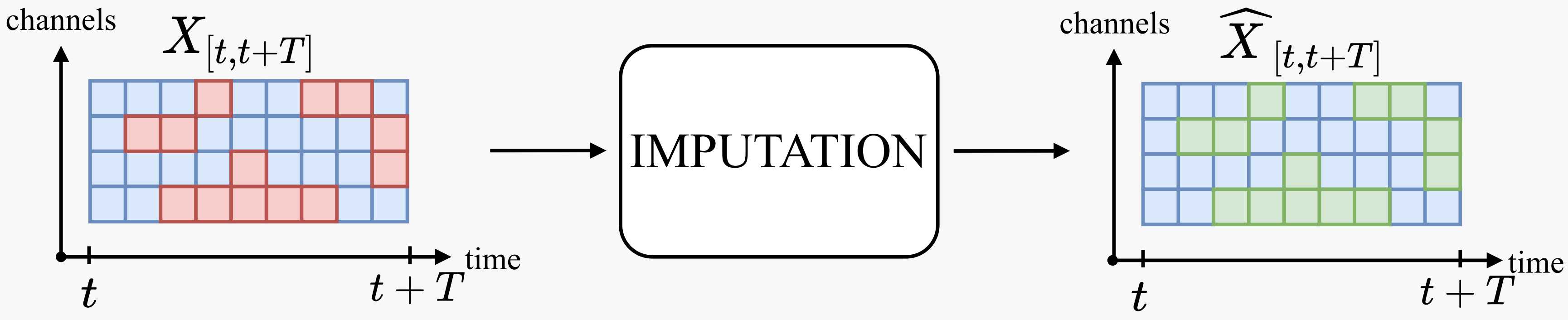
In real-world data acquisition systems (e.g., sensor networks), it is not rare that system faults results in missing data in the acquired stream.

However, many signal processing methods rely on complete, regularly sampled sequences.

We need a way to infer, i.e., impute, missing observations.

Multivariate time series imputation

The objective of multivariate time series imputation (MTSI) is to properly fill missing values in a (multivariate) sequence of data $X_{[t,t+T]}$.



Group all valid observations into set $\mathcal{X}_{[t,t+T]} = \{\mathbf{x}_t^i \mid \mathbf{x}_t^i \in X_{[t,t+T]},\ \mathbf{x}_t^i \text{ is valid}\}$, this problem translates into estimating missing observations as

$$\hat{\mathbf{x}}_t^i \approx \mathbb{E}\left[p\left(\mathbf{x}_t^i \mid \mathcal{X}_{[t,t+T]}\right)\right] \qquad \forall\, i, t \text{ such that } \mathbf{x}_t^i \notin \mathcal{X}_{[t,t+T]}$$

Embedding relational inductive biases

Common deep learning solutions consist in using autoregressive models for sequential data:

- RNNs
- TCNs

The relational constraints are often strong (e.g., in sensor networks) and embedding them in the information processing can be extremely beneficial.

- Most state-of-the-art deep learning methods for MTSI overlook this aspect.

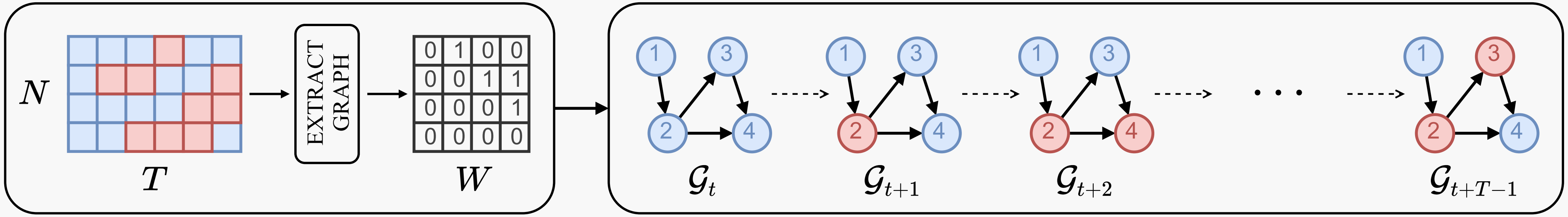
We propose a different point of view by casting the problem in the graph-processing settings, modelling multivariate time series as sequences of attributed graphs.

