

ABSTRACT

Title of Document: APPLICATION OF NEURAL NETWORKS TO
EMULATION OF RADIATION
PARAMETERIZATIONS IN GENERAL
CIRCULATION MODELS

Alexei Belochitski, Doctor of Philosophy, 2012

Directed By: Prof. Ferdinand Baer (the academic advisor),
Department of Atmospheric and Oceanic
Science, and Drs. Michael Fox-Rabinovitz and
Vladimir Kranospolsky, Earth System Science
Interdisciplinary Center, University of Maryland,
College Park.

A novel approach based on using neural network (NN) techniques for approximation of physical components of complex environmental systems has been applied and further developed in this dissertation. A new type of a numerical model, a complex *hybrid* environmental model, based on a combination of deterministic and statistical learning model components, has been explored. Conceptual and practical aspects of developing hybrid models have been formalized as a methodology for applications to climate modeling and numerical weather prediction. The approach uses NN as a machine or statistical learning technique to develop *highly accurate and fast emulations* for model physics components/parameterizations. The NN emulations of the most time consuming model physics components, short and long wave

radiation (LWR and SWR) parameterizations have been combined with the remaining deterministic components of a general circulation model (GCM) to constitute a hybrid GCM (HGCM). The parallel GCM and HGCM simulations produce very similar results but HGCM is significantly faster. The high accuracy, which is of a paramount importance for the approach, and a speed-up of model calculations when using NN emulations, open the opportunity for model improvement. It includes using extended NN ensembles and/or more frequent calculations of full model radiation resulting in an improvement of radiation-cloud interaction, a better consistency with model dynamics and other model physics components.

First, the approach was successfully applied to a moderate resolution (T42L26) *uncoupled* NCAR Community Atmospheric Model driven by climatological SST for a decadal climate simulation mode. Then it has been further developed and subsequently implemented into a *coupled* GCM, the NCEP Climate Forecast System with significantly *higher resolution* (T126L64) and *time dependent* CO_2 and tested for decadal climate simulations, seasonal prediction, and short- to medium term forecasts.

The developed *highly accurate* NN emulations of radiation parameterizations are on average *one to two orders of magnitude faster* than the original radiation parameterizations. The NN approach was extended by introduction of NN ensembles and a compound parameterization with quality control of larger errors.

Applicability of other statistical learning techniques, such as approximate nearest neighbor approximation and random trees, to emulation of model physics has also been explored.

APPLICATION OF NEURAL NETWORKS TO EMULATION OF RADIATION
PARAMETERIZATIONS IN GENERAL CIRCULATION MODELS

By

Alexei Belochitski

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2012

Advisory Committee:

Professor Ferdinand Baer, Chair

Dr. Michael Fox-Rabinovitz, Advisor

Dr. Vladimir Krasnopolsky, Advisor

Professor James Carton

Professor Da-Lin Zhang

Professor Michael Jakobson, Dean's Representative

© Copyright by
Alexei Belochitski
2012

Preface

Dr. Charles Caramello

Dean of the Graduate School 2123 Lee Building

College Park, Maryland 20742

Dear Dr. Caramello,

Mr. Alexei Belochitski's dissertation presented to the dissertation committee at the Department of Atmospheric and Oceanic Science at the University of Maryland, College Park, contains the following co-authored work:

Chapter 3 was published as *Krasnopolsky V. M., M. S. Fox-Rabinovitz, and A. A. Belochitski, 2008: Decadal Climate Simulations Using Accurate and Fast Neural Network Emulation of Full, Long-, and Short-wave Model Radiation, Monthly Weather Review, 136, pp. 3683-3695, doi: 10.1175/2008MWR2385.1* and as *Krasnopolsky V.M., M.S. Fox-Rabinovitz, Y.-T. Hou, S.J. Lord, and A.A. Belochitski, 2010: Accurate and Fast Neural Network Emulations of Model Radiation for the NCEP Coupled Climate Forecast System: Climate Simulations and Seasonal Predictions, Monthly Weather Review, 138, pp. 1822-1842, doi: 10.1175/2009MWR3149.1*

Chapter 6 was published as *V. M. Krasnopolsky, M.S. Fox-Rabinovitz, H.L. Tolman, and A. A. Belochitski, 2008: Neural Network Approach for Robust and Fast Calculation of Physical Processes in Numerical Environmental Models: Compound*

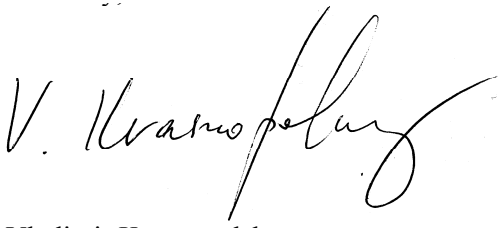
Parameterization with a Quality Control of Larger Errors, Neural Networks, 21, pp. 535-543, doi:10.1016/j.neunet.2007.12.019

Chapter 7 was published as *V. M. Krasnopolsky, M.S. Fox-Rabinovitz,, and A. A. Belochitski, 2008: Ensembles of Numerical Climate and Weather Prediction Models Using Neural Network Emulations of Model Physics, IEEE World Congress on Artificial Intelligence, Hong Kong, June 1-6, 2008, CD-ROM, pp. 1524-1531*

Inclusion of this work has the approval of Mr. Belochitski's dissertation advisors, Drs. Vladimir Krasnopolsky and Michael Fox-Rabinovitz, and the AOSC program chair Prof. James Carton.

Mr. Belochitski's examining committee has determined that the student made a substantial contribution to the work listed above.

