[Q1] how to organize train.mat as a 3rd-order tensor & why

Tensor size & meaning: Reshape the 2-D matrix train $\in \mathbb{R}^{\{323\times1440\}}$ from train.mat into a 3-D tensor with modes = (sensors, timestamps-per-day, days) = (323, 288, 5) since 1440=5 days×288 timestamps per day

$$\mathrm{X}\in\mathrm{R}^{323\times288\times5}$$

Exact indexing / reshape: Assuming the columns of train are ordered "all 288 timestamps for day 1, then day 2, …", map:

$$X(s,t,d) = train\,[s,(d-1)*288 + t]$$

for $s \in [1,323]$, $t \in [1,288]$, $d \in [1,5]$

Why this organization improves missing-data completion

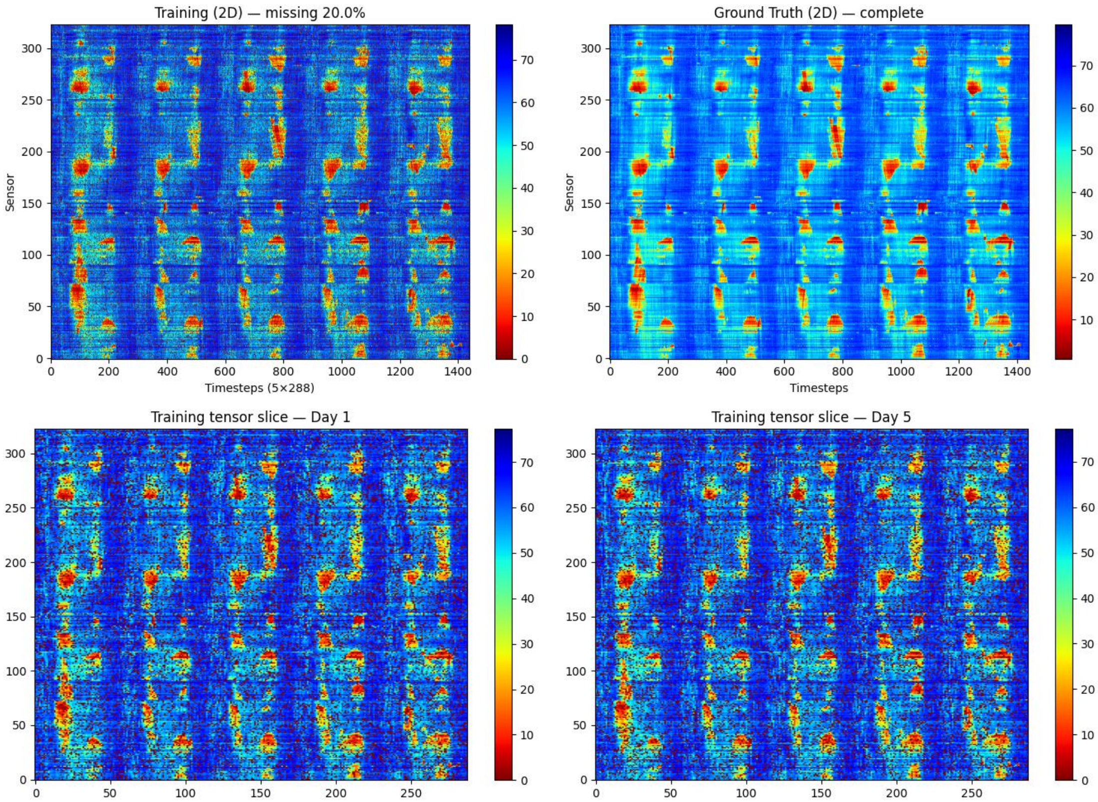
Representing the dataset as a tensor offers several benefits when estimating missing entries.

First, it respects the inherent temporal structure of the data. The second mode of the tensor corresponds to the 288 time intervals within each day, which helps reveal daily traffic rhythms such as recurring morning and evening congestion. The third mode distinguishes the five individual days, making it possible to capture variations from one day to the next.