Multivariate TS as a sequence of graphs

We can describe any multivariate timeseries $\mathbf{X} \in \mathbb{R}^{T \times N \cdot d}$ as a sequence of T attributed graphs $\mathcal{G}_t(\boldsymbol{X}_t, \boldsymbol{W})$, with node attribute matrix $\boldsymbol{X}_t \in \mathbb{R}^{N_t \times d}$ and adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{N \times N}$.

If no relational information is available, the adjacency matrix can be obtained from:

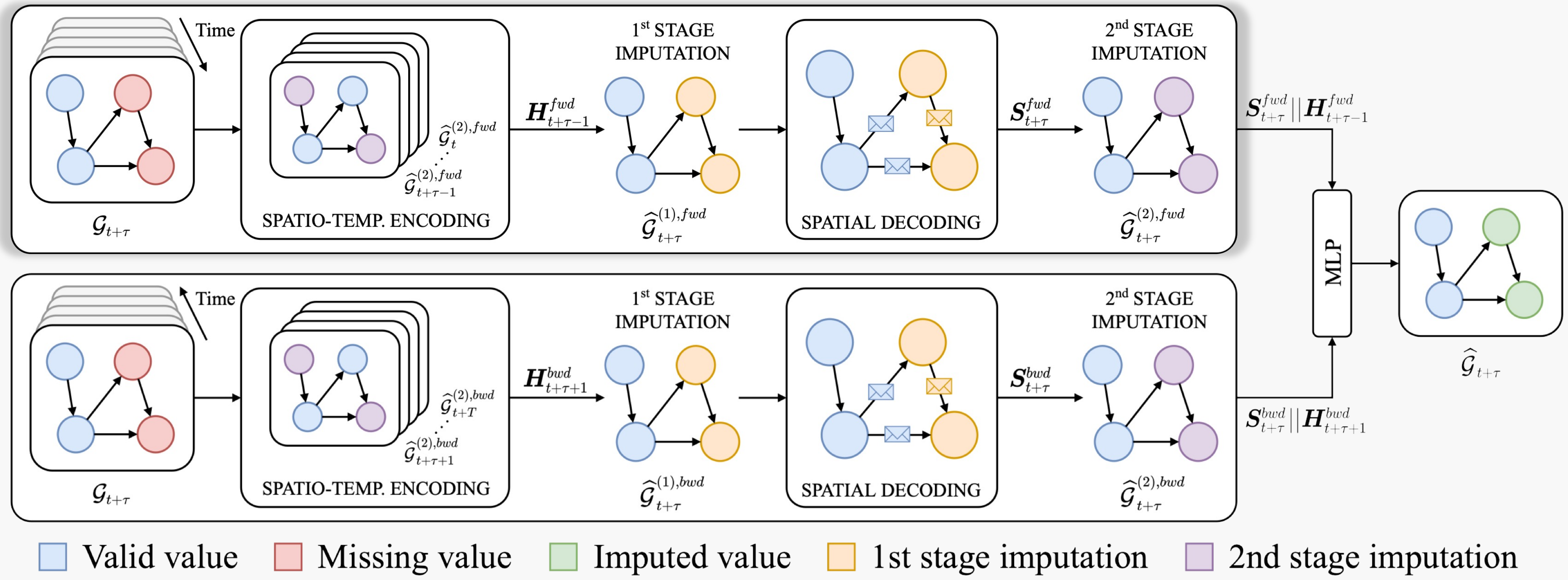
- Pairwise similarity (e.g., Pearson correlation).
- More advanced methods (e.g., graph learning).



These types of representations are popular among spatio-temporal forecasting methods.

