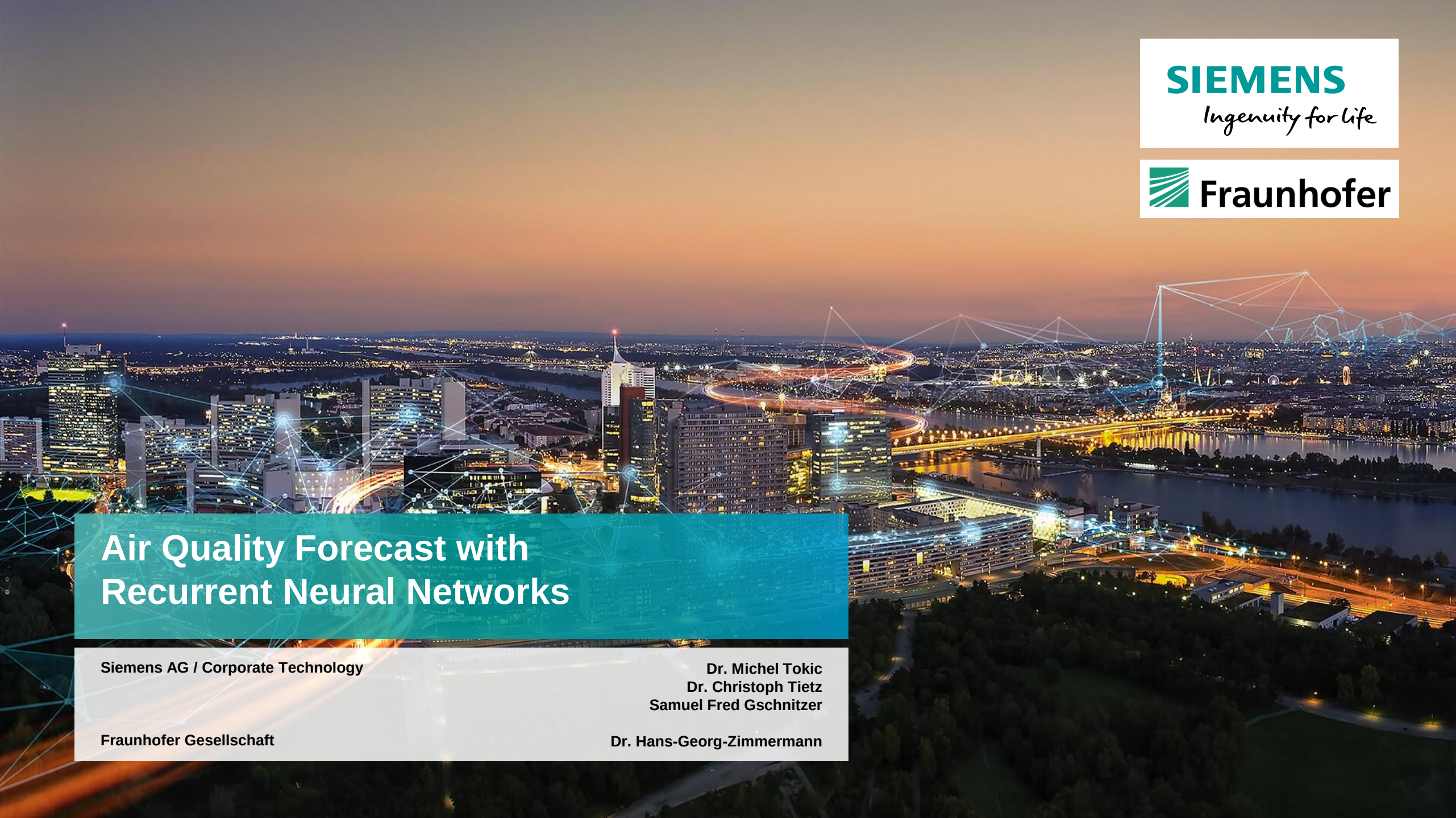


Air Quality Forecast with Recurrent Neural Networks

Siemens AG / Corporate Technology

Dr. Michel Tokic
Dr. Christoph Tietz
Samuel Fred Gschnitzer

Fraunhofer Gesellschaft

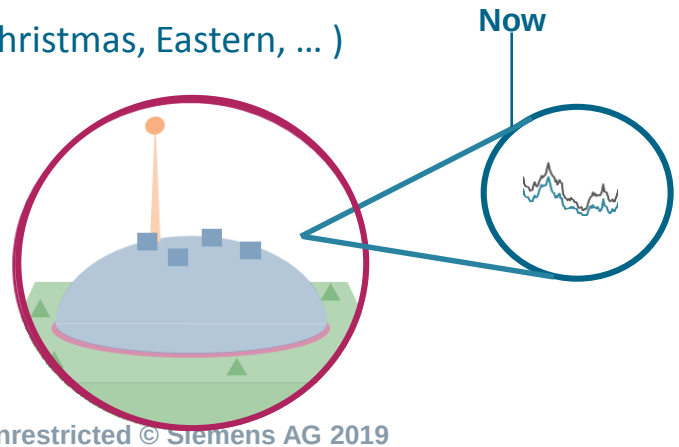
Dr. Hans-Georg-Zimmermann

Methodology behind Air Quality Forecasting

Data input

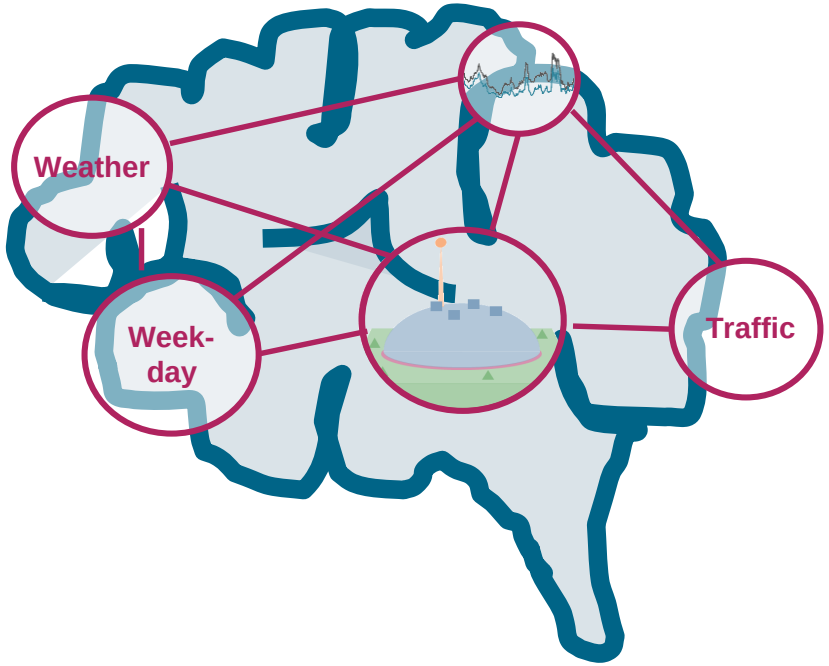
Past 7 days (hourly grid):

- Historic air pollution data
- Weather data
  - Humidity
  - Solar irradiation
  - Cloud cover
  - Temperature
  - Wind speed/direction
- Recurring events (e.g. Holidays, Christmas, Eastern, …)