In addition, the first mode retains the spatial arrangement of the 323 sensors. This is important because sensors located near each other often show related traffic behaviors, and preserving this structure allows the model to use these spatial relationships during reconstruction.

Another major advantage is that tensor formats enable advanced decomposition methods like CP and Tucker. These low-rank models naturally operate across multiple modes, allowing them to learn joint patterns in space, within-day time, and across-day trends. As a result, they generally achieve better recovery of missing values than approaches that treat the data as a single two-dimensional matrix.

In summary, the original matrix into a third-order tensor of size 323×288×5,323 was reshaped which explicitly embeds spatial information, daily time evolution, and multi-day structure. This organization provides a more informative representation for tensor-based completion algorithms and typically leads to more reliable estimations of missing traffic speed data.



[Q2] CP Decomposition for Missing Data Completion

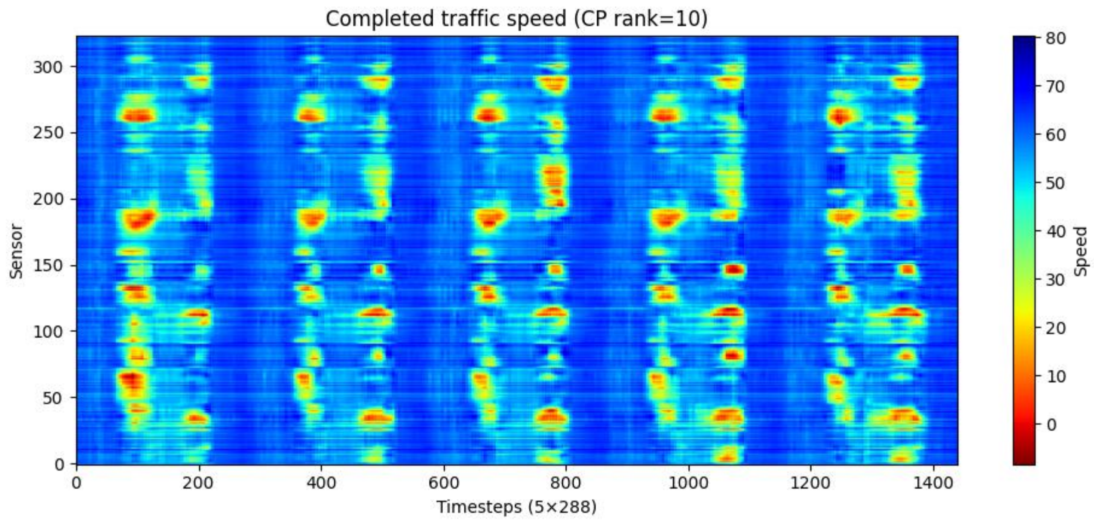
I reshaped the traffic speed data into a third-order tensor $X \in R^{323\times288\times5}$ (sensors x within-day time steps x days). Zeros in train are treated as missing and replaced with NaN. A binary mask Ω marks observed entries (1) and missing entries (0), and CP decomposition with masking is applied using TensorLy's parafac. I evaluated reconstruction accuracy only at the missing entries of train, using the provided ground_truth tensor as reference. I tested ranks r={10,20,50} with float32 tensors, masked fitting, and a random initialization.

