Author’s Response of third review of Assessing seasonal climate predictability using a deep learning application: NN4CAST

February 18, 2026.

Detailed Comments
Responses are marked in blue.

Response to Reviewer 4

I like to thank authors for considering comments. However, based on responses, I am still not convinced with ACC and RMSE terminology. As of response, it seems to be more like a pattern correlation rather than ACC? And if this is pattern correlation, please specify the domain, and use appropriate term in figures and in text.

We thank the reviewer for this comment and for encouraging us to clarify the terminology. We understand the source of confusion. In the climate prediction literature, to our knowledge, the anomaly correlation coefficient (ACC) is commonly used to analyze temporal or spatial (pattern) correlation between predicted and observed anomaly fields. Therefore, what we compute in the time series panels corresponds to the pattern correlation of anomaly fields evaluated year-by-year over the specified domain. To avoid any ambiguity, we have revised the figures (Figs. 2 and 4) captions to explicitly state this. In particular, we now clarify that each yearly value represents the spatial anomaly (pattern) correlation between predicted and observed anomaly fields over the full domain. In the first example this domain corresponds to the TNA, while for the second one to the European continent.

Response to Editor

Please address the additional comments raised by the reviewer regarding the evaluation. In addition, based on my own assessment, the current model description is not sufficiently clear and remains overly configuration-driven (i.e., largely presented as a list of tunable hyperparameters). While the manuscript explains the general encoder–decoder concept (compressing predictors into a latent representation and reconstructing the target) and notes that the implementation is a fully connected network, key implementation details required for reproducibility are still underspecified. In particular, please clarify the exact input/output tensor shapes and data vectorization, and provide a concrete description of how optional components, such as convolutional layers, skip connections, batch normalization, and dropout, are implemented beyond being listed as hyperparameters. For a GMD-style model description, a compact layer-by-layer specification (including dimensions) and a schematic of the default computation graph would substantially improve clarity and reproducibility.

We thank the Editor for the request and agree that greater implementation detail improves reproducibility. We have refined the model description to enhance clarity and reproducibility, according to the comment.

Regarding the spatial correlations, we have explicitly addressed the treatment of spatial correlations by detailing how the predictor and predictand fields are vectorized and reshaped within the network.

With respect to implementation details, we have introduced a dedicated subsection in the methodology that provides a comprehensive layer-by-layer specification of the architecture, including input and output tensor dimensions, as well as the implementation and role of optional components such as convolutional layers, skip connections, batch normalization, and dropout. To complement this detailed description, we have also included a schematic diagram illustrating the model's computation graph and the flow of data through the various layers. We attach to this response this new subsection:

"To ensure full reproducibility, we provide a layer-by-layer description of the default architecture used in the Pacific–European teleconnection experiment (Sect. 4.2), corresponding to the hyperparameter configuration in Table 3. For this configuration, num_conv_layers = 0 and no skip connections are used; thus, the model reduces to a fully connected encoder–decoder network. If convolutional layers were included, they would use two-dimensional convolutions with unit stride and same padding, preserving the spatial dimensions. For an input tensor of size (H, W), the output would retain the same spatial resolution, (H, W), while the channel dimension would become equal to the number of filters F, yielding a tensor of shape (H, W, F). After flattening, this corresponds to a vector of size H × W × F. For each hidden dense layer, the following sequence of operations is applied: Dense → Batch Normalization → ELU activation. Batch normalization, kernel regularizer and dropout are included only when use_batch_norm=True, use_initializer=True and use_dropout=True, respectively. When enabled, kernel regularization and dropout are applied immediately after the flattening layer. In the present configuration, a dropout rate of 0.1 is used. The output layer employs a linear activation function and does not include dropout. The resulting output vector is reshaped to the predictand field size. Model parameters are optimized by minimizing the mean squared error (MSE) using the Adam optimizer with a learning rate of 10−4. A schematic representation of the architecture is shown in Figure 2.”