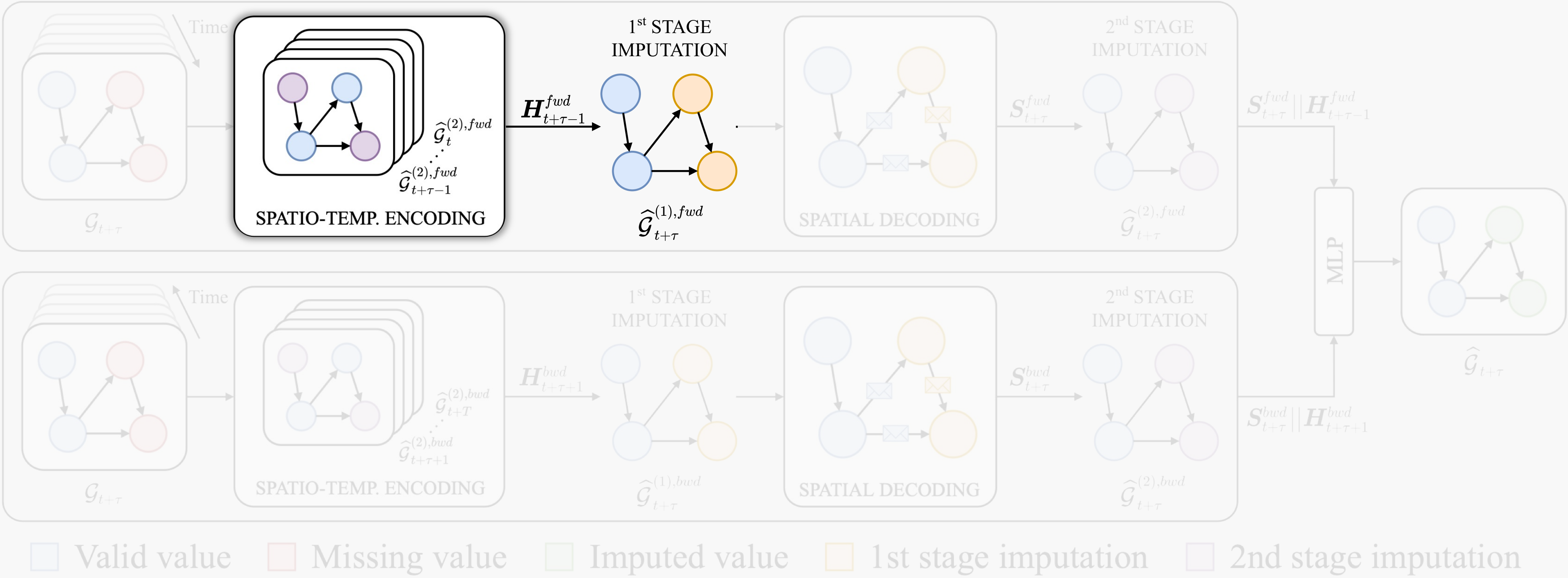
Graph Recurrent Imputation Network (GRIN)

GRIN is a graph-based, bidirectional, recurrent neural network which aims to reconstruct the input sequence by leveraging on both the temporal and spatial dimensions.