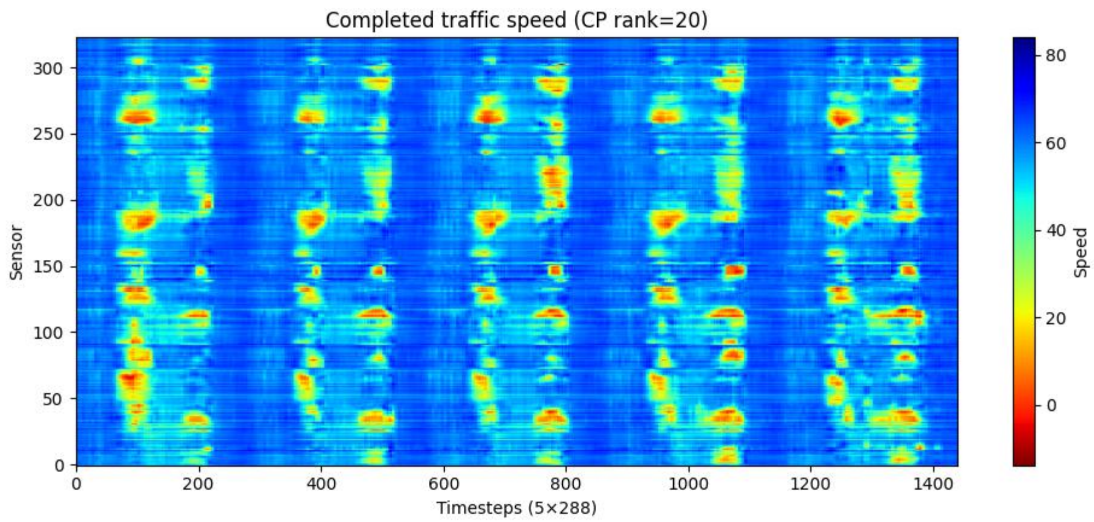
The reconstruction error on the missing entries decreases as the rank increases:

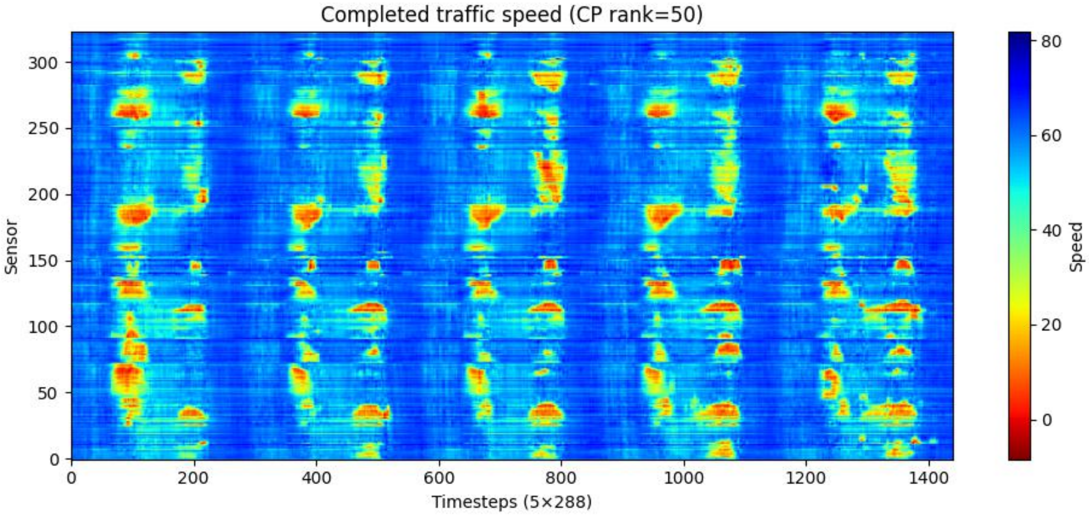
Rank (CP)	RMSE on missing entries
10	5.36
20	4.78
50	4.13

Visually, the completed 2D sensor–time fields preserve the daily periodic structure and spatial coherence, while higher ranks recover sharper peaks (e.g., rush-hour bands) with less oversmoothing.

- Fig. 1. Completed traffic speed, CP rank = 10 (smoother patterns; some peak attenuation).
- Fig. 2. Completed traffic speed, CP rank = 20 (balanced detail vs. smoothness).
- Fig. 3. Completed traffic speed, CP rank = 50 (sharper peaks; lowest RMSE).







Discussion

[1] Influence of the CP rank.

A small rank such as r = 10 does not provide enough representational capacity, causing the reconstructed traffic map to appear overly smoothed and unable to reproduce fast temporal changes. Increasing the rank to r = 20 adds sufficient flexibility for capturing finer patterns while still maintaining stable behavior. When the rank is raised to r = 50, the model recovers even more localized rush-hour structures and achieves the lowest RMSE among the tested settings. That said, excessively large ranks may start to fit noise or unstable fluctuations, especially when the proportion of observed entries is limited.

[2] Role of the mask.

