

# Temporal Ablations (GNN Nowcasting)

Mohammadmehdi Rajabpourshirazy

December 18, 2025

## 1 Temporal Ablations

### 1.1 Motivation

The nowcasting model processes a single time step of node features; no recurrent unit (e.g., GRU/LSTM) is used. Temporal information is therefore provided through engineered *lag features*. A lag of order  $k$  corresponds to the value observed  $k$  time steps earlier: lag1 refers to  $(t - 1)$ , lag2 to  $(t - 2)$ , and lag3 to  $(t - 3)$ . In our dataset, the sensor backlog history is represented by feature channels such as `sensor_backlog_lag1`, `sensor_backlog_lag2`, and `sensor_backlog_lag3`.

### 1.2 Experimental protocol

To isolate temporal effects from sensor placement, we fix the sensor configuration to  $k = 10$  sensors (the best-performing setup in the sensor selection study) and keep the topology, architecture, and hyperparameters unchanged. We create four temporal variants by zeroing subsets of lag channels in the input:

- **No lags:** all lag channels are removed (instantaneous features only).
- **Lag1 only:** lag2 and lag3 are removed, keeping only lag1.
- **Lag1+Lag2:** lag3 is removed, keeping lag1 and lag2.
- **Full (Lag1–3):** all lag channels are kept (default).

Each variant is trained and evaluated over three random seeds; we report mean  $\pm$  standard deviation of test RMSE. We report both *micro* RMSE (computed over all ports and time steps) and *macro* RMSE (average of per-port RMSE).

### 1.3 Results

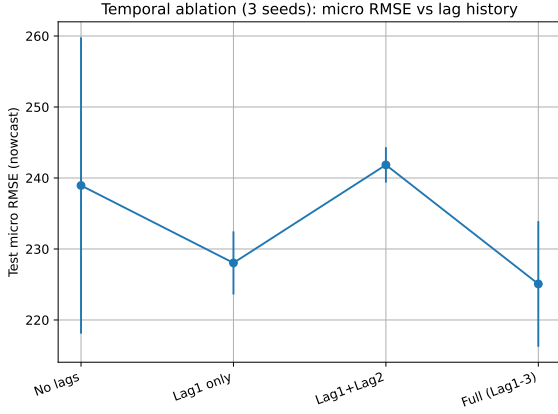
### 1.4 Discussion

The results in Table 1 indicate that temporal history is beneficial on average: the full lag stack (Lag1–3) achieves the best performance in both micro and macro RMSE. Using only Lag1 degrades performance but remains consistently better than removing history entirely, suggesting that immediate temporal context helps stabilize predictions.

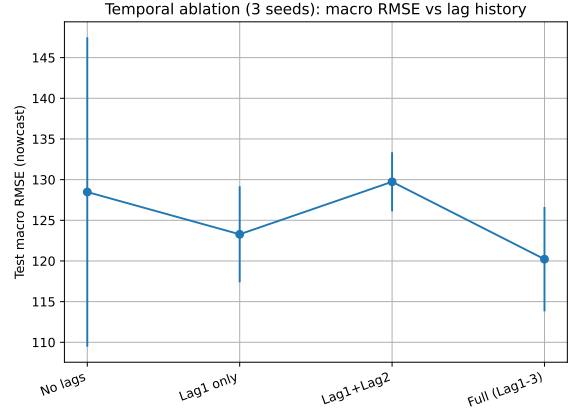
A notable outcome is that the relationship is not monotonic: **Lag1+Lag2 performs worse than Lag1 only**. This can occur because lag channels are treated as additional correlated input features rather than being integrated by an explicit temporal state model. Adding lag2 may introduce stale or noisy information, promote reliance on shortcuts that do not generalize to uninstrumented nodes, or interact unfavorably with regularization (e.g., dropout). In contrast,

Table 1: Temporal ablations for nowcasting with  $k = 10$  sensors fixed. Values are mean  $\pm$  std over 3 seeds. Lower is better.

Temporal context	Test micro RMSE (mean $\pm$ std)	Test macro RMSE (mean $\pm$ std)
No lags (history removed)	238.946 $\pm$ 20.871	128.476 $\pm$ 19.024
Lag1 only	228.038 $\pm$ 4.447	123.279 $\pm$ 5.908
Lag1 + Lag2	241.846 $\pm$ 2.489	129.735 $\pm$ 3.635
Full (Lag1–3)	<b>225.075 <math>\pm</math> 8.846</b>	<b>120.220 <math>\pm</math> 6.416</b>



(a) Micro RMSE (mean  $\pm$  std) vs. lag history.



(b) Macro RMSE (mean  $\pm$  std) vs. lag history.

Figure 1: Temporal ablations with  $k = 10$  sensors fixed. Error bars reflect variability across 3 training seeds.

the full lag stack provides a more consistent short window, which can help the model capture short-term trends more reliably.

Finally, the **No lags** configuration shows the largest variability across seeds, suggesting that without temporal context the learning problem becomes less stable and more sensitive to optimization randomness.

## 1.5 Visualization