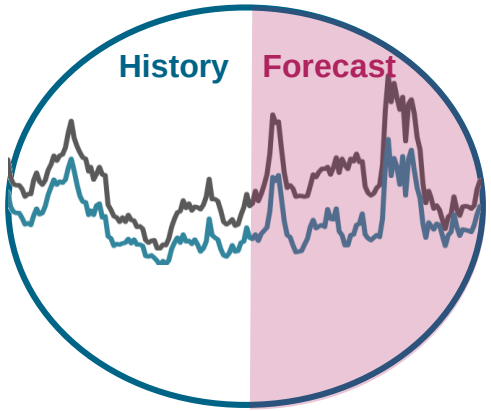
Artificial neural network prediction

- Can predict concentration of air pollutants (PM2.5, PM10 and NO_X)
- High accuracy
- Patterns are averaged values per hour



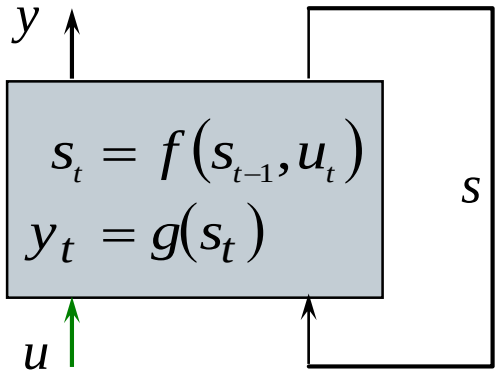
Data Output

- 5-day air quality forecast (hourly grid)
- Short-term mitigation measures → CyAM (City Air Management)



Michel Tokic / Christoph Tietz / Samuel Fred Gschnitzer / Hans-Georg-Zimmermann

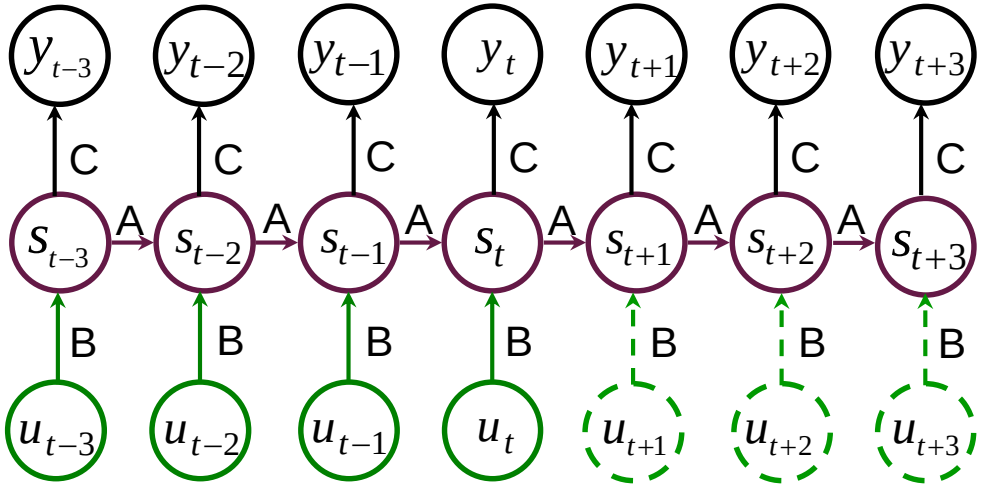
Modelling Open Dynamical Systems using RNNs: Inconsistencies between Past and Future Modeling



$$s_t = \tanh(As_{t-1} + Bu_t)$$ state transition

$$y_t = Cs_t$$ output equation

$$\sum_{t=1}^{T} (y_t - y_t^d)^2 \rightarrow \min_{A,B,C}$$ identification

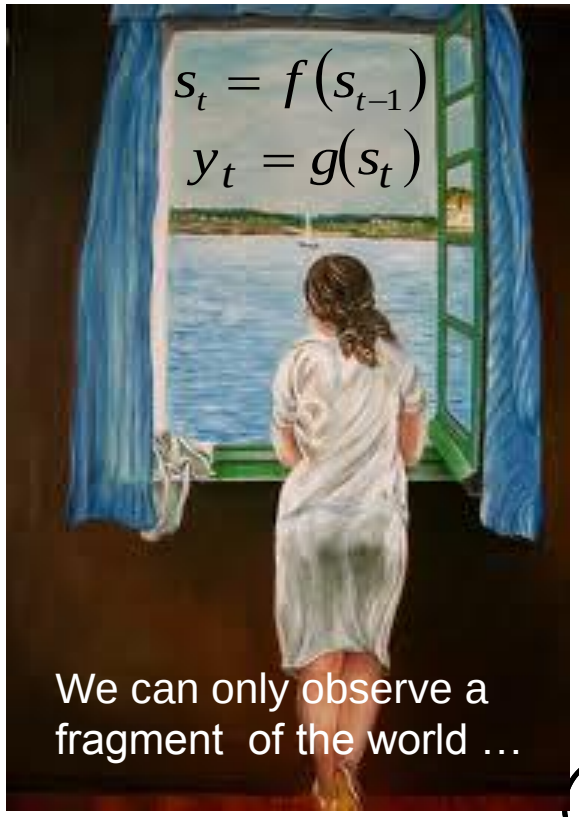


Finite unfolding in time transforms time into a spatial architecture.

The analysis of open systems with RNNs allows a decomposition of the autonomous & external driven part.

Long-term predictability depends on a strong autonomous subsystem (Matrix A)

From Closed Dynamical Systems to Historical Consistent Neural Networks (HCNN)



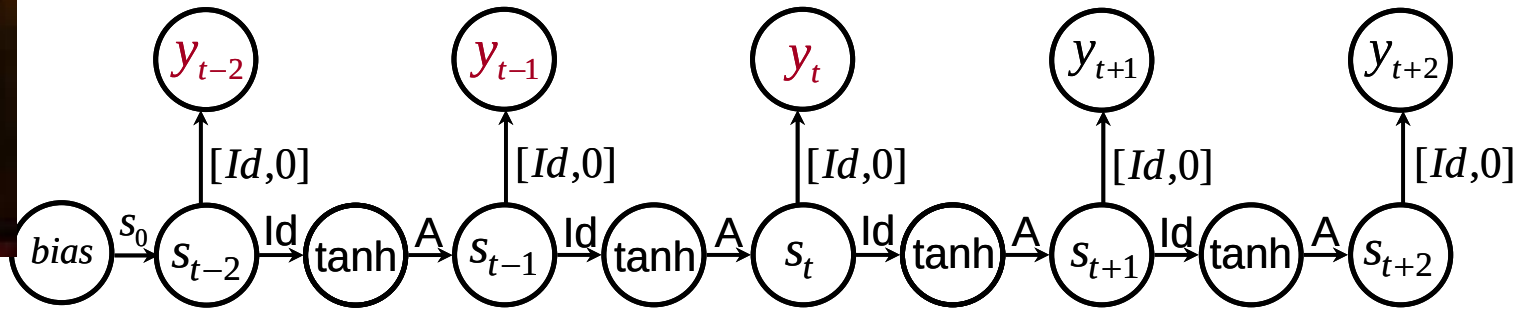
Source: Salvador Dali, 1925

$$s_t = A\tanh(s_{t-1}) \quad , s_0 \qquad \text{state transition}$$

$$y_t = [Id, 0]s_t \qquad \text{output equation}$$

$$\sum_{t=1}^{T}(y_t - y_t^d)^2 \rightarrow \min_{A,s_0} \qquad \text{identification}$$