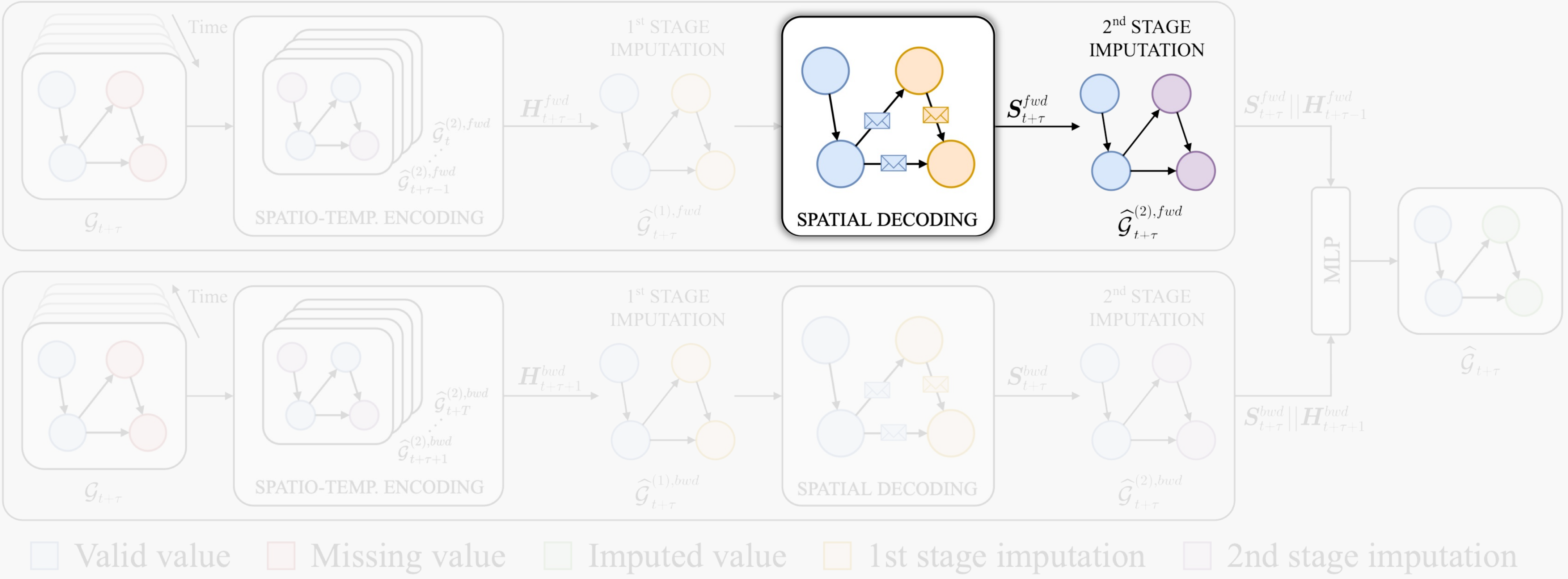
Graph Recurrent Imputation Network (GRIN)

1. Feed a recurrent GNN with $\hat{\mathcal{G}}^{(2)}_{t+\tau-1}$ and obtain representation $\boldsymbol{H}_{t+\tau-1}$
2. Impute missing values in i-th node features $\boldsymbol{x}^{i}_{t+\tau}$ using $\boldsymbol{h}^{i}_{t+\tau-1}$ $\Rightarrow$ $\hat{\mathcal{G}}^{(1)}_{t+\tau}$



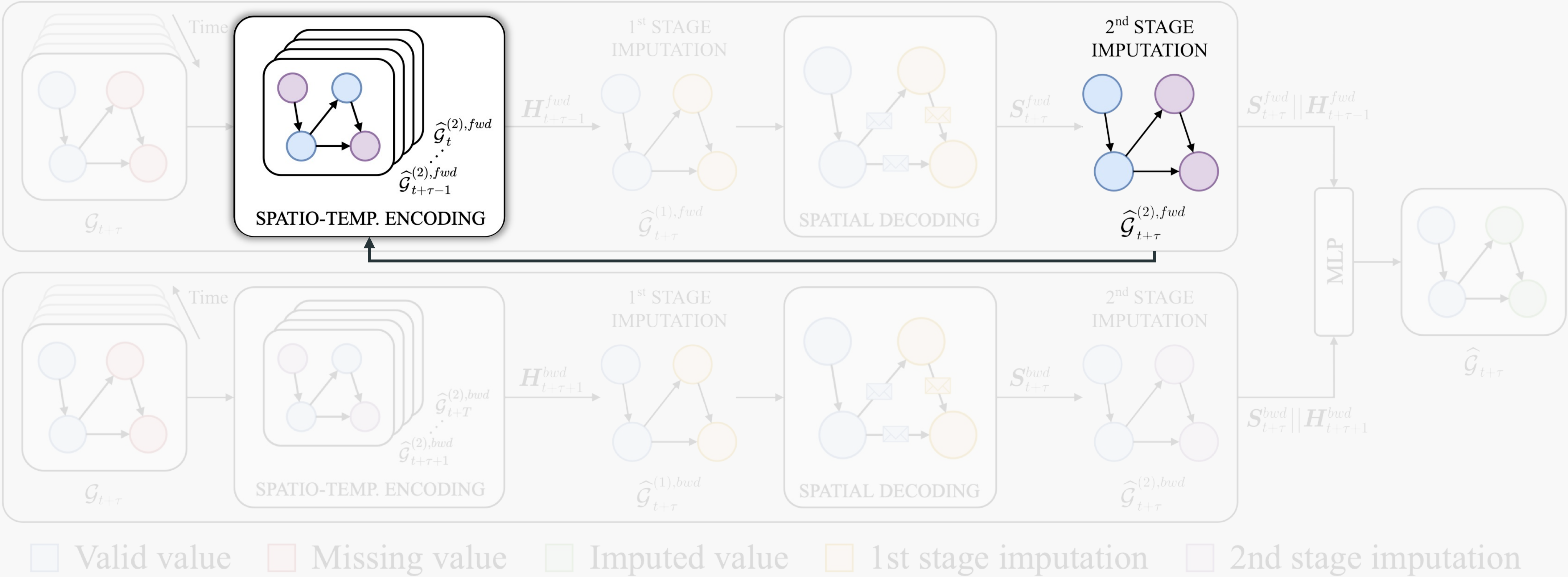
Graph Recurrent Imputation Network (GRIN)