Sincerely,

A handwritten signature in black ink, appearing to read 'V. Krasnopolsky', with a stylized, flowing script.

Vladimir Krasnopolsky

Research Advisor and Physical Scientist

Environmental Modeling Center, NCEP/NWS/NOAA

5200 Auth Rd., Camp Spring, MD 20746

tel: 301-763-8000 ext. 7262; fax: 301-763-8545

Acknowledgements

The work presented in this study would not be possible without the tremendous support of my two research advisors. Dr. Michael Fox-Rabinovitz directed, encouraged, educated, and motivated me all along the way. His guidance ranging from in-depth discussions of results to the broader vision of my work helped my formation as a scientist. Dr. Vladimir Krasnopolsky pioneered the techniques described, investigated and further developed in this dissertation. His direction and broad knowledge gave me the opportunity to do research at the forefront of modern science.

I would like to thank my academic advisor, Prof. Ferdinand Baer, who guided me through the graduate program at AOSC.

The DOE, NOAA, and NSF Grants have supported the research presented in this dissertation. The supercomputer facilities at DOE/NERSC, NSF/NCAR, and NOAA/NCEP have been used for model calculations. I am grateful for the collaboration and insightful discussions with Drs. Yefim Kogan of UCSD Scripps, Phil Rasch of DOE PNNL, Peter Blossey of the University of Washington, Steve Lord, Yu-Tai Hou, Suranjana Saha, Shrinivas Moorthi, Fanglin Yang of NOAA NCEP, and Peter Binev, Ronald DeVore, Philipp Lamby of the University of South Carolina. I would like to thank Prof. Marat Khairoutdinov of SUNY Stony Brook for sharing the code of his cloud resolving model (CRM) System for Atmospheric Modeling (SAM).

Table of Contents

Preface.....	ii
Acknowledgements.....	iv
Table of Contents.....	v
List of Tables	vii
List of Figures.....	ix
Chapter 1: Introduction.....	1
1.1 Motivation.....	1
1.2 Neural Networks Overview	4
1.3 Hybrid Environmental Modeling.....	5
1.4 Other Approaches to Emulation of Model Physics.....	6
1.5 Quality Control of Larger Errors in an NN Emulation of Model Physics.....	7
1.6 Ensemble of Neural Network Emulations for Climate Model Physics	8
Chapter 2: Neural Networks Approach to Emulation of Model Physics Components	10
2.1 Mappings.....	10
2.1.1 Definition of Mapping	10
2.1.2 Some Generic Properties of Mappings	11
2.2 Approximation of Nonlinear Mappings Using Neural Networks.....	13
2.2.1 Neural Networks in Terms of Approximation Theory	13
2.2.2 Training Set.....	15
2.2.3 Selection of NN Architecture.....	16
2.2.4 NN Training.....	18
2.3 Multiple NN Emulations for the Same Target Mapping and the NN Ensemble Approach.....	20
2.4 Application of NN Emulations to Development of Hybrid Atmospheric Models	22
Chapter 3: Neural Network Emulation of Full Model Radiation: Decadal Simulations and Seasonal Predictions.....	25
3.1 Brief Descriptions of GCMs	25
3.1.1 NCAR CAM	25
3.1.2 NCEP CFS and GFS	25
3.2 Long- and Short- Wave Radiation Parameterizations in GCMs.....	26
3.2.1 NCAR CAM	26
3.2.2 NCEP CFS and GFS	27
3.3 NN Emulations for Full Model Radiation	30
3.3.1 Bulk Approximation Error Statistics	30
3.3.2 NCAR CAM	31
3.3.3 NCEP CFS	33
3.4 Validation of Parallel Decadal Model Simulations	39
3.4.1 NCAR CAM	39
3.4.2 NCEP CFS	55
3.5 Discussion.....	73

Chapter 4: Neural Network Emulation of Full Model Radiation: Short-to-Medium Range Eight Days Forecasts with NCEP GFS.....	76
4.1 Brief Description of the High Resolution Version of NCEP GFS.....	76
4.2 NN Emulations for Full Model Radiation	77
4.3 Validation of Parallel Eight Day Model Forecasts	84
4.4 Discussion	97
Chapter 5: Investigation of Other Approaches to Emulation of Model Physics: Tree Approximation of the Long Wave Radiation Parameterization in the NCAR CAM	101
5.1. Motivation.....	101
5.2 Description of Algorithms	103
5.2.1 Approximate Nearest Neighbors.....	103
5.2.2 Regression Trees	105
5.3 Numerical Experiments	107
5.4 Results of a 10-Year Climate Simulation	112
5.5 Discussion	114
Chapter 6: Compound Parameterization of Full Radiation with a Quality Control of Larger Errors in NCAR CAM.....	117
6.1 Accuracy and quality control of NN emulations	117
6.2 Compound Parameterization Designs and Their Validation on Independent Data Sets.	121
6.3 Validation of Compound Parameterization in NCAR CAM	127
6.4 Discussion	129
Chapter 7: Ensemble of Neural Network Emulations for Climate Model Physics: The Impact on Climate Simulations.....	131
7.1 Ensemble Approaches for NWP and Climate Simulations.....	132
7.1.1 Ensembles with Perturbed Initial Conditions	133
7.1.2 Ensembles with Perturbed Physics	135
7.1.3 Short Term Ensembles with Perturbed Physics	137
7.2 Neural Network Ensembles with Perturbed Physics	140
7.3 Comparisons of Different Ensembles using Perturbed NCAR CAM LWR...	141
7.4 Discussion	151
Chapter 8: Summary	154
Glossary of Acronyms	160
Bibliography	162

