Addressing Deep Learning Model Uncertainty in Long-Range Climate Forecasting with Late Fusion

Ken C. L. Wong[1], Hongzhi Wang[1], Etienne E. Vos[2], Bianca Zadrozny[3], Campbell D. Watson[4], Tanveer Syeda-Mahmood[1]

[1]IBM Research, San Jose, CA, USA
[2]IBM Research, Johannesburg, GP, South Africa
[3]IBM Research, Rio De Janeiro, RJ, Brazil
[4]IBM Research, Yorktown Heights, NY, USA

Contact: clwong@us.ibm.com

Motivation

- Long range climate forecasting can save lives and property
  - High impact extreme events, e.g., heat waves, cold fronts, floods, droughts can result in tremendous loss of lives and property
  - The longer the range of accurate forecasting, the more the time for preparation and response

- Deep learning models
  - Become more popular on climate forecasting
  - Model uncertainties – models trained with identical hyperparameters are usually different
  - Model uncertainties can be more prominent with limited climate data
  - Reduce reliability especially with long-range forecasting

- Goal
  - Reduce deep learning model uncertainties and improve accuracy in seasonal forecasting

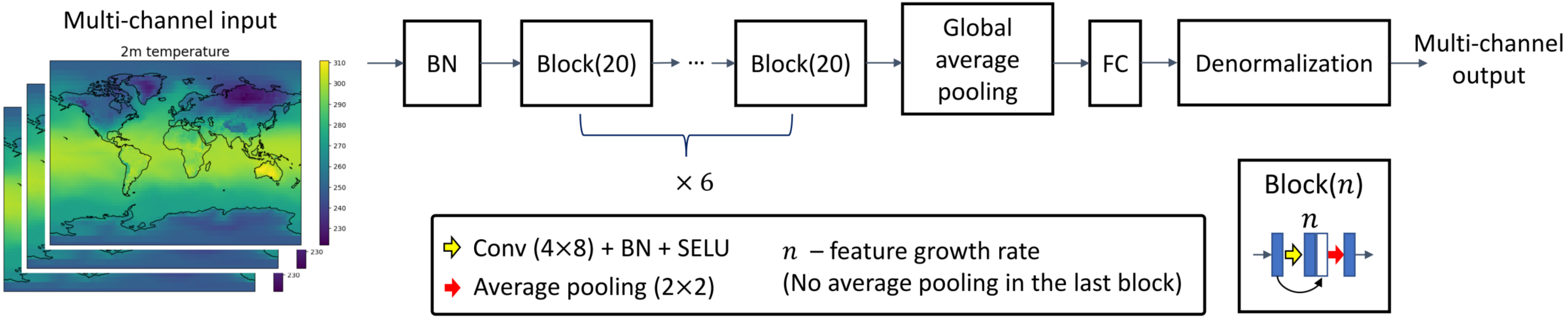
Contributions

- Propose a network architecture for 2m temperature prediction
  - Denormalization layer – provides the benefits of data normalization without normalizing the data

- Propose a late fusion approach that systematically combines the predictions from multiple models to reduce expected errors of the fused results

Network Architecture for 2m Temperature Forecasting

- Convolutional neural network
  - Multi-channel input tensor formed by stacking the maps of 2m temperature of a fixed input horizon
  - Multi-channel output, each channel contains the 2m temperature of a location at a fixed lead time
    - E.g., 8 locations → 8 channels (scalars)
  - Six dense blocks, each with one convolutional layer with 20 filters
  - Batch normalization (BN) layer as the first layer for input data normalization
  - Denormalization layer as the last layer: $x_o(c) = x_i(c)\sigma(c) + m(c)$
    - The fully connected (FC) layer only needs to provide normalized prediction → same advantage as normalizing observed data



Late Fusion

- Late fusion

Combines predictions from different models to **reduce expected errors from all models**:

$$f(s_i) = \sum_j w^j f^j(s_i) \quad \text{with} \quad \sum_j w^j = 1 \qquad \textbf{(1)}$$

where $f^j(s_i)$ is the prediction by the jth model of input s_i. The pairwise correlation between models j_1 and j_2 is:

$$M[j_1, j_2] = \sum_i \left[f^{j_1}(s_i) - t(s_i)\right]\left[f^{j_2}(s_i) - t(s_i)\right] \qquad \textbf{(2)}$$

with $t(s_i)$ the true value. The weights are then computed (from the validation data) by:

$$\mathbf{w} = \arg\min_{\mathbf{w}} \mathbf{w}^{\mathrm{T}}\mathbf{M}\mathbf{w} = \frac{\mathbf{M}^{-1}\mathbf{1}_K}{\mathbf{1}_K^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{1}_K} \qquad \textbf{(3)}$$

