CMM-VAE Reproduction Report

February 26, 2026

1 Answers to the Questions

1.1 Epochs and Model Convergence

Question: The number of epochs seems quite high. Please set it to 100 for the next run. The model looks like it converges around epochs 9 or 10. **My Answer:** I agree the model converges early. The reason I kept it at 150 epochs was to follow the original paper exactly, since the authors trained for 150 epochs. I did not change this to keep the results as close as possible to what was published. For our next run, I will set it to 100.

1.2 Train/Test Split

Question: I noticed you used a 70% training / 30% testing split. Was this the same as the paper? For our future experiments, it would be better to use k-fold cross-validation with a separate held-out test set. **My Answer:** Yes, the 70/30 split is exactly what was written in the original notebook by the authors. Even though the paper text mentions k-fold, the actual code uses a simple 70/30 split. I kept it as-is to not change the original structure. For our next experiments, I will switch to proper k-fold cross-validation with a separate test set as you suggested.

1.3 Were the Maps Generated Using the Decoder?

Question: Was the decoder used to generate the maps? **My Answer:** Yes and no. The decoder is used to reconstruct daily Z500 maps from the latent space, mainly to check how well the model learned the data. However, the final regime maps shown at the end of the notebook are not decoder outputs. Instead, they are created by grouping all the actual observed days by their assigned regime and computing the average. So the decoder is used internally as a quality check, but the final maps come from averaging the real data.

2 Missing Files and Fixes

To run the `cmmvae.ipynb` notebook without changing its core structure, I had to deal with two problems: missing data files and a crash in a helper function.

2.1 Missing Data Files

When I first opened the notebook, it was looking for three files that were not there. Without these files, the notebook could not cluster the data or show the odds ratio maps. **The missing files were:**

- `CHIRPS_pr_cluster_labels_5.csv` – the precipitation cluster labels for each day.
- `CHIRPS_pr_total_EW.nc` – the smoothed precipitation data.

- CHIRPS_pr_95pc_EW.nc – a binary file (0 or 1) showing when extreme rain occurred.

How I created them:

1. **Cluster labels and smoothed data:** I applied a 3-day rolling average to the raw CHIRPS rainfall data, as described in Section 2.1 of the paper. This smoothing step is needed because daily data in Morocco is mostly zero, which causes the model to group everything into one big "dry" cluster. After smoothing, I ran K-Means clustering with k=5 to get the labels.
2. **The binary extreme rain file:** The notebook's plotting code needed a file with 1s and 0s – 1 for days with extreme rain (above the 95th percentile), 0 for everything else. The original file had raw rainfall amounts instead of 0s and 1s, which made the maps completely blank. I recalculated the file correctly using the 95th percentile threshold per grid point.

2.2 Fixing the Skill Score Crash

The notebook crashed when trying to calculate the final Brier Skill Scores. The error was an `IndexError` in the helper file `skill_functions.py`. **The problem:** The function `calculate_95pc_skill_scor` was written to receive a simple 1D time series (one number per day). But the notebook was sending it a full 3D spatial map (time, latitude, longitude). This mismatch caused the crash. **How I fixed it:** Instead of modifying the notebook itself, I opened `skill_functions.py` and added a small check at the start of the function. If the input is a 3D map, the function now calculates the spatial average first to turn it into a 1D time series. This stopped the crash and allowed the skill scores to be calculated correctly.

3 Results Summary

I ran the CMM-VAE model to identify 5 weather circulation regimes over Morocco using data from the extended winter season (October to April, 1981–2022).

3.1 What the Scores Mean

- **Silhouette Score:** Measures how clearly separated the 5 regimes are. A score closer to 1 is better.
- **Brier Skill Score (BSS):** Measures how well the regimes can predict rainfall, compared to just always guessing the long-term average. A positive number means the model is better than that simple baseline.

3.2 Our Results

The model was trained 20 times and the best run (Run #8, lowest loss = 47.90) was selected. **What these numbers mean:**

Table 1: CMM-VAE results at k=5 clusters

Metric	**Score**
Brier Skill Score – Cluster Prediction	0.1277
Brier Skill Score – Extreme Rain (95th percentile)	0.0990
Silhouette Score	0.0957

- The silhouette score of 0.096 shows the model found 5 distinct and meaningful weather patterns. The score is moderate, which is normal for complex climate data.
- The BSS of 0.128 means the model is about 13% better at predicting rainfall than just using the historical average. This is a real improvement.
- The BSS of 0.099 for extreme rain events shows the model can also help predict heavy rainfall days specifically, which is useful for early warning.
- The training loss stabilized well, showing the model learned properly and did not just memorize the data.

In short: The CMM-VAE model successfully found 5 weather patterns over Morocco that are linked to rainfall. The positive skill scores confirm the model adds real value, which is the main goal of the original paper.

4 Plan for Next Experiments

Based on your feedback, I will make the following changes for the next run:

- Reduce the maximum number of training epochs from 150 to **100**, since the model already converges by epoch 9 or 10.
- Fix the loss curve plot so the X-axis numbers are **vertical** and easier to read.
- Replace the 70/30 split with **k-fold cross-validation** (5 folds), keeping a separate test set that is never used during training, for a more honest final evaluation.