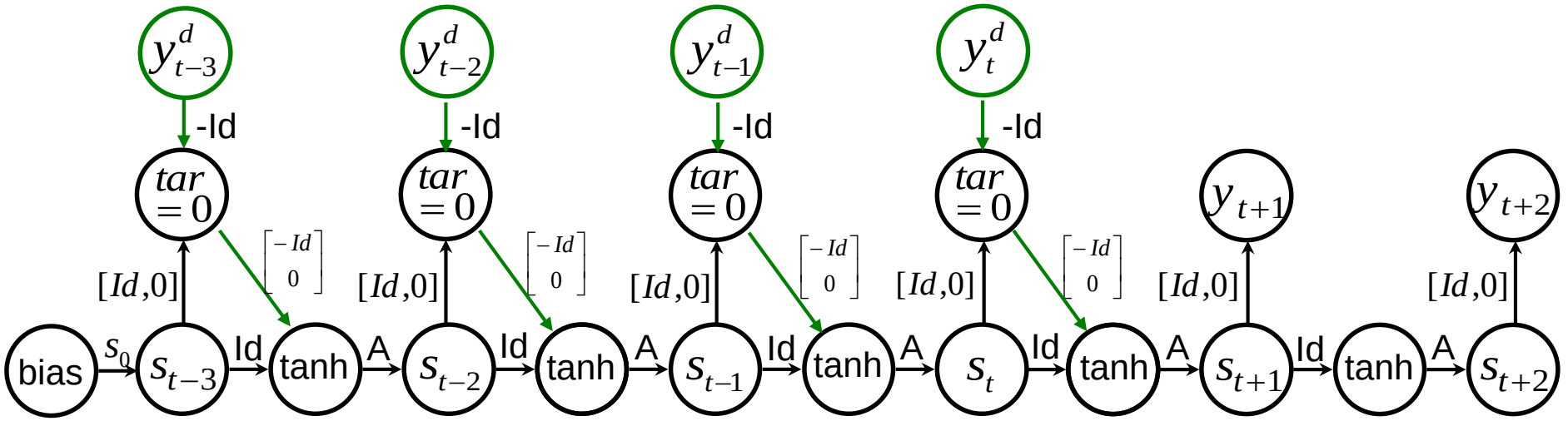
The model is unfolded along **history** → only 1 training example



… but to understand the dynamics of the observables, we have to reconstruct at least a part of the hidden states of the world. → **Forecasting is based on observables and hidden states.**

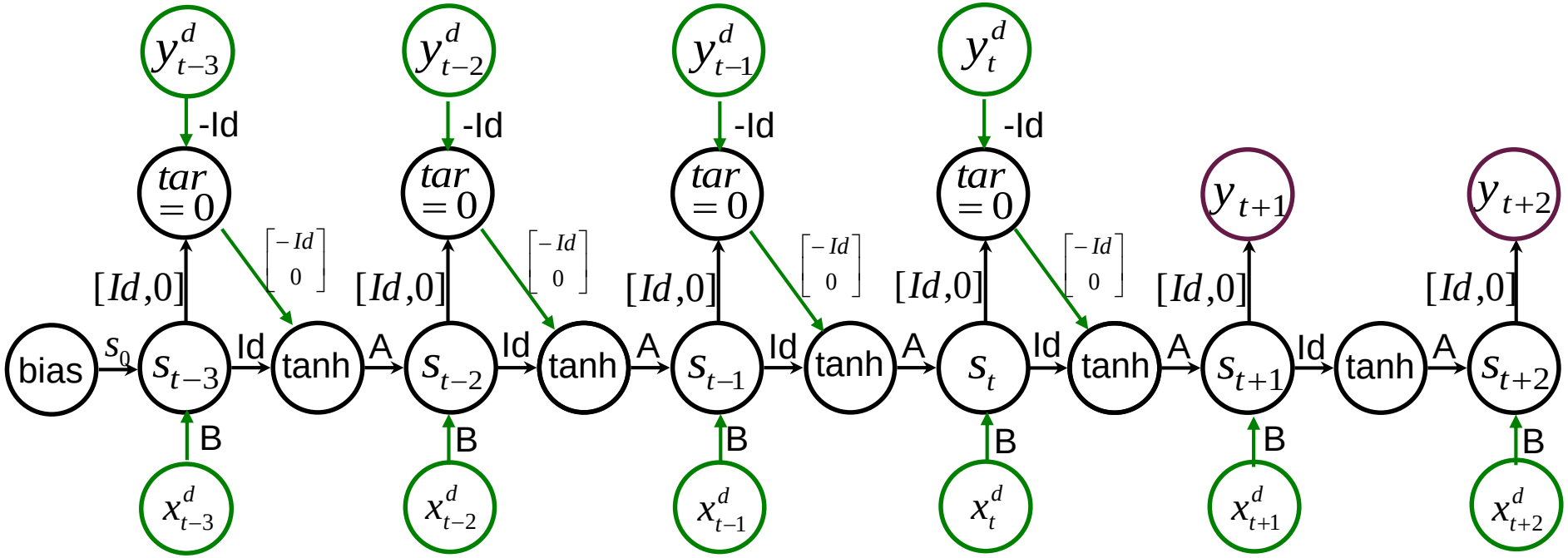
Identification of Dynamical Systems: The Role of Observations

Embed the original architecture into a larger architecture, which is easier to learn. After the training, the extended architecture has to converge to the original model.



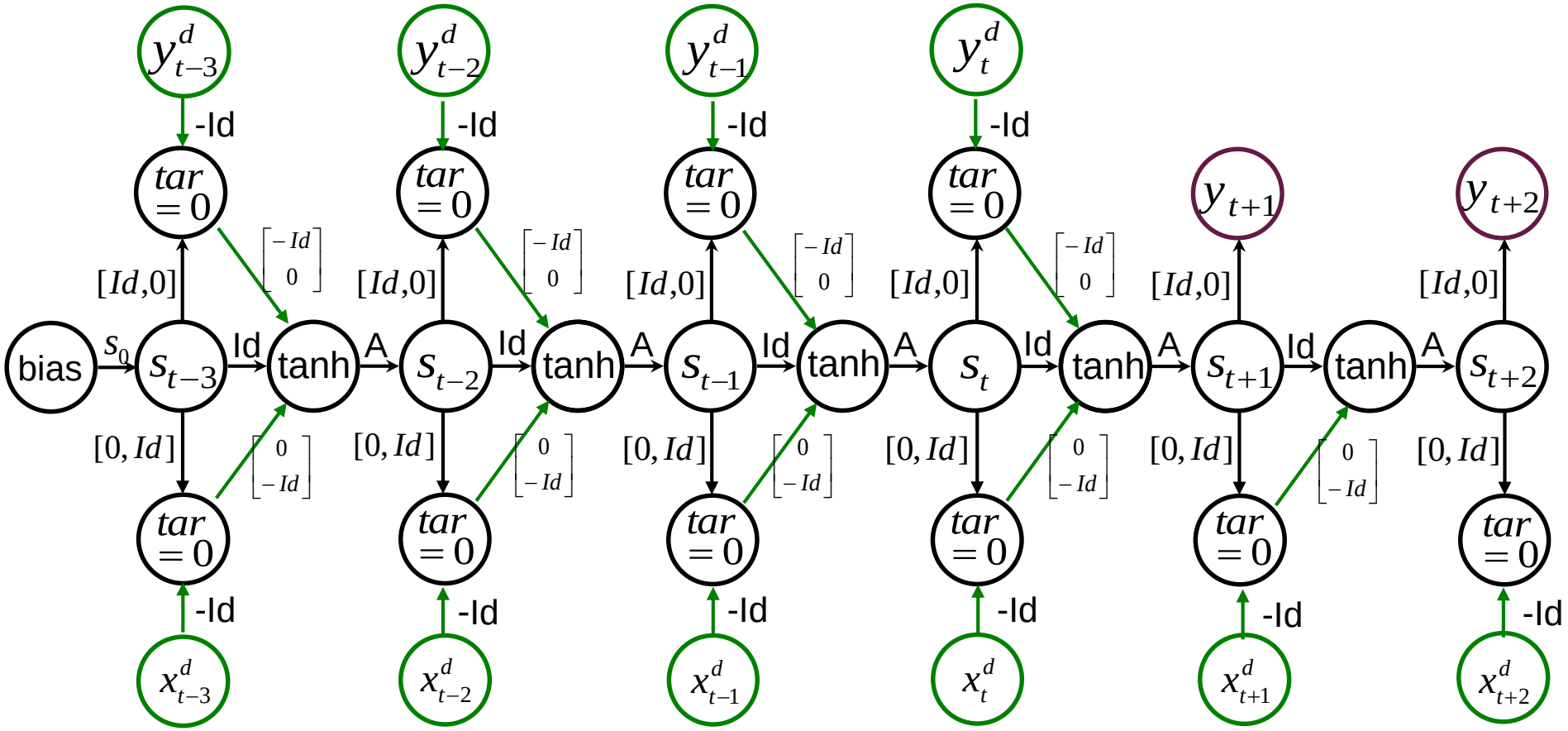
- **The essential task is NOT to reproduce the past observations, but to identify related hidden variables, which make the dynamics of the observables reasonable.**
- Use an architectural teacher-forcing (**ATF**) to support the learning of the HCNN. Replace expectations y_τ by observations y^d_τ: $r^{upper}_\tau = y_\tau - (y_\tau - y^d_\tau) = y^d_\tau$
- The state flow decomposes in 1) known observables, 2) reconstructed hidden variables and 3) unidentifiable random variables, which act as a net-internal regularization.