List of Tables

Table 1. Statistics estimating the accuracy of HRs (in K/day) calculations, and computational performance for NCAR CAM-2 LWR and SWR using NN emulation vs. the original parameterization.....	29
Table 2. Statistics estimating the accuracy of HRs (in K/day) calculations and the computational performance for NCEP CFS (T126L64) LWR and SWR using NN emulation vs. the original parameterization.....	31
Table 3. Time (40-year) and global means for model diagnostics from NCAR CAM-2 control climate simulations with the original LWR and SWR, simulation with NN emulations for the full radiation using NN 50 (LWR) and NN 55 (SWR), and their differences.....	38
Table 4. Inputs and outputs of LWR and SWR NNs developed for GFS.....	76
Table 5. Statistics estimating the accuracy of HRs (in K/day) calculations and the computational performance for NCEP GFS (T574L64) LWR and SWR using NN emulation vs. the original parameterization.....	79
Table 6. Total RMSE and absolute value of total bias for emulation with neural network, approximate nearest neighbors, CART and Random Forests applied to the whole vector or component-wise.....	105

Table 7. Error Statistics for SWR NN Emulation NN55 and SWR Compound Parameterization: Bias and total RMSE, $RMSE_{26}$ at the lower model level, and Extreme Outliers (Min Error & Max Error).....	123
---	-----

List of Figures

Fig.1 Vertical distributions of NN emulation errors for two models: NCAR CAM (26 vertical layers) and NCEP CFS (64 vertical layers).....	33
Fig. 2 Zonal and time mean LWR Heating Rates, in K/day, for the NN LWR run (the upper left panel), the control run (the upper right panel) and their difference (the bottom panel).....	40
Fig. 3 Same as in Fig. 2 but for the SWR.....	41
Fig. 4 Zonal and time mean 2-meter temperature, in K, for the full radiation NN and control runs (the upper panel) and their difference (the bottom panel).....	42
Fig. 5 Zonal mean vertical distribution of time mean temperature, in K, for the full radiation NN run (the upper left panel), the control run (the upper right panel), and their difference or bias (the bottom panel).....	43
Fig. 6 Time mean temperature at 850 hPa, in K, for the full radiation NN run (the upper panel), the control run (the middle panel), and their difference (the bottom panel).....	46
Fig. 7 Vertical profile of time mean temperature, in K, at the Resolute, Canada station for the full radiation NN run (the dashed line), the control run (the solid line), and observations (the dotted line).....	47

Fig. 8 Winter-summer difference for time mean temperature at 850 hPa, in K, for the full radiation NN run (the upper panel), and the control run (the bottom panel).....	48
Fig. 9 Global mean time series for time mean temperature at 850 hPa, in K, for the full radiation NN run (the dotted line), and the control run (the solid line).....	49
Fig. 10 Annual cycle for global mean temperature at 850 hPa, in K, for the full radiation NN run (the dashed line), and the control run (the solid line).....	50
Fig. 11 Annual cycle for precipitation, in mm/day, for the full radiation NN run (the upper left panel), the control run (the upper right panel), and their difference or bias (the bottom panel).....	51
Fig. 12 The upper row: zonal and time mean Top of Atmosphere Upward Long (left panel) and Short (right panel) Wave Fluxes (in W per m ²) for the winter.....	59
Fig. 13 The 17-year (1990-2006) time-mean (NN run) SST distributions and bias/differences for winter (DJF: December-January-February, left column) and for summer (JJA: June-July-August, right column) for the full radiation NN run vs. the control run.....	60
Fig. 14. The same as in Fig. 3 but for total precipitation (PRATE). The contour levels for the PRATE fields are 2, 4, 8, 16 and 32 mm/day.....	61

Fig. 15 The same as in Fig. 4 but for total clouds. The contour intervals for the cloud fields are 20% and for the differences – 4%.....	62
Fig. 16 The same as in Fig. 5 but for convective precipitation clouds. The contour intervals for the cloud fields are 10% and for the differences – 4%.....	63
Fig. 17 The same as in Fig. 5 but for boundary layer clouds. The contour levels for the cloud fields are 10, 20, 40, 60, 80 and 100 % and for the differences – 4%.....	64
Fig. 18 The 17-year (1990-2006) time series of the total precipitable water anomaly (PWAT), with the seasonal cycle subtracted, for the full radiation NN run (dash-dotted line) and for two control runs, CTL (solid line) and CTL1 (dotted line).....	65
Fig. 19 The 17-year (1990-2006) time series for the Nino3.4 index for the reanalysis (CDAS) (the upper panel), and for the parallel full radiation NN (the bottom panel) and two control runs (the middle panels).....	66
Fig. 20 The 17-year (1990-2006) time series for global mean temperature at 850 hPa (in K) for the parallel full radiation NN (solid line) and the old control (large-dashed line) and new control (short-dashed line) runs.....	67
Fig. 21 Biases or differences between the NN and control runs (NN-CTL) and differences between two control runs (CTL1-CTL) for seasonal predictions.....	68