3. Exploit relationships between nodes at time $t+\tau$ through a GNN and obtain $\boldsymbol{S}_{t+\tau}$
4. Refine imputations using $\boldsymbol{S}_{t+\tau} \Rightarrow \hat{\mathcal{G}}^{(2)}_{t+\tau}$



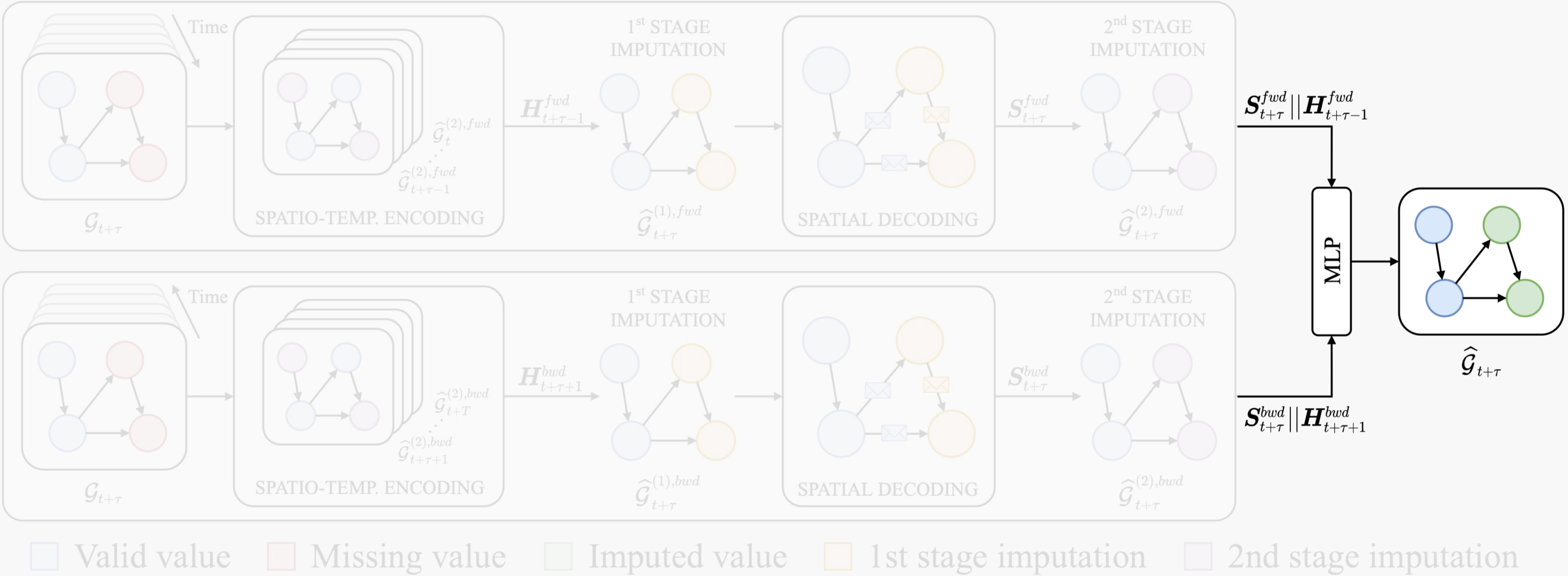
Graph Recurrent Imputation Network (GRIN)