System Identification Including Future Information



Here we assume that we have future information (e.g. weather forecasts) which can be used along the whole unfolding of the network. In this design a second matrix B has to be learned.

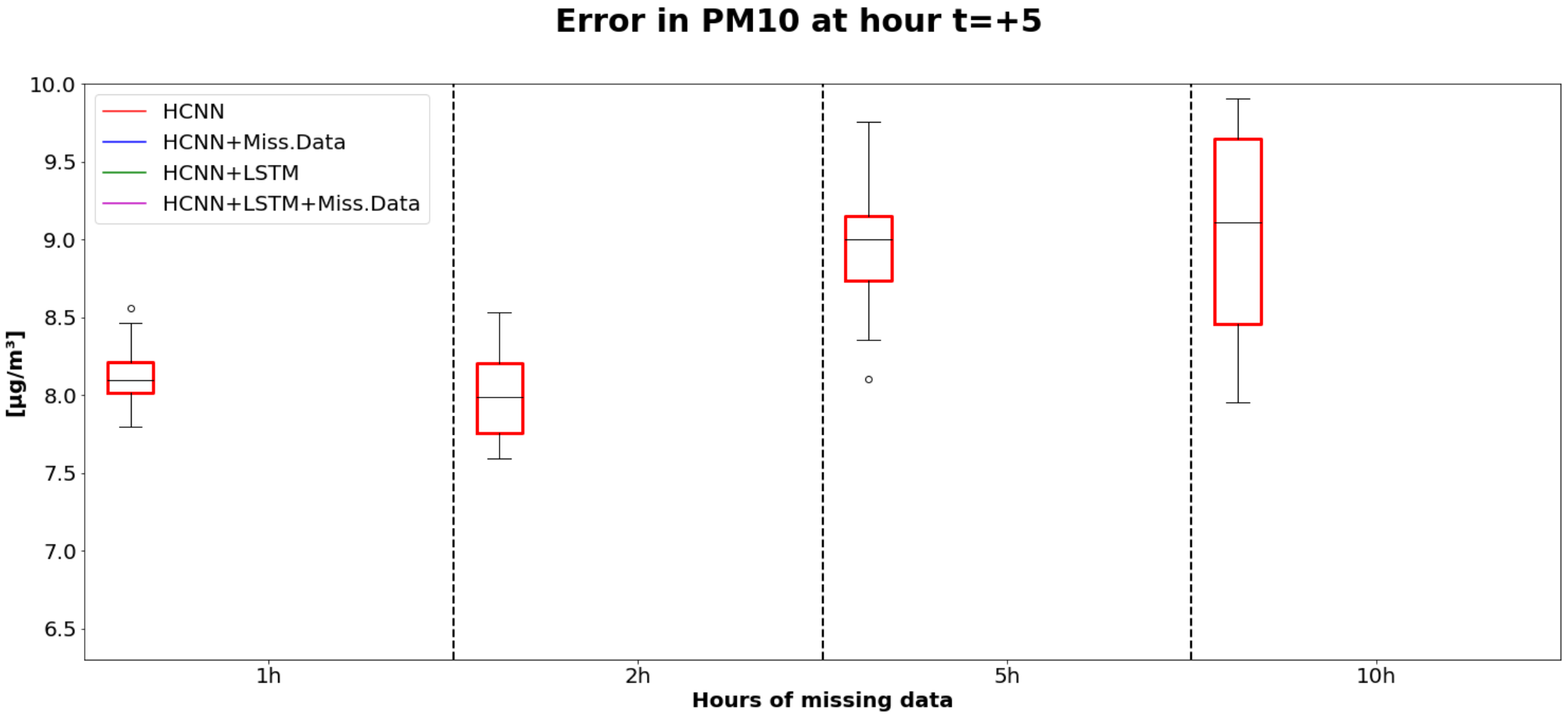
The internalization of Externals



As above the computation of the dynamics always uses the exact measurements of the externals (e.g. weather) but it may be able to reconstruct unobserved hidden variables in the externals x.

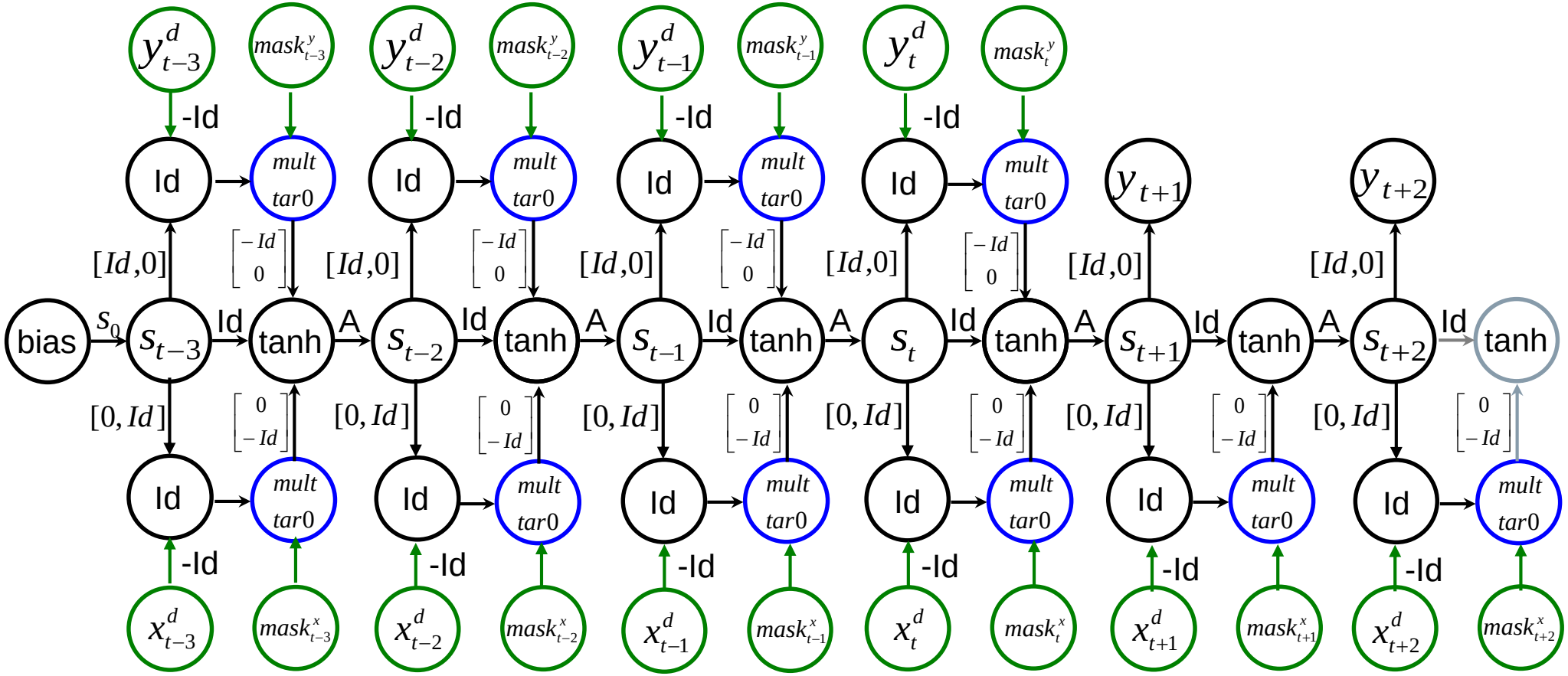
In addition, in the above design missing values of the externals can be corrected by the model.