Fig. 22 Autocorrelation function for several NN input profiles.....	77
Fig. 23 Anomaly correlation at 850 mb (left column) and 500 mb (right column) for the northern hemisphere (upper row), tropics (medium row), and southern hemisphere (lower row) calculated for temperature fields.....	84
Fig. 24 Anomaly correlation at 500 mb for the northern hemisphere (upper row), tropics (medium row), and southern hemisphere (lower row) calculated for geopotential height fields.....	85
Fig. 25 Anomaly correlation calculated for the surface layer pressure fields. See caption to Fig. 24.....	86
Fig. 26 Temperature bias calculated for the northern hemisphere as a function of the forecast time (horizontal axis) and height in mb (vertical axis) for the control run, PRNNCTL (upper left), for PRNNCF5 (upper right), and for PRNNGFS.....	87
Fig. 27 The same as in Fig. 26 calculated for the tropics.....	88
Fig. 28 The same as in Fig. 26 calculated for the southern hemisphere.....	89
Fig. 29 Vector wind RMSE calculated for the northern hemisphere as a function of the forecast time (horizontal axis) and height in mb (vertical axis) for the control run,	

PRNNCTL (upper left), for (PRNNCFS – PRNNCTL) (upper right), and for (PRNNGFS – PRNNCTL) (lower left).....	91
Fig. 30 The same as in Fig. 29 calculated for the tropics.....	92
Fig. 31 The same as in Fig. 29 calculated for the southern hemisphere.....	93
Fig. 32 Comparison of neural network, approximate nearest neighbor, and several regression tree emulations.....	106
Fig. 33 Scatterplots for the approximation of the heating rates in the 26th and 9th vertical layer with componentwise CART (RPRC) and the vector-valued Random Forest (RFV) approximation.....	108
Fig. 34 Approximation of three representative heating rate profiles. Black line with markers: original parameterization.....	109
Fig. 35 Comparison of the predicted annual zonal means of the LWR heating rates computed with the original parameterization (top row), a tree based emulation (center row) and a neural network emulation (bottom row).....	112

Fig. 36 Comparison of the predicted annual means of the two meter air temperatures computed with the original parameterization (top row), a tree-based emulation (center row) and a neural network emulation (bottom row).....	113
Fig. 37. Compound parameterization design for the NCAR CAM SWR.....	115
Fig. 38. The correlation (binned scatter plot, the error bar shows the standard deviation inside the bin) between the actual error (<i>prmse</i> of the NN emulation NN55) and the error predicted by the error NN calculated vs. the original parameterization on an independent test data set.....	117
Fig. 39. Probability density distributions of emulation errors for the SWR NN emulation NN55 (solid line) and for the compound SWR parameterization.....	121
Fig. 40. Scatter plot for HRs (heating rates) calculated using the SWR NN emulation NN55 (the left panel) vs. the original SWR parameterization (left and right horizontal axes) and for HRs calculated using the SWR compound parameterization (the right panel) vs. the original SWR parameterization.....	123
Fig. 41. Errors (vs. the original SWR parameterization) produced by the SWR NN emulation during the model run (gray line), errors predicted by the error NN (black line), and errors produced after introducing CP instead of the SWR NN emulation (light gray line).....	125

Fig. 42 The PICE and PPE scenario.....	131
Fig. 43 The STPPE scenario.....	135
Fig. 44. Probability density function for ε_j	142
Fig. 45. Mean error (bias) and RMSE, in W/m^2 , for the mean winter DJF (December through February) surface net LWR flux (FLNS).....	143
Fig. 46. Min and max errors, in W/m^2 , for the mean winter DJF FLNS.....	144
Fig. 47. Mean error (bias) and RMSE, in W/m^2 , for the winter DJF (December through February) top of the model net LWR flux (FLNT).....	145
Fig. 48. Min and max errors, in W/m^2 , for FLNT climate.....	146
Fig. 49. Mean error (bias) and RMSE, in hPa, for the pressure at the surface level (PSL).....	147
Fig. 50. Min and max errors, in hPa, for PSL climate.....	148

Chapter 1: Introduction

1.1 Motivation.

One of the main problems of development and implementation of high-quality high-resolution environmental models is the complexity of physical (chemical and biological) processes involved. For example, calculation of model physics in a GCM (General Circulation Model) usually takes a very significant part of the total model computations. Evidently, this percentage is model dependent but full model radiation is the most time-consuming component of GCMs (e.g., Morcrette et al. 2007, 2008, Manners et al. 2009). In both climate modeling and NWP (Numerical Weather Prediction), the calculation of radiative transfer is necessarily a trade-off between accuracy and computational efficiency. Very accurate methods exist, such as line-by-line procedures that could be employed ideally to calculate radiative fluxes for every grid-point at every time-step. If the radiation transfer were to be computed for every grid point and at all time steps, it would generally require as much CPU time or more than the rest of the model components, i.e., model dynamics and other physical parameterizations (Morcrette et al. 2008). Therefore a number of simplifications are usually made to reduce this cost to manageable levels.

For example, in the majority of modern radiative schemes, the correlated-k method (Lacis and Oinas 1991) is typically used to reduce the integration over wavelength by effectively binning wavelengths with similar absorption coefficients (k-terms). This simplification reduces greatly the number of monochromatic radiative transfer calculations required. The number of k-terms can be adjusted, which provides a trade-off between the accuracy and efficiency required for a given application.

However, the correlated-k methods cannot be made sufficiently computationally efficient to allow calculations for every grid-point at every time-step.

Such a situation is an important motivation for developing new alternative numerical algorithms that provide faster calculations of model physics while carefully preserving their accuracy. Two techniques have been proposed to improve temporal and spatial resolution of radiation calculations: (1) the technique that improves interpolation of the radiative calculations from the coarse grid to the fine one (Morcrette et al. 2008) or improve radiative calculations between the time steps for which full radiative calculations are performed (Venema 2007, Manners et al. 2008), and (2) the technique that introduces either new fast radiation *parameterizations* (Chevallier et al. 1998, 2000) or accurate and fast *emulations* of existing radiation schemes and parameterizations (Krasnopolsky 1997, Krasnopolsky et al. 2005a, 2008a) that can be used in a model at each grid point and at each time step instead of original slow radiative calculations.