with K the number of models to be fused

Experiments

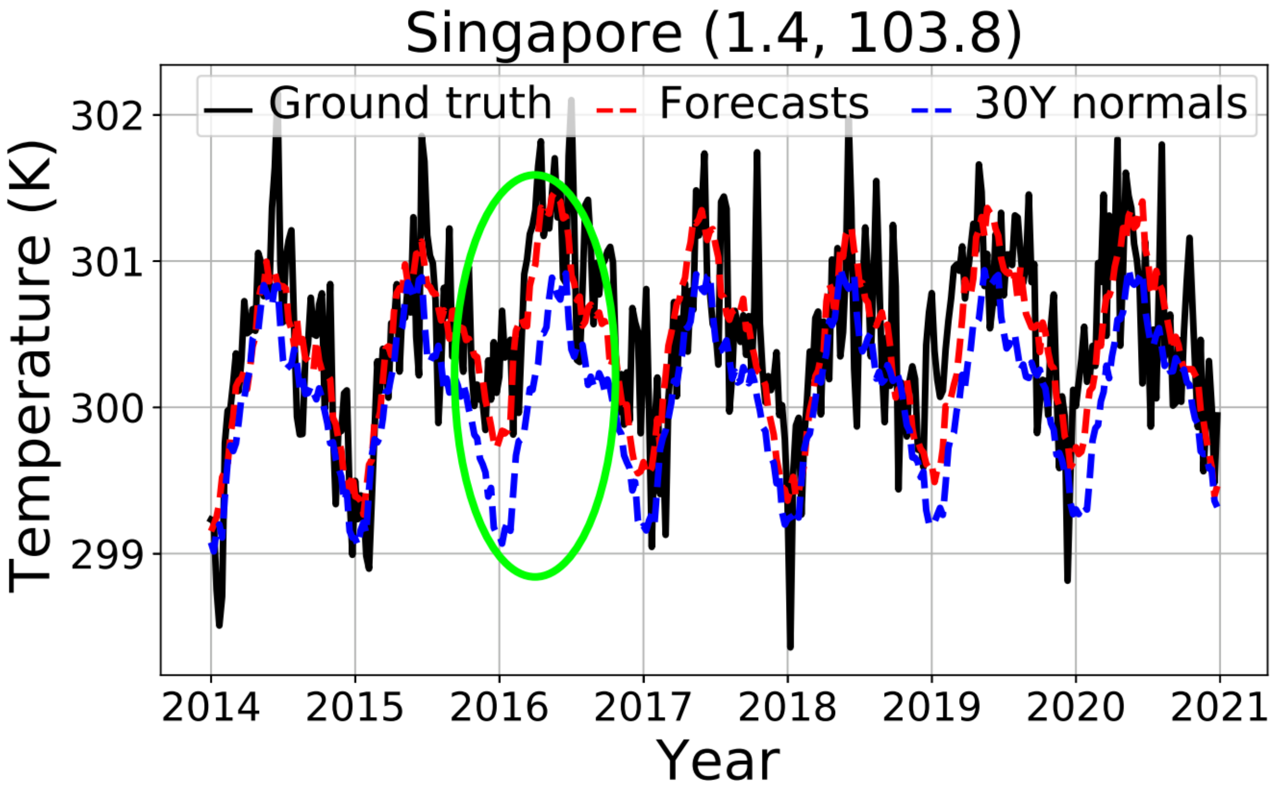
- Data – 2m temperature maps of the ERA5 reanalysis data
  - Training – 1979 – 2007 (1508 weeks)
  - Validation – 2008 – 2011 (208 weeks)
  - Testing – 2012 – 2020 (468 weeks)

- Frameworks
  - Trained 20 models per lead time
  - **Late fusion** – at each lead time, predictions of all models were combined
  - **Best model** – at each lead time, the model with the least validation RMSE was chosen

- Evaluation metric
  - Root mean square error skill score (RMSESS) ∈ $[-\infty, 1]$: $\mathrm{RMSESS} = 1 - \frac{\mathrm{RMSE}_{\mathrm{model}}}{\mathrm{RMSE}_{\mathrm{clim}}}$
  - Compares the model forecasts with the 30-year climate normals (> 0 means the model is better)