Results HCNN + Trivial interpolation of missing data

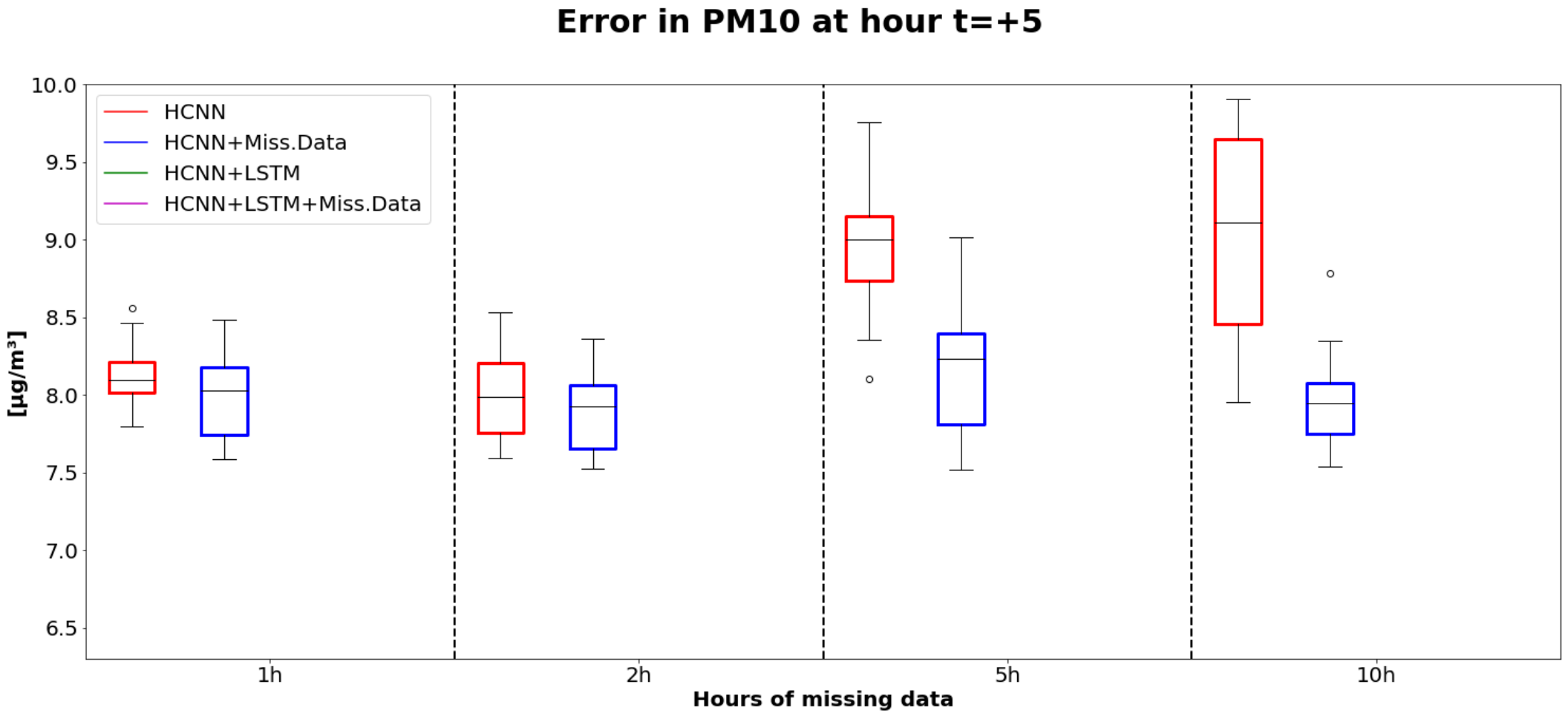


Modeling including Future Information and Missing Information

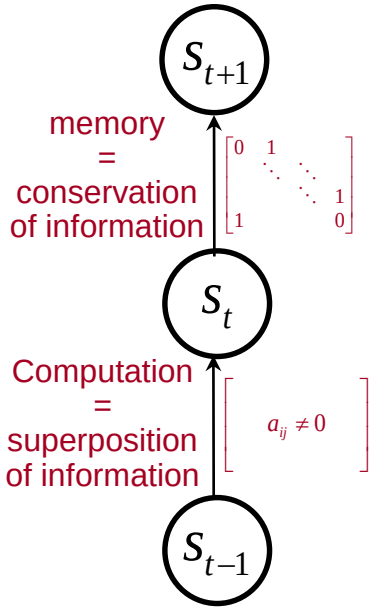
Here for all input vectors we have 0/1-masks, indicating which elements of the inputs are known.

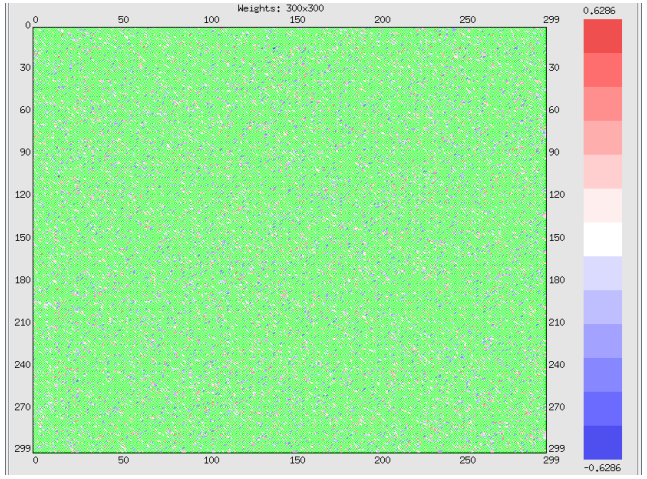


Results HCNN + Model based handling of Missing Data



Function & Structure: Long Memory, Different Time Scales & Sparse Neural Nets





To avoid numerical explosions, large recurrent systems cannot be fully connected.

Backpropagation is able to learn systems with around 50 nonzero weights per column.

$$sparsity \approx \frac{50}{\dim(A)}$$

Structure & Function: **Sparsity is not only a necessary condition for large systems, it describes a tradeoff between memory and computational power.**

A random sparse matrix A allows the modeling of dynamical systems on different time scales.