To reduce the cost further, calculations are usually made at lower temporal and/or spatial resolutions. Quite drastic reductions in temporal resolution are often made (e.g., radiation calculations are made every three hours for the climate and global forecast models at NCEP and UKMO (Manners et al. 2008)). Between radiative transfer calculations major changes may occur in the radiative profiles (caused primarily by two factors: changes in clouds and changes in the angle of incident solar radiation) that are not represented. A reduced horizontal resolution approach (the radiative calculations are performed on a coarser grid with a following interpolation

of the results to an original finer grid) is used to speed up radiation calculations at ECMWF (Morcrette et al. 2007, 2008). A reduced vertical resolution approach (the full radiation is calculated at every other vertical level and interpolated on the intermediate levels) is used in the Canadian operational Global Environmental Multiscale model (e. g. Cote et al. 1998a, 1998b). Such approaches reduce horizontal or vertical variability of radiation fields. Thus, these approaches may reduce the accuracy of a model's radiation calculation and its spatial or/and temporal consistency with other parts of model physics and with model dynamics, which may, in turn, affect negatively the accuracy of climate simulations and weather predictions.

A fast neural network (NN) based long wave radiation parameterization NeuroFlux (Chevallier et al. 1998, 2000) has been developed and tested in the ECMWF model. The NeuroFlux approach has a limited application (as discussed in Krasnopolsky et al. 2005b) because it has been developed for a particular formulation (Washington and Williamson 1977) of the long wave radiation physics only. Also, because of NeuroFlux's suboptimal design (as discussed in Krasnopolsky et al. 2005b), at vertical resolution of 60 layers and more, both accuracy and rapidity of NeuroFlux cannot be achieved simultaneously (Morcrette et al. 2008). Consequently, the NeuroFlux is used only for the 4D-Var linearized physics (Janiskova et al., 2002) where the accuracy requirements are less stringent.

The NN based approach introduced in Krasnopolsky (1997), Krasnopolsky et al. (2005a, 2008a) will be discussed in substantial detail further in this dissertation.

1.2 Neural Networks Overview

The neural network (NN) approach is a relatively new, diverse, and powerful statistical learning technique (also known as machine learning, learning from data, predictive learning, or the data-driven approach) that started developing rapidly in the mid-1980s after several major basic types of NNs were introduced in the works of Kohonen (1982), Hopfield (1982) Rumelhart et al. (1986), and Lippmann (1989) In the 1990s this technique matured; several well written and fundamental textbooks have been published (Beale and Jackson, 1990; Bishop, 1995; Haykin, 1994; Ripley, 1996; Vapnik, 1995; Cherkassky and Mulier, 1998) that introduced NNs as a new powerful statistical learning approach capable of providing a diverse family of flexible nonlinear data driven models for various applications. This approach became appealing to a broad community of professionals, including scientists working in different fields of geosciences like satellite remote sensing, meteorology, oceanography, and geophysical numerical modeling. Since then a significant number of NN applications have been developed in these fields. Selected atmospheric and oceanic applications have been reviewed for the atmospheric and oceanic community by Gardner and Dorling (1998), Hsieh and Tang (1998), and Krasnopolsky (2007a) and for the NN community by Krasnopolsky and Chevallier (2003) and Krasnopolsky and Fox-Rabinovitz (2006b). Selected remote sensing applications have been reviewed for remote sensing experts by Atkinson and Tatnall (1997) and for the NN

community by Krasnopolsky and Schiller (2003). Applications of the NN technique for developing nonlinear generalizations of multivariate statistical analysis have been reviewed by Hsieh (2004).

1.3 Hybrid Environmental Modeling

A new notion of hybrid environmental numerical models (ENM) has been introduced in Krasnopolsky (1997), and Krasnopolsky et al. (2005a, 2008a), which combines deterministic (e.g. model dynamics) and statistical or machine learning (e.g. NN emulations of model physics) components to perform calculations more effectively than original completely deterministic ENM.

A NN *emulation* of a model physics parameterization is a functional imitation of this parameterization so that the results of model calculations with the original parameterization and with its NN emulation are *physically* (and climatologically) identical. High quality of NN emulations is achieved due to the high accuracy of approximation of the original components.

Due to the capability of modern Statistical or Machine Learning Techniques (SLTs and MLTs), such as NN and the tree approximation used in our research, to provide an *unprecedented accuracy* for approximation of complex systems like model physics, our NN emulations of model physics parameterizations are practically identical to original physical parameterizations. As a result, HEM using this emulation produces results, which are physically identical to those of the original ENM. In other words, the underlying idea of the approach is not developing a new parameterization but rather emulating a parameterization already very carefully tested and validated by its developers off-line and then on-line through experimentation

with the entire model. It is achieved by using for NN training data simulated by running an original model (i.e. ENM) with the original parameterization. Using model-simulated data for NN training allows us to achieve a very high accuracy for approximation because simulated data are free of the problems typical for empirical data (problems like high level of observational noise, sparse spatial and temporal coverage, poor representation of extreme events, etc.). In the context of this approach, the accuracy and improved computational performance of HEM and NN emulations is always measured against the ENM using the original parameterization. It is noteworthy that the developed NN emulation has *the same inputs and outputs* as the original parameterization and is used as its *functional substitute* in the model.