The 2nd stage imputation $\hat{\mathcal{G}}^{(2)}_{t+\tau}$ is then fed back to the recurrent GNN to update the state, obtaining representation $\boldsymbol{H}_{t+\tau}$.



Graph Recurrent Imputation Network (GRIN)

Obtain final imputations by combining (with an MLP) the representations extracted by processing the sequence in both forward and backward directions.



Experimental setting

We compare the performances in the imputation task of the following approaches:

- `Mean` – impute using the average value in the series
- `KNN` – take the average of the (observed) values of the neighbors
- `MF` (Matrix Factorization) – factorize sequence into lower-dimensional matrices and reconstruct
- `MICE` – iterative method based on chained equation
- `VAR` – 1-step-ahead VAR predictor of order 5

Deep learning SOTA

- `rGAIN` – generative adversarial method (with RNN encoder/decoder)
- `BRITS` – deep autoregressive method

Graph-based

- `MPGRU` – 1-step-ahead graph predictor with recurrent cell similar to `GRIN`
- `GRIN`

We test all the methods on three real-world datasets coming from relevant application domains: air quality monitoring, traffic and smart grids.

Traffic and energy consumption

PEMS-BAY

- 325 traffic sensors from the San Francisco Bay Area
- Average speed (mph)
- Every 5 minutes for 6 months
- Thresholded gaussian kernel on distances

CER-E

- 485 smart meters installed by the premises of Irish small and medium enterprises
- Energy consumption (kWh)
- Every 30 minutes for 1.5 years
- Thresholded correntropy similarity

For both scenarios, we simulate two different settings:

Point missing

Each sensor has a probability p of failing in transmitting the recorded value at each time step.

Block missing

Each sensor has a probability p of failing for $t \in [t_{\min}, t_{\max}]$ consecutive time steps.

https://www.ucd.ie/issda/data/commissionforenergyregulationcer/

Air Quality

AQI-36 (subset of AQI)

- 36 sensors in Beijing
- Pollutant PM2.5 ($\mu g/m^3$)
- Every hour for 1 year
- Real missing rate $\sim 13\%$
- Thresholded gaussian kernel on distances

AQI

- 437 sensors spread over 43 cities in China
- Pollutant PM2.5 ($\mu g/m^3$)
- Every hour for 1 year
- Real missing rate $\sim 26\%$
- Same as AQI-36

For both datasets, we consider two different settings:

In-sample

The model is trained on all the available data except those that are missing.

Out-of-sample

The model is trained and evaluated on disjoint sequences.

In both cases the model does not have access to the ground-truth data used for the final evaluation.

https://www.microsoft.com/en-us/research/project/urban-air/

Traffic and energy consumption – Results

Table 1: Results on the traffic and smart grids datasets. Performance averaged over 5 runs.