Different Versions of LSTM Formulations for HCNNs

$$s_t = switch_1(u) \cdot s'_{t-1} + switch_2(u) \cdot A\tanh(s'_{t-1})$$

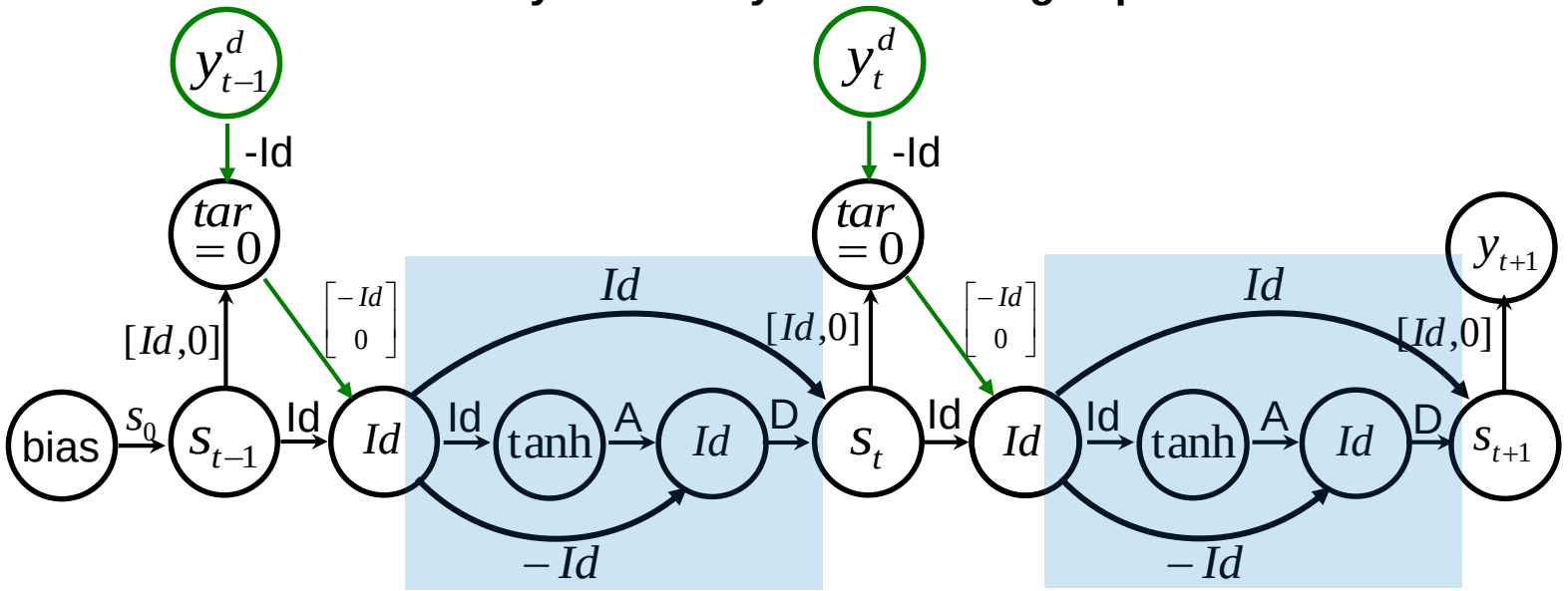
An LSTM Formulation for HCNNs

compare Hochreiter, Schmidhuber: Long short-term memory in Neural Computation, 1997

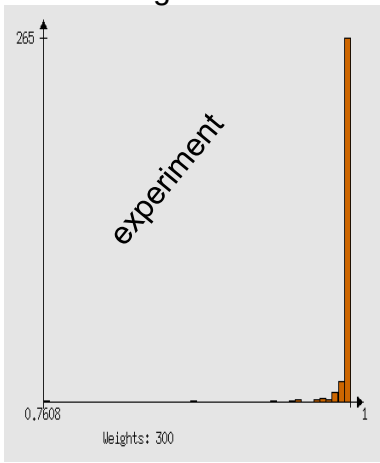
An exponential smoothing embedding of an HCNN with a diagonal matrix D improves long memory analog to LSTM. Choose $0 < D_{ii} \leq 1$, start with $D_{ii} = 1$, notation: $s'_t = Teacher\ Forcing(s_t)$

$$s_t = (1-D)\ s'_{t-1} + DA\tanh(s'_{t-1}) = s'_{t-1} + D(A\tanh(s'_{t-1}) - s'_{t-1})$$

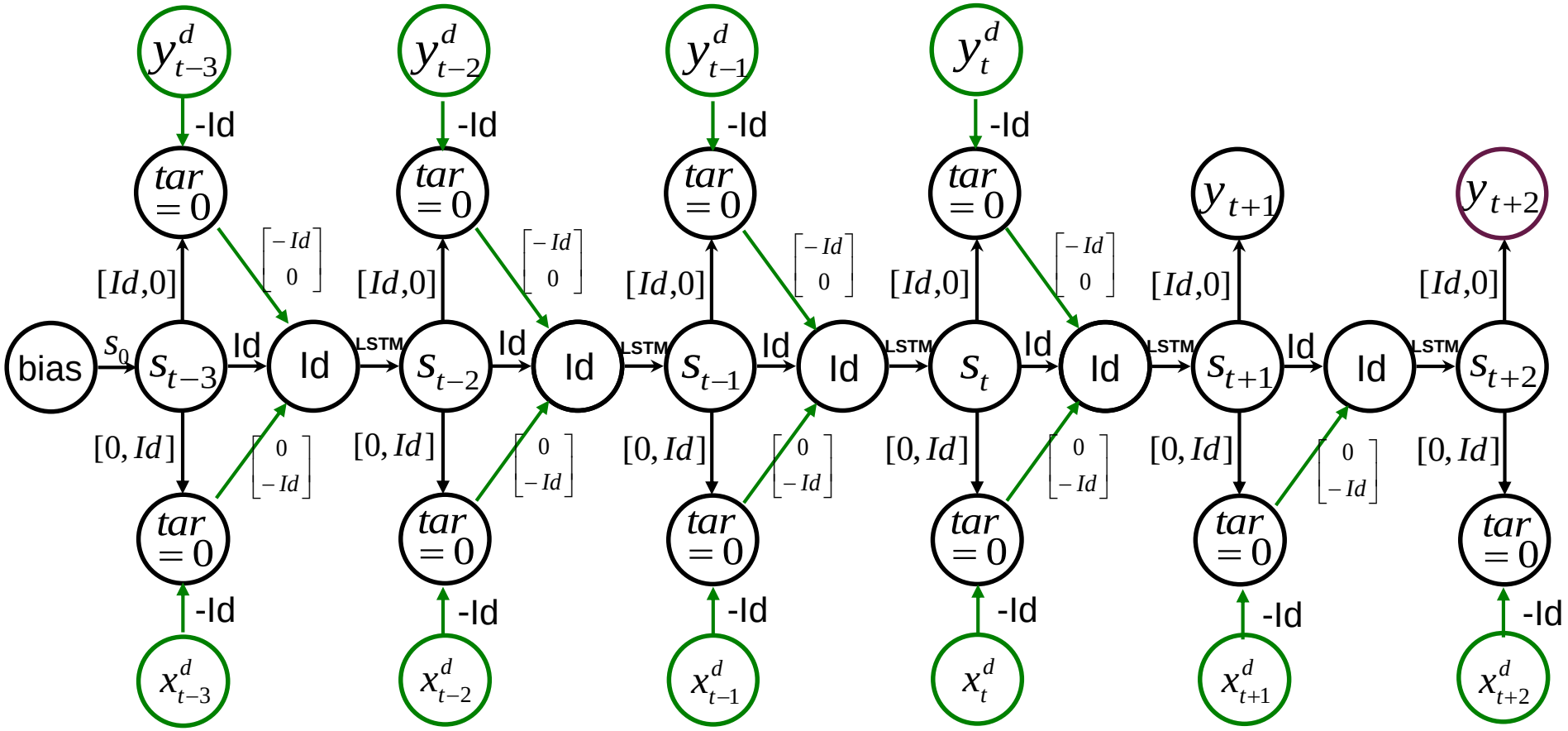
But is this feature really necessary if we use large sparse state transition matrices?