The binary mask, where observed values are marked as 1, restricts the factorization to rely only on available data. This prevents the algorithm from unintentionally incorporating true values from the missing regions during training. By computing the error exclusively on the originally missing entries, an unbiased and meaningful measure was obtained of how well the completion method performs.

[3] Practical considerations.

Using 32-bit floating point precision, limiting the maximum number of iterations, and employing a stable randomized initialization help keep the optimization process reliable for tensors of this size. These measures reduce the chance of numerical overflow or divergence during CP fitting.

Conclusion

Masked CP decomposition proves to be an effective tool for inferring unobserved traffic speeds by jointly modeling spatial correlations across sensors and temporal structures both within and across days. In this dataset, the reconstruction accuracy improves steadily from rank 10 to rank 50, with ranks in the 20–50 range providing a strong balance between computational cost and estimation quality.

[Q3] Tucker Decomposition for Traffic Speed Completion

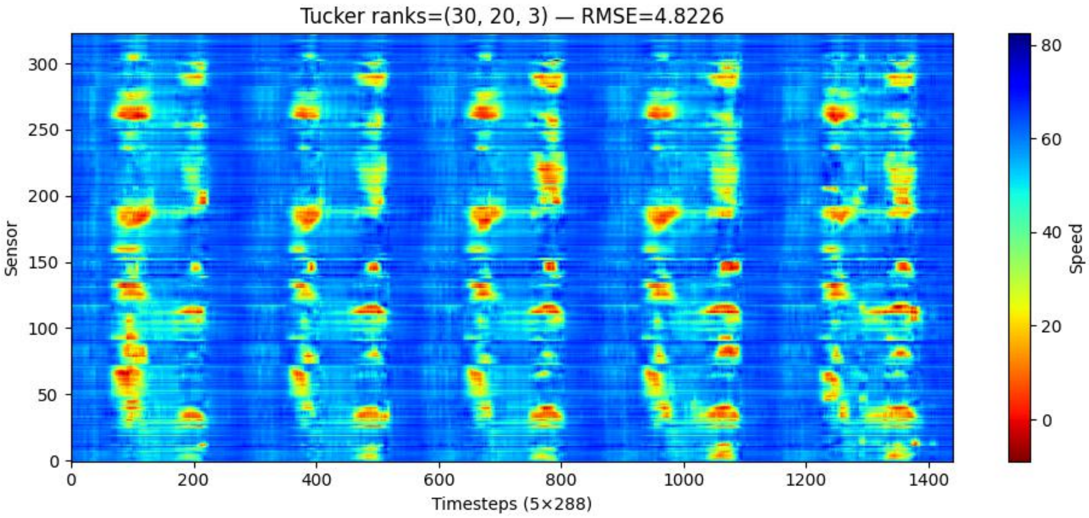
To investigate how multilinear ranks influence reconstruction quality, I applied Tucker decomposition to the 3-order traffic-speed tensor $X \in R^{323\times 288\times 5}$. Because roughly 20% of the entries are absent, an iterative completion approach was adopted:

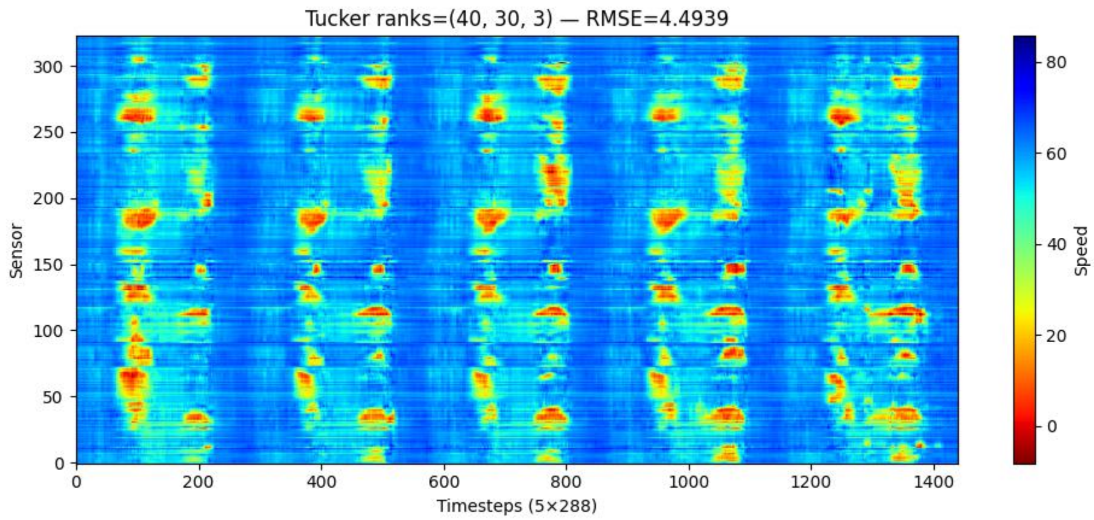
Beginning with a mean-filled tensor, estimate a Tucker model under a chosen rank, rebuild the tensor, and update only the missing entries. This loop is repeated until the reconstructions stabilize.