D	M	Block missing MAE	Block missing MSE	Block missing MRE(%)	Point missing MAE	Point missing MSE	Point missing MRE(%)
PEMS-BAY	Mean	5.46 ± 0.00	87.56 ± 0.00	8.75 ± 0.00	5.42 ± 0.00	86.59 ± 0.00	8.67 ± 0.00
	KNN	4.30 ± 0.00	49.90 ± 0.00	6.90 ± 0.00	4.30 ± 0.00	49.80 ± 0.00	6.88 ± 0.00
	MF	3.28 ± 0.01	50.14 ± 0.13	5.26 ± 0.01	3.29 ± 0.01	51.39 ± 0.64	5.27 ± 0.02
	MICE	2.94 ± 0.02	28.28 ± 0.37	4.71 ± 0.03	3.09 ± 0.02	31.43 ± 0.41	4.95 ± 0.02
	VAR	2.09 ± 0.10	16.06 ± 0.73	3.35 ± 0.16	1.30 ± 0.00	6.52 ± 0.01	2.07 ± 0.01
	rGAIN	2.18 ± 0.01	13.96 ± 0.20	3.50 ± 0.02	1.88 ± 0.02	10.37 ± 0.20	3.01 ± 0.04
	BRITS	1.70 ± 0.01	10.50 ± 0.07	2.72 ± 0.01	1.47 ± 0.00	7.94 ± 0.03	2.36 ± 0.00
	MPGRU	1.59 ± 0.00	14.19 ± 0.11	2.56 ± 0.01	1.11 ± 0.00	7.59 ± 0.02	1.77 ± 0.00
	GRIN	**1.14 ± 0.01**	**6.60 ± 0.10**	**1.83 ± 0.02**	**0.67 ± 0.00**	**1.55 ± 0.01**	**1.08 ± 0.00**
CER-E	Mean	1.49 ± 0.00	5.96 ± 0.00	72.47 ± 0.00	1.51 ± 0.00	6.09 ± 0.00	71.51 ± 0.00
	KNN	1.15 ± 0.00	6.53 ± 0.00	56.11 ± 0.00	1.22 ± 0.00	7.23 ± 0.00	57.71 ± 0.00
	MF*	0.97 ± 0.01	4.38 ± 0.06	47.20 ± 0.31	1.01 ± 0.01	4.65 ± 0.07	47.87 ± 0.36
	MICE	0.96 ± 0.01	3.08 ± 0.03	46.65 ± 0.44	0.98 ± 0.00	3.21 ± 0.04	46.59 ± 0.23
	VAR	0.64 ± 0.03	1.75 ± 0.06	31.21 ± 1.60	0.53 ± 0.00	1.26 ± 0.00	24.94 ± 0.02
	rGAIN	0.74 ± 0.00	1.77 ± 0.02	36.06 ± 0.14	0.71 ± 0.00	1.62 ± 0.02	33.45 ± 0.16
	BRITS	0.64 ± 0.00	1.61 ± 0.01	31.05 ± 0.05	0.64 ± 0.00	1.59 ± 0.01	30.07 ± 0.11
	MPGRU	0.53 ± 0.00	1.84 ± 0.01	25.88 ± 0.09	0.41 ± 0.00	1.22 ± 0.01	19.51 ± 0.03
	GRIN	**0.42 ± 0.00**	**1.07 ± 0.01**	**20.24 ± 0.04**	**0.29 ± 0.00**	**0.53 ± 0.00**	**13.71 ± 0.03**

Air Quality – Results

Table 2: Results on the air datasets. Performance averaged over 5 runs.