Leave the decision among LSTM and SPARSITY to the learning of a HCNN



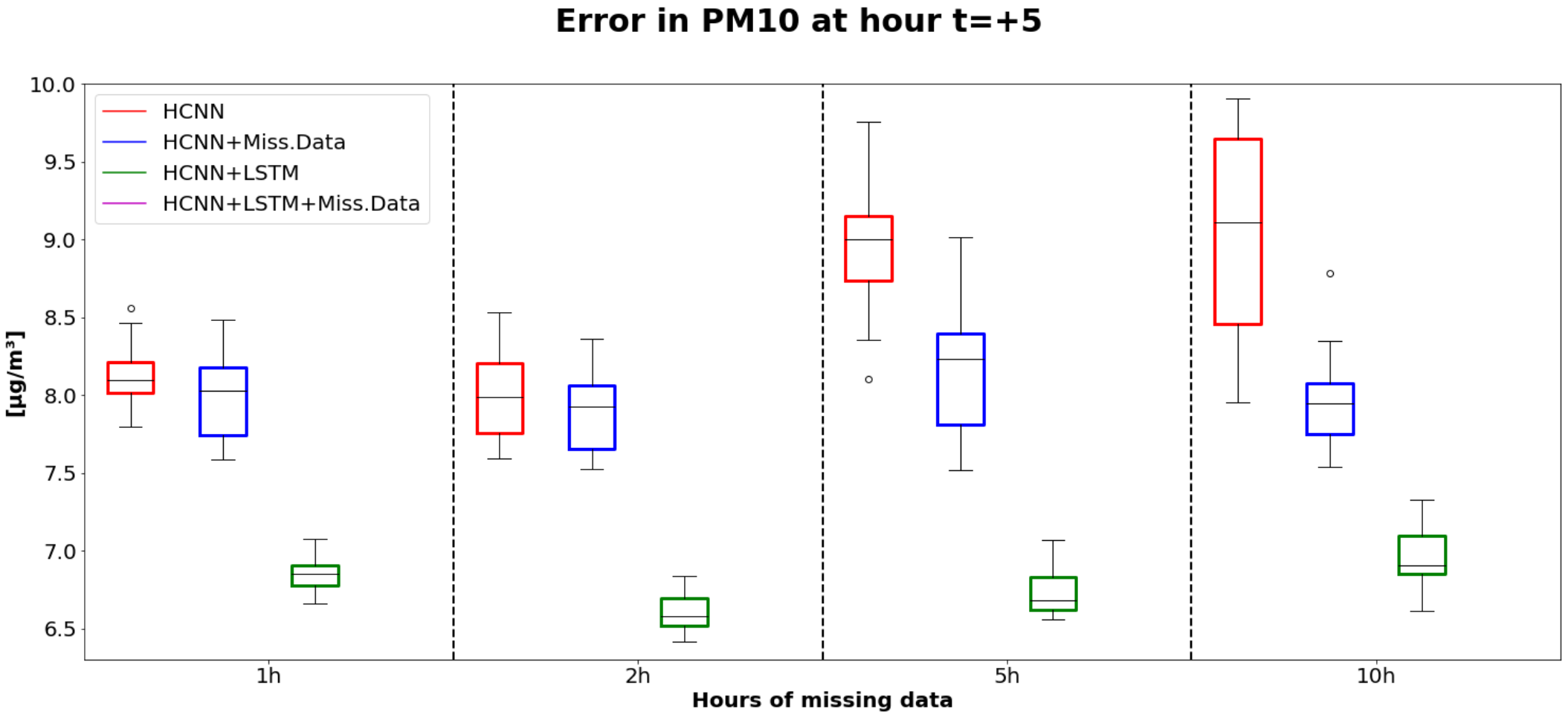
Modeling including Future Infomation and LSTM



As above the computation of the dynamics always uses the exact measurements of the externals (e.g. weather) but it may be able to reconstruct unobserved hidden variables in the externals x.

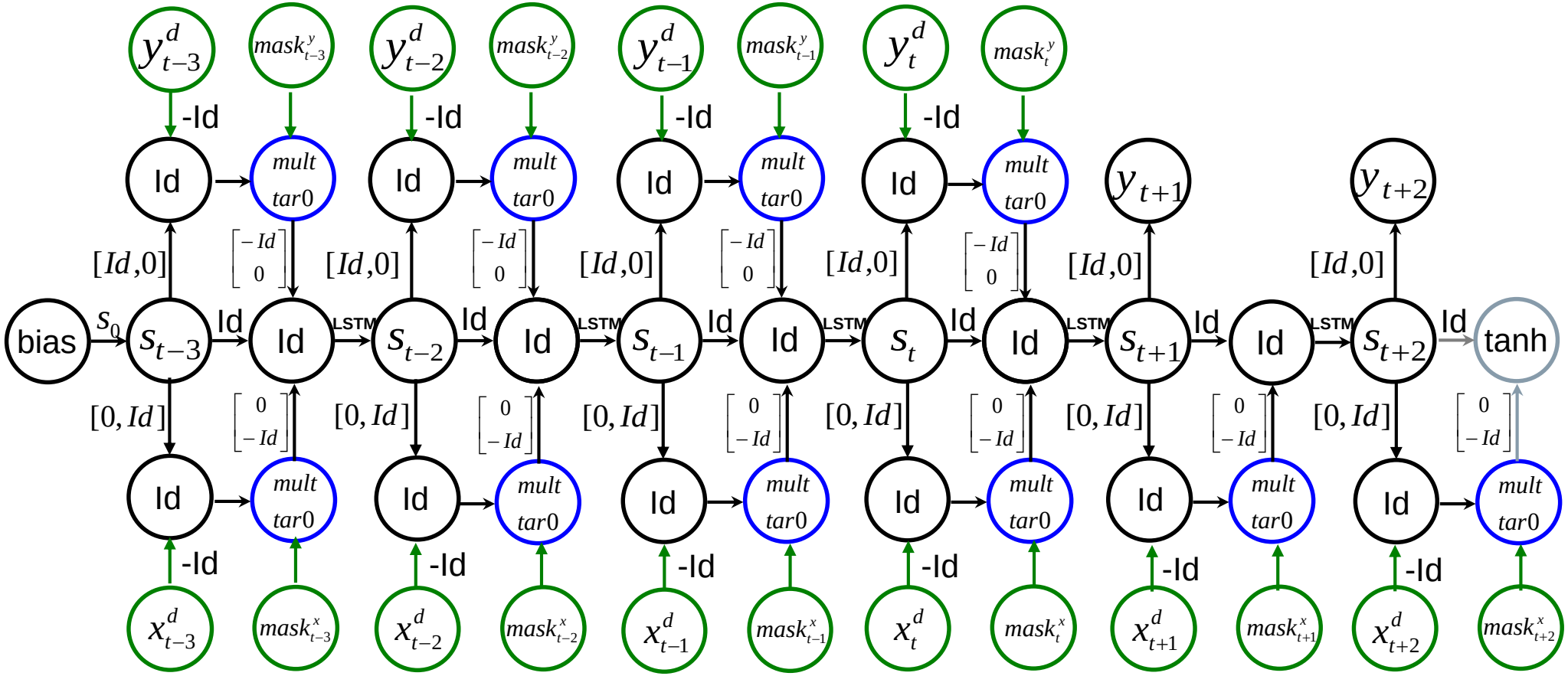
In addition, in the above design missing values of the externals can be corrected by the model.