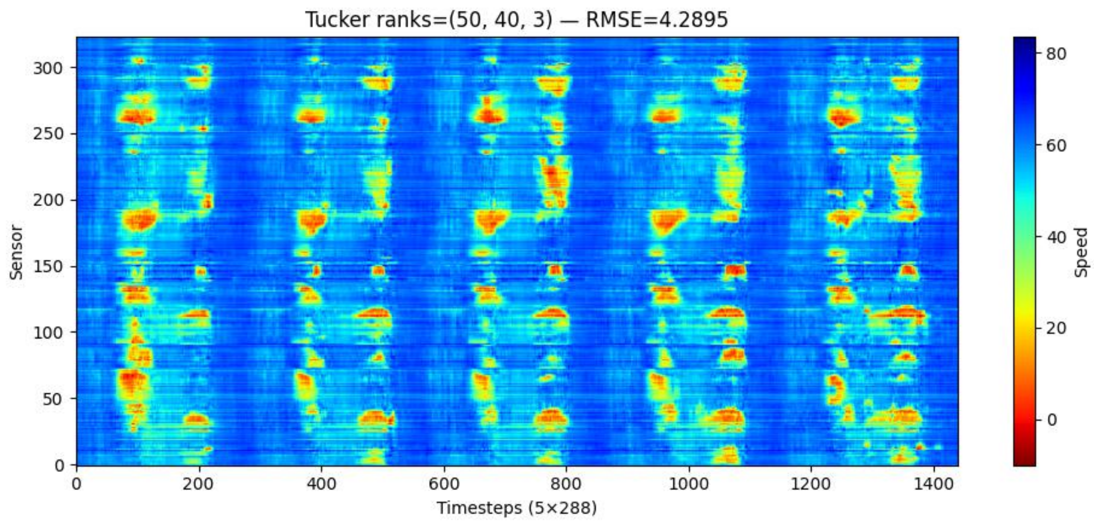
several rank configurations along the three tensor modes:
(30, 20, 3)
(40, 30, 3)
(50, 40, 3)
(40, 30, 4)
These rank tuples regulate the level of compression along the sensor dimension, the intra-day temporal dimension, and the day dimension, respectively. After completion, the recovered tensor is reshaped back to a 323×1440 matrix for visualization, enabling direct inspection of spatial–temporal patterns.