		In-sample			Out-of-sample		
D	M	MAE	MSE	MRE (%)	MAE	MSE	MRE (%)
AQI-36	Mean	53.48 ± 0.00	4578.08 ± 00.00	76.77 ± 0.00	53.48 ± 0.00	4578.08 ± 00.00	76.77 ± 0.00
	KNN	30.21 ± 0.00	2892.31 ± 00.00	43.36 ± 0.00	30.21 ± 0.00	2892.31 ± 00.00	43.36 ± 0.00
	MF	30.54 ± 0.26	2763.06 ± 63.35	43.84 ± 0.38	–	–	–
	MICE	29.89 ± 0.11	2575.53 ± 07.67	42.90 ± 0.15	30.37 ± 0.09	2594.06 ± 07.17	43.59 ± 0.13
	VAR	13.16 ± 0.21	513.90 ± 12.39	18.89 ± 0.31	15.64 ± 0.08	833.46 ± 13.85	22.02 ± 0.11
	rGAIN	12.23 ± 0.17	393.76 ± 12.66	17.55 ± 0.25	15.37 ± 0.26	641.92 ± 33.89	21.63 ± 0.36
	BRITS	12.24 ± 0.26	495.94 ± 43.56	17.57 ± 0.38	14.50 ± 0.35	662.36 ± 65.16	20.41 ± 0.50
	MPGRU	12.46 ± 0.35	517.21 ± 41.02	17.88 ± 0.50	16.79 ± 0.52	1103.04 ± 106.83	23.63 ± 0.73
	GRIN	**10.51 ± 0.28**	**371.47 ± 17.38**	**15.09 ± 0.40**	**12.08 ± 0.47**	**523.14 ± 57.17**	**17.00 ± 0.67**
AQI	Mean	39.60 ± 0.00	3231.04 ± 00.00	59.25 ± 0.00	39.60 ± 0.00	3231.04 ± 00.00	59.25 ± 0.00
	KNN	34.10 ± 0.00	3471.14 ± 00.00	51.02 ± 0.00	34.10 ± 0.00	3471.14 ± 00.00	51.02 ± 0.00
	MF	26.74 ± 0.24	2021.44 ± 27.98	40.01 ± 0.35	–	–	–
	MICE	26.39 ± 0.13	1872.53 ± 15.97	39.49 ± 0.19	26.98 ± 0.10	1930.92 ± 10.08	40.37 ± 0.15
	VAR	18.13 ± 0.84	918.68 ± 56.55	27.13 ± 1.26	22.95 ± 0.30	1402.84 ± 52.63	33.99 ± 0.44
	rGAIN	17.69 ± 0.17	861.66 ± 17.49	26.48 ± 0.25	21.78 ± 0.50	1274.93 ± 60.28	32.26 ± 0.75
	BRITS	17.24 ± 0.13	924.34 ± 18.26	25.79 ± 0.20	20.21 ± 0.22	1157.89 ± 25.66	29.94 ± 0.33
	MPGRU	15.80 ± 0.05	816.39 ± 05.99	23.63 ± 0.08	18.76 ± 0.11	1194.35 ± 15.23	27.79 ± 0.16
	GRIN	**13.10 ± 0.08**	**615.80 ± 10.09**	**19.60 ± 0.11**	**14.73 ± 0.15**	**775.91 ± 28.49**	**21.82 ± 0.23**

Conclusions

We introduced a novel GNN architecture (GRIN) which aims at reconstructing missing data in the different channels of a multivariate time series by learning spatio-temporal representations through message passing.

Code to reproduce the experiments available at:

https://github.com/Graph-Machine-Learning-Group/grin

Find more about our group at:

Graph Machine Learning Group
gmlg.ch

Andrea Cini

Ivan Marisca

Cesare Alippi

THANKS FOR THE ATTENTION

Questions?

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

Bibliography

1. Che, Z., Purushotham, S., Cho, K., Sontag, D. & Liu, Y. **Recurrent Neural Networks for Multivariate Time Series with Missing Values.** *Sci Rep* **8**, 6085 (2018).
2. Cao, W. *et al.* **BRITS: Bidirectional Recurrent Imputation for Time Series.** *Advances in Neural Information Processing Systems* **31**, (2018).
3. Yoon, J., Jordon, J. & Schaar, M. **GAIN: Missing Data Imputation using Generative Adversarial Nets**. in *International Conference on Machine Learning* 5689–5698 (PMLR, 2018).
4. Luo, Y., Cai, X., Zhang, Y., Xu, J. & Yuan, X. **Multivariate time series imputation with generative adversarial networks.** in *Proceedings of the 32nd International Conference on Neural Information Processing Systems* 1603–1614 (Curran Associates Inc., 2018).
5. Liu, Y., Yu, R., Zheng, S., Zhan, E. & Yue, Y**. NAOMI: Non-Autoregressive Multiresolution Sequence Imputation.** in *Advances in Neural Information Processing Systems* vol. 32 (Curran Associates, Inc., 2019).
6. Yi, X., Zheng, Y., Zhang, J. & Li, T. **ST-MVL: Filling Missing Values in Geo-sensory Time Series Data.** In IJCAI 2016.