1.4 Other Approaches to Emulation of Model Physics

While artificial neural networks can be considered the current state-of-the-art black box methodology for a wide range of high-dimensional approximation problems, and justifiably so, they may not necessarily be the best solution for the application considered in this dissertation. While the application of neural networks to the problem of learning parameterizations has produced excellent results, it is not without limitations. Foremost among these is that the neurons have large support (a half space in \Re^n) and their superposition is a complex nonorthogonal expansion. This makes capturing of local or multiscale phenomena difficult. This brings into question whether NNs are the ultimate statistical learning technique (SLT) solution to numerically emulating parameterizations. Therefore we will explore an alternative to neural networks within the class of non-parametric approximation methods. We will

restrict ourselves to basic design decisions and discuss the features of two common statistical learning paradigms, approximate nearest neighbors and regression trees.

1.5 Quality Control of Larger Errors in an NN Emulation of Model Physics

Tremendous complexity, multidimensionality, and nonlinearity of the climate/weather system and numerical models describing this system lead to complexity and multidimensionality of our NN emulations and data sets that are used for their development and validation. The development of NN emulations of model physics and their accuracy depends significantly on our ability to generate a representative training set to avoid using NNs for extrapolation beyond the domain covered by the training set. Owing to the high dimensionality of the input domain (i.e., dimensionality of the NN input vector) which is of the order of several hundreds or more, it is difficult if not impossible to cover the entire domain, especially its “far corners” associated with rare or extreme events, even when we use model simulated data for the NN training. Also, the domain may change with time as in the case of climate change. In such situations the emulating NN may be forced to extrapolate beyond its generalization ability which may lead to larger errors in NN outputs and, as a result, to errors in the numerical models in which they are used. Therefore, we will explore a quality control (QC) procedure, which can predict and eliminate larger errors of NN emulations during the integration of highly nonlinear numerical models, not just relying upon the robustness of the model that can vary significantly for different models. Such a mechanism would make our NN emulation approach more reliable, robust, and generic.

1.6 Ensemble of Neural Network Emulations for Climate Model Physics

During the last decade, the ensemble approach demonstrated a significant success in numerical weather prediction (NWP) modeling (Palmer, 2007; Buizza, 2005) and in climate modeling (Broccoli, 2003; Murphy, 2004; Staniforth, 2005). The traditional ensemble approach widely used in NWP is based on introducing perturbations into initial conditions because NWP forecasts (specifically, for short- to medium-term or 1 to 10 day weather predictions) are the initial condition problems. Hereafter, we will call this kind of ensembles the perturbed initial condition ensemble (PICE).

It was also found that, for both the NWP and especially for climate applications, the spread of PICE forecasts is insufficient to systematically capture the natural climate and weather variability (both spatial and temporal). Another approach to ensemble modeling based on perturbing model physics developed and implemented for ensemble forecast systems (Buizza, 2005; Buizza, 1999). Climate simulations which are from months to decades (and sometimes centuries) long are not initial condition but rather boundary condition problems. In other words, climate simulations “forget” the initial conditions after two-three weeks of model integrations, and are driven by the right hand side (r.h.s.) or model physics forcing.. For this kind of problems, an ensemble approach based on perturbation of model physics (or perturbation of model forcing) seems to be appropriate. The perturbed physics ensembles are expected to be more suitable for climate model simulations and projections (Staniforth, 2005).

In this chapter we investigate different possibilities of using the neural network (NN) emulation technique, introduced in the earlier chapters, in combination with ensemble

approaches. We discuss two types of perturbed physics ensembles: a long term perturbed physics ensemble (PPE) and a short-term perturbed physics ensemble (STPPE).

We also show that the NN emulation technique can be efficiently used to create PPE and STPPE. We demonstrate that all three aforementioned types of ensembles (PICE, PPE, and STPPE) can significantly benefit, in terms of their numerical performance, from using accurate NN emulations of model physics; however, STPPE becomes especially efficient (orders of magnitude faster than PICE and PPE) when the NN technique is used to produce the ensemble of perturbed realizations of model physics.

Chapter 2: Neural Networks Approach to Emulation of Model Physics Components

2.1 Mappings

2.1.1 Definition of Mapping

A mapping M between two vectors X (input vector) and Y (output vector) can be symbolically written as

$$Y = M(X); X \in \mathfrak{R}^n, Y \in \mathfrak{R}^n \quad (1)$$

A large number of important practical geophysical applications may be considered mathematically as a mapping like (1). Keeping in mind that a NN technique will be used to approximate this mapping, we will call it a target mapping, using a common term from nonlinear approximation theory (DeVore, 1998). The target mapping may be given to us explicitly or implicitly. It can be given explicitly as a set of equations based on first principles and/or empirical dependencies (e.g., radiative transfer or heat transfer equations) or as a computer code. Observational records represent an implicit target mapping. In this case the target mapping is assumed to be hidden in or behind observed data and to generate these data.

Multidimensional, nonlinear mappings (1) are complicated mathematical objects that are not very well studied. There are many different interesting properties of these mappings that could be discussed. However it will be easier for us to focus on some generic properties of the mapping (1) that are typical and important for the applications presented in this dissertation, keeping in mind that our goal is to develop a NN emulation for the target mapping (1).