Results HCNN + LSTM (long-short term memory)

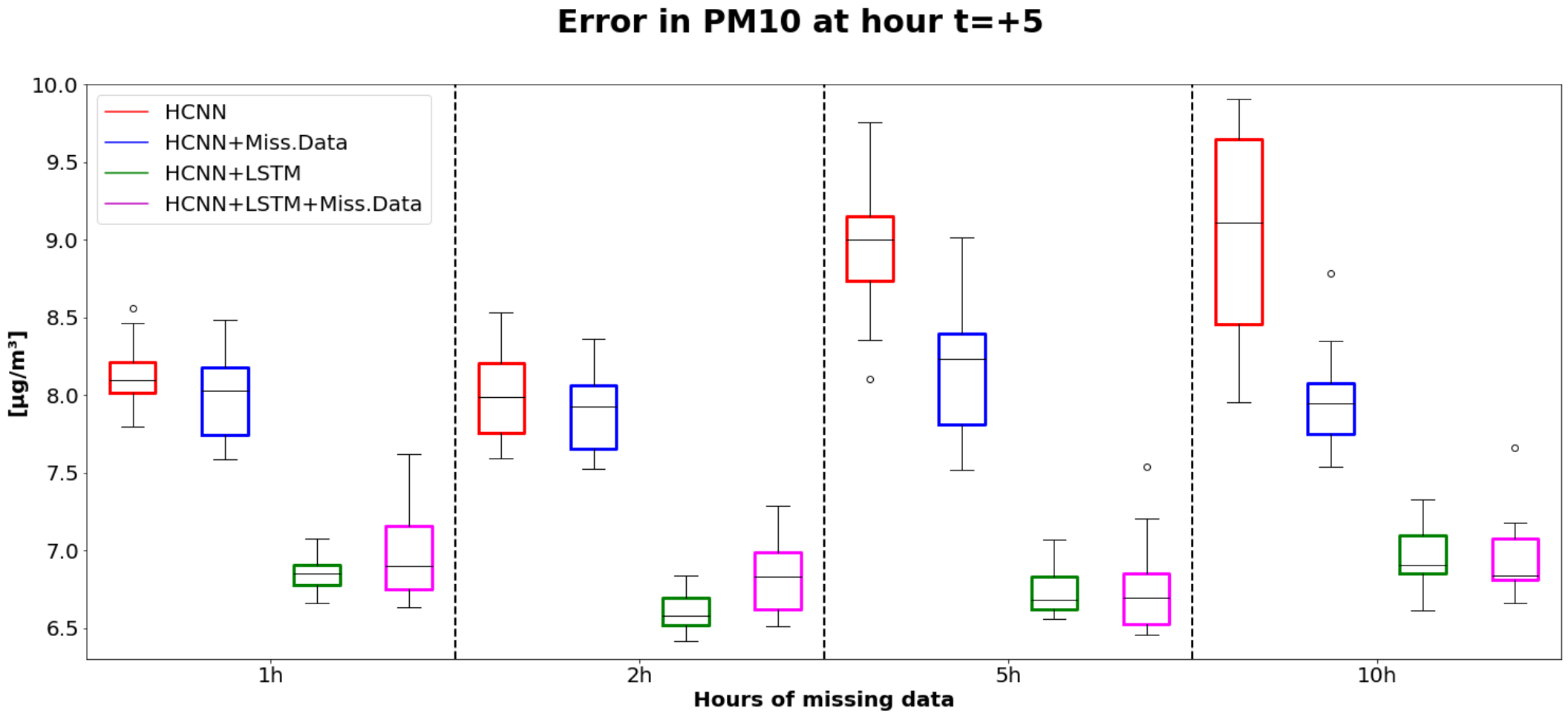


HCNN including Future Information + Missing Data + LSTM

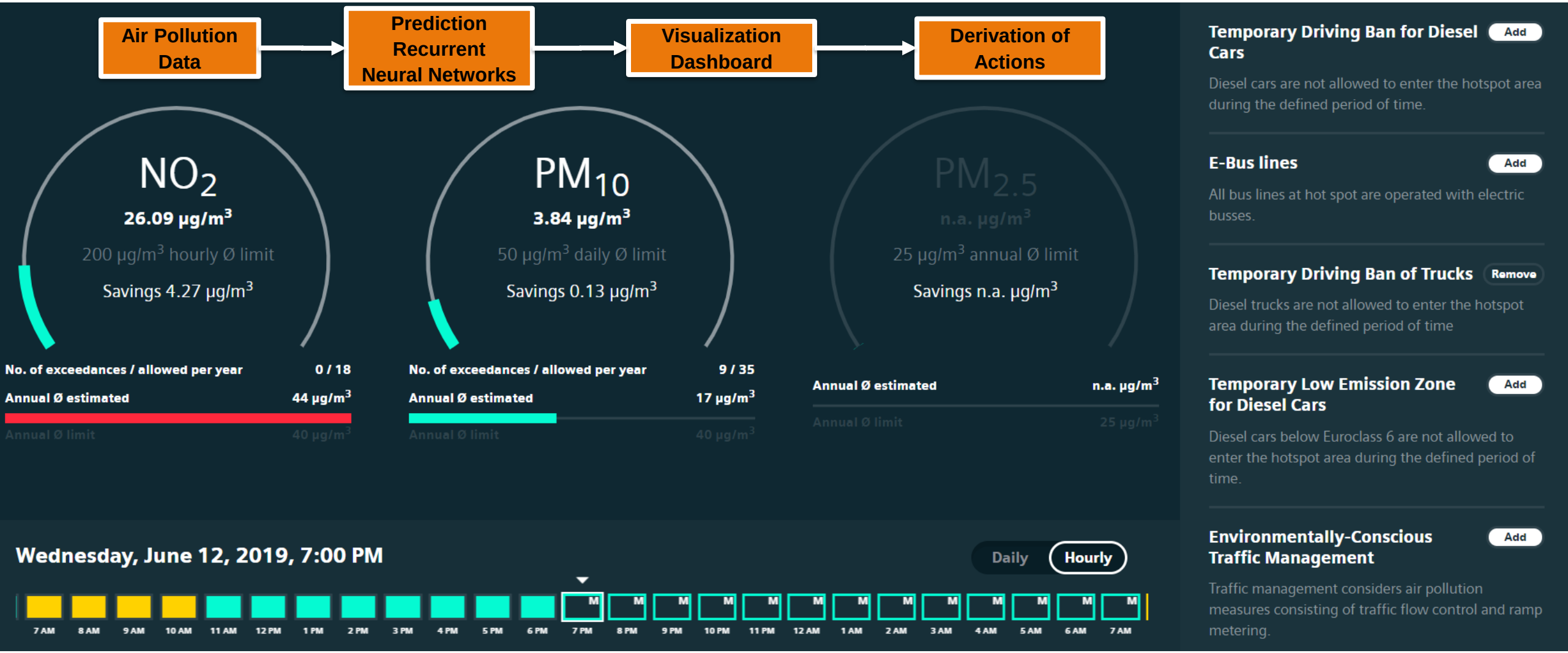
Here for all input vectors we have 0/1-masks, indicating which elements of the inputs are known.



Results HCNN + LSTM + Model based handling of Missing Data



Predictive Analytics can be used for Air Quality Forecast to Improve the Living Quality in Cities (CyAM: City Air Management)



Source: https://cyam.siemens.cloud

Michel Tokic / Christoph Tietz / Samuel Fred Gschnitzer / Hans-Georg-Zimmermann