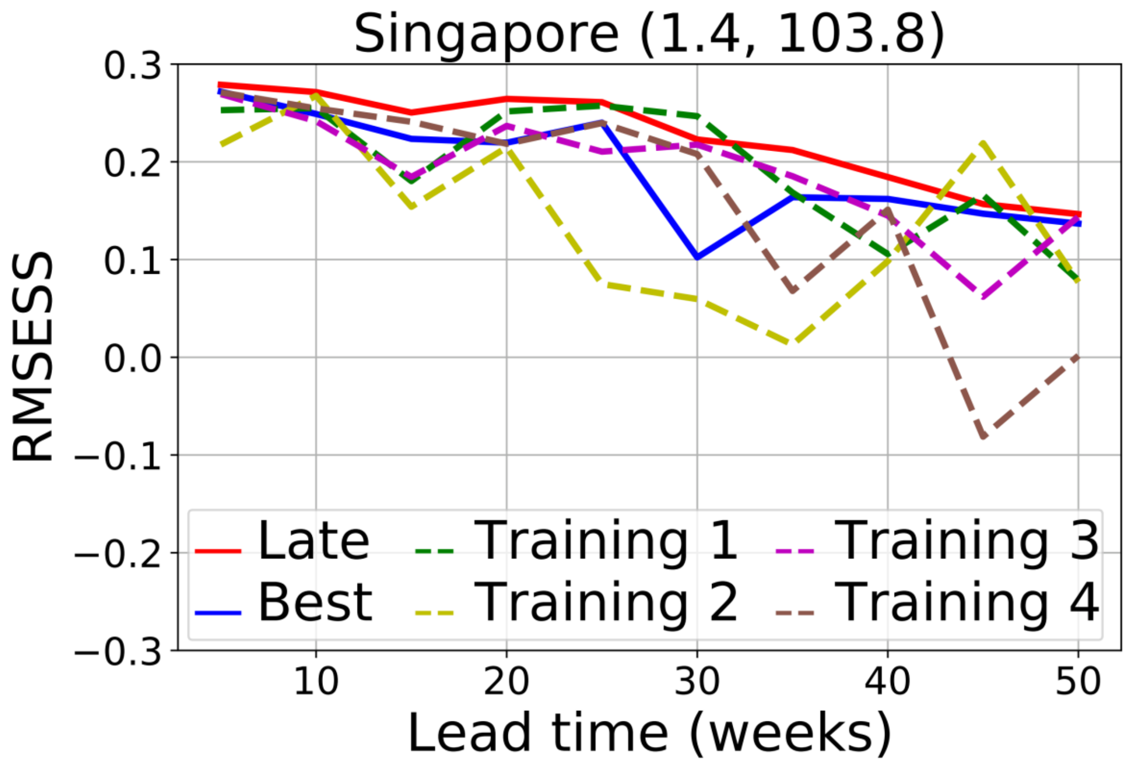
Results

- Example of forecasts of a model on testing data with lead time = 5 weeks
  - Forecasts closely followed the ground truth
  - The model was able to predict the anomaly in 2016 (the hottest year on record)



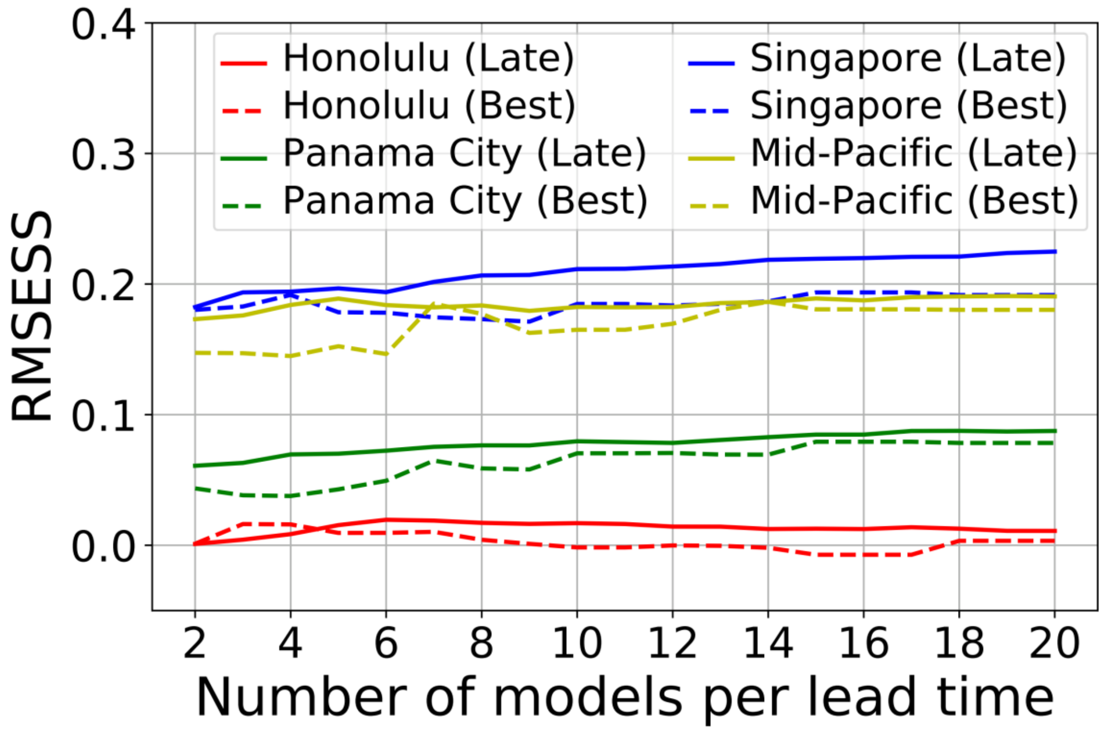
Results

- Forecasts vs lead time
  - Most RMSESS > 0 → the models were better than climate normals
  - Models trained with same hyperparameters are different especially at large lead times
  - The late fusion framework outperformed the best model framework



Results

- Performance with increasing number of models
  - The late fusion framework gradually improved, outperformed the best model



Thanks!

Contact: clwong@us.ibm.com