2.1.2 Some Generic Properties of Mappings

The first essential property of the target mapping is its mapping dimensionalities. A mapping is characterized by two dimensionalities: dimensionality n of the input space \mathfrak{R}^n and dimensionality m of the output space \mathfrak{R}^m . The second property of the mapping (1) is the mapping domain. If all components of the input vector \mathbf{X} are scaled to $[-1,1]$, the volume of the input space \mathfrak{R}^n is equal to 2^n and therefore grows exponentially with n . Once the space is discretized, e.g., by K values per dimension, then the problem size grows even faster, as K^n . This is usually called the curse of dimensionality (Bishop, 1995; Vapnik, 2006). Fortunately, the components of the input vector \mathbf{X} are usually interrelated or multicollinear (Aires et al., 2004b) because of the physical or statistical reasons, which leads to both positive and negative consequences. These correlations effectively reduce the size, and sometimes dimensionality, of the part of the input space \mathfrak{R}^n spanned by the input vectors \mathbf{X} (Bishop, 1995). This part is called the mapping domain D and is determined by a particular application. Understanding the configuration of the mapping domain and its properties is very important for a proper NN training and application. The components of the output vector \mathbf{Y} are usually also interrelated. As a result, the output vectors also span only a fraction of the output space \mathfrak{R}^m . This part of the output space is called the range R . Understanding the properties of the range is very important for the proper testing and application of the developed NN approximations of a target mapping (1).

Another property of the mapping (1) that is important in the context of the applications reviewed in this dissertation is the mapping complexity. Mapping

complexity is an intuitively clear notion. The mapping M performs some transformation of the input vector X to produce the output vector Y , and this transformation may be more or less complex. However, no formal definition of complexity is available. Nevertheless we can talk about the physical complexity of the mapping (1) that corresponds to the complexity of the physical processes represented mathematically by this mapping. Correspondingly, we can introduce semi-quantitative characteristics of physical complexity, the number of equations describing the physics, the type of these equations, the dimensionality of the equations, etc. This “measure” of complexity is obviously ambiguous due to, for example, existence of alternative mathematical formalisms. The second type of complexity that can be introduced is mapping’s numerical or computational complexity. For this type of complexity a quantitative measure, like the number of elementary numerical operations required for calculating Y given X , can be introduced. However, this measure is also ambiguous because different numerical schemes can be applied to the same set of equations. The third definition of mapping complexity is the functional complexity. It describes the complexity of the functional dependency of the outputs Y versus inputs X or the “smoothness” of this dependency. For example, for a function of one variable an approximation procedure can be used for measuring functional complexity. If n is the minimal order of a polynomial that approximates the function with the desired accuracy, the function may be considered to have polynomial complexity of the order n . The direct generalization of this approach for the case of a multidimensional mapping (1) is hardly possible.

2.2 Approximation of Nonlinear Mappings Using Neural Networks

2.2.1 Neural Networks in Terms of Approximation Theory

The simplest NN, a multi layer perceptron (MLP), which in traditional NN terms, corresponds to an NN with one “hidden” “layer” and a linear output “layer”, is a generic analytical nonlinear approximation or model for a mapping like the target mapping (1). The MLP NN uses for the approximation a family of functions like

$$y_q = NN(X, a, b) = a_{q0} + \sum_{j=1}^k a_{qj} z_j, q = 1, 2, \dots, m \quad (2)$$

$$z_j = \phi \left(b_{j0} + \sum_{i=1}^n b_{ji} x_i \right) \quad (3)$$

where x_i and y_q are components of the input and output vectors, respectively, a and b are fitting parameters or NN weights, ϕ is a so-called activation or “squashing” function (a nonlinear function, often a hyperbolic tangent), n and m are the numbers of inputs and outputs, respectively, and k is the number of the highly nonlinear basis function z_j (see equation (3)) in the expansion (2). The expansion (2) is a linear expansion (a linear combination of the basis functions z_j (eq. (3))), and the coefficients a_{qj} ($q = 1, \dots, m$ and $j = 1, \dots, k$) are linear coefficients in this expansion. It is essential that the basis functions z_j are nonlinear with respect to inputs x_i ($i = 1, \dots, n$) and to the fitting parameters or coefficients b_{ji} ($j = 1, \dots, k$). As a result of the nonlinear dependence of the basis functions on multiple fitting parameters b_{ji} the basis $\{z_j\}, j=1, \dots, k$ turns into a very flexible set of non-orthogonal basis functions that have a great potential to adjust to the functional complexity of the

mapping (1). It has been shown by many authors in different contexts that the family of functions (2) and (3) can approximate any continuous or almost continuous (with a finite number of finite discontinuities such as a step function) mapping (Cybenko, 1989; Funahashi, 1989; Hornik, 1991; Chen and Chen, 1995a, 1995b). The accuracy of the NN approximation or the ability of the NN to resolve details of the target mapping (1) is proportional to the number of basis functions (hidden neurons) k (Attali and Pages, 1997).

The numerical complexity of NN (2) can be well approximated by the number of NN weights, a 's and b 's, in (2-3) (Krasnopolsky 2007a):

$$N_C = k \cdot (n + m + 1) + m$$

The time, T_{NN} , required for estimating the NN (1) is directly proportional to the NN numerical complexity N_C ,

$$T_{NN} = c \cdot N_C$$

with the coefficient of proportionality c depending mainly on a hardware and software environment of the computer used.

Obviously, the numerical complexity, N_C , increases linearly with the increase of vertical resolution of a model (the number of the vertical layers, L) because both n and m depend linearly on L . Thus, as a result, the time required for estimating NN, T_{NN} , increases linearly with the increase of the vertical resolution of the model. The time required for estimating the original parameterization, T_O , also increases with the increase of vertical resolution. For the original parameterization, the dependence of the calculation time on vertical resolution is strongly conditioned by the numerical scheme implemented.