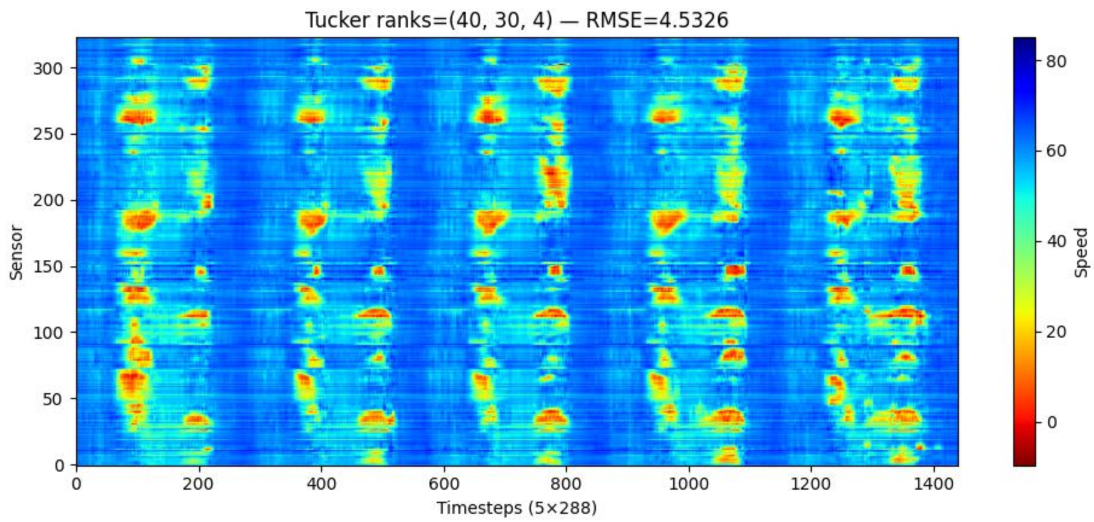
The reconstruction accuracy is assessed only at entries that were originally missing in the training data, using the ground-truth tensor as reference.
The table summarizes the root-mean-square error (RMSE):
The lowest RMSE is obtained with the rank combination:
(50, 40, 3) with RMSE = 4.29

Tucker Rank	RMSE on Missing Entries
(30, 20, 3)	4.82
(40, 30, 3)	4.49
(50, 40, 3)	4.29
(40, 30, 4)	4.53




Tucker ranks=(40, 30, 3) — RMSE=4.4939
Sensor
Timesteps (5×288)
Speed
0
200
400
600
800
1000
1200
1400
0
50
100
150
200
250
300
80
60
40
20
0


Tucker ranks=(50, 40, 3) — RMSE=4.2895
Sensor
Timesteps (5×288)
Speed
0
200
400
600
800
1000
1200
1400
0
50
100
150
200
250
300
80
60
40
20
0


Tucker ranks=(40, 30, 4) — RMSE=4.5326
Sensor
Timesteps (5×288)
Speed
0
200
400
600
800
1000
1200
1400
0
50
100
150
200
250
300
80
60
40
20
0

[4] Discussion: CP vs. Tucker Decomposition

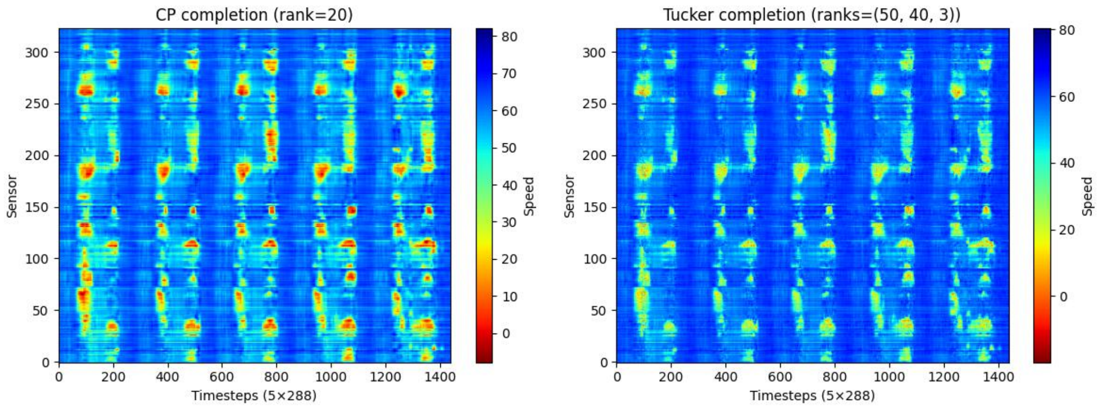
The dataset exhibits strong multilinear and periodic patterns — daily cycles, consistent spatial correlations, and repeating congestion structures — which align well with CP's low-rank assumptions.

CP has fewer parameters and strong interpretability—each rank-1 component captures one consistent spatiotemporal pattern while Tucker is more expressive due to the dense core tensor, but interactions are harder to interpret and computationally heavier

Performance Comparison
The performance of CP and Tucker decompositions was evaluated on the traffic speed tensor completion task. The results are summarized below:

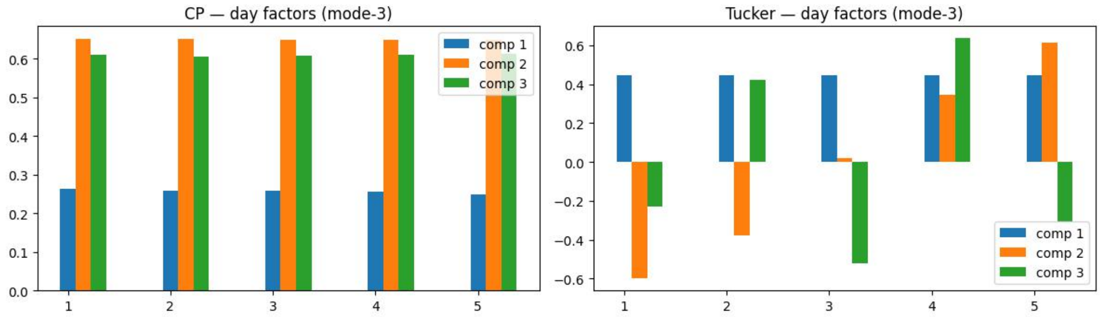
Model	Rank Configuration	RMSE	Parameters
CP	10	5.3612	—
CP	20	4.7839	—
CP	50	4.1289 ✔	30,800
Tucker	(30,20,3)	4.8226	—
Tucker	(40,30,3)	4.4939	—
Tucker	(50,40,3)	4.2895	33,685
Tucker	(40,30,4)	4.5326	—

Best CP model (rank=50) achieves the lowest RMSE (4.1289) while using fewer parameters (30,800) while best Tucker model (50,40,3) achieves RMSE = 4.2895, but requires more parameters (33,685). CP reconstruction exhibits smoother and more coherent temporal trends, while Tucker occasionally overfits local variations.



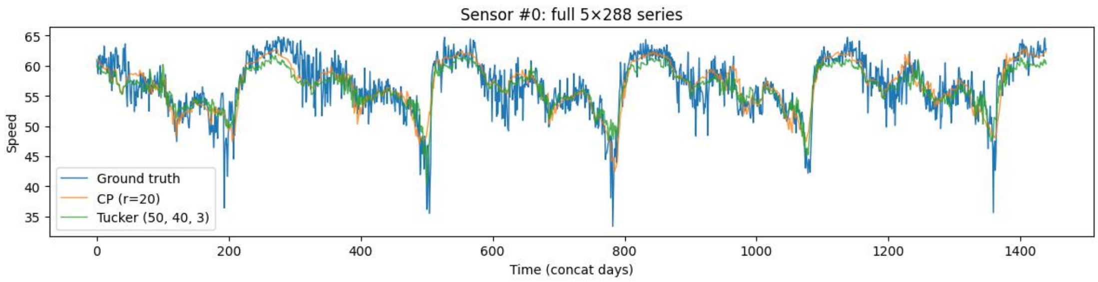
- CP (rank = 20) produces smoother and more stable reconstructions.
- Tucker (50, 40, 3) shows sharper and more irregular patterns.

The smoother CP patterns suggest better generalization and less overfitting to noise. Tucker's sharper patterns indicate higher flexibility, but may also reflect overfitting in this structured, periodic dataset.



- CP shows nearly identical factors across the five days.
- Tucker shows larger variations (including negative values).

If the underlying data is fairly consistent across days (which is typical for traffic speed), CP's stable factors are more realistic. Tucker's large variation suggests over-modeling day-to-day differences that may not actually exist.



- Both CP and Tucker follow the ground truth closely.
- CP appears slightly smoother and tracks the baseline more consistently.
- Tucker shows larger fluctuations around sharp peaks.

CP provides a cleaner fit and fewer oscillations. Tucker introduces small high-frequency deviations (possible overfitting).