Thus, the dependence of the speedup, η , provided by an NN emulation, on vertical resolution is determined by the ratio of the two aforementioned calculation times:

$$\eta = \frac{T_O}{T_{NN}}$$

the time required for estimating the original parameterization, T_O , and the time required for estimating the NN emulation, T_{NN} . Therefore, the change of the speedup η with the increase of model vertical resolution will strongly depend on the physical complexity of the original parameterization and on the numerical scheme implemented.

2.2.2 Training Set

In practical applications the target mapping (1) is usually represented and presented to the NN by a data set (training set) that consists of N pairs or records of input and output vectors X and Y ,

$$C_T = \{X_p, Y_p\}_{p=1, \dots, N} \quad (4)$$

where $Y_p = M(X_p) + \xi_p$, ξ_p is an error of observation or calculation with the probability density function $\rho(\xi)$, and $X_p \in D$ and $Y_p \in R$. The training set is all that the NN “knows” about the target mapping that it is supposed to approximate. This is the reason why the MLP NN belongs to a class of data-driven methods (Cherkassky and Mulier, 1998).

The training set is all the NN “knows” about the mapping (1), and therefore it has to be representative. It means that the training set has to have a sufficient complexity corresponding to the complexity of the target mapping, allowing the NN to

approximate the target mapping with a desired accuracy. The set should have a sufficient number N of properly distributed data records that adequately resolve the functional complexity of the target mapping (1). The set should have finer resolution where the target mapping is not very smooth and coarser resolution where it is smoother. In other word, the domain D should be properly sampled. It may be oversampled but not under sampled. The paramount question remains, however, as to just how we should estimate this target mapping smoothness in order to obtain desired representativeness of the training set (DeVore, 1998). As we discussed above, the interrelations between inputs simplify the sampling task for cases of high input dimensionality, reducing the size and the effective dimensionality of the domain.

For applications considered in this dissertation an explicit theoretical (based on first principles) or an empirical model for the target mapping (1) is available, it can be used to simulate the data set (equation (4)). With simulated data we have significantly more control over the sampling of the target mapping domain (the number and distribution of the data points) and, as a result, over the NN accuracy and the ability of the emulating NN to resolve the target mapping. The level of noise in the simulated data is usually lower than that in the observed data. The simulated and observed data can, in principle, be fused together in an integrated data set using an appropriate technique that is able to account for the different error statistics and statistical properties of these two data types. One example of fused data is the analyzed data produced by a data assimilation system.

2.2.3 Selection of NN Architecture

To approximate a particular target mapping (1) with the MLP NN (2) and (3), we should first select the NN architecture or “topology”. That is, we must select the number of the inputs n , the number of the outputs m , and the number of neurons k in the hidden layer. For each particular problem, n and m are determined by the input and output dimensionalities of the target mapping (the dimensions of the input and output vectors \mathbf{X} and \mathbf{Y}). Practical implementation of this approach allows for multiple solutions in terms of the number of NN designs that can be used for an approximation. The MLP NN presented by equations (2) and (3) can be implemented as a single NN with m outputs, m single-output NNs, or several multiple output NNs with the total number of outputs equal to m .

The possible choices among many topological solutions, from a single NN with m outputs to m single output NNs, demonstrate an important flexibility of the NN technique that offers a speed versus accuracy trade-off. This additional flexibility can be effectively used for various applications. Another degree of flexibility is provided by the availability of different normalizations for NN inputs and outputs. This topic is discussed in detail in Krasnopolsky and Fox-Rabinovitz (2006b).

The number of hidden neurons k that determines the complexity of the approximating NN in each particular case should be determined when taking into account the complexity of the target mapping to be approximated. The more complicated the mapping, the more hidden neurons k is required (Attali and Pages, 1997) (or the higher the required complexity N_c of the approximating NN) to approximate this mapping with the desired accuracy or resolution. There is always a trade-off between the desired resolution of the target mapping and the complexity of the NN emulation.

However, from our experience the complexity k of the approximating NN should be carefully controlled and kept to the minimum level sufficient for the desired accuracy of the approximation to avoid over-fitting and to allow for a smooth and accurate interpolation. Unfortunately, there are no universal rules or recommendations to be given here. Usually, k is determined using experience and experiments.

2.2.4 NN Training

After NN topological parameters are defined, the weights (a and b) can be found using the training set C_T and the maximum likelihood method (Vapnik, 1995) by maximizing the functional

$$L(a, b) = \sum_{i=1}^N \ln \rho(Y_i - NN(X_i, a, b)) \quad (5)$$

with the respect to free parameters (NN weights) a and b . Here $\rho(\xi)$ is the probability density function for errors ξ_p . If the errors ξ_p are normally distributed, equation leads to the minimization of the least squares error, loss, risk, or cost function with respect to the NN weights a and b ,

$$E(a, b) = \sum_{i=1}^N (Y_i - NN(X_i, a, b))^2 \quad (6)$$

This procedure is usually called NN training. It is noteworthy that for a probability density function $\rho(\xi)$ other than the normal one the error function should be derived from the maximum likelihood functional (5). The error function may be significantly different than the least squares error or loss function (6). However, in the majority of applications the least squares error function (6) is applied.

Optimal values for weights are obtained by minimizing the error function (5) or (6); this task is a nonlinear minimization problem. A number of methods have been developed for solving this problem (Bishop, 1995; Haykin, 1994). Here we briefly outline one of them, a simplified version of the steepest (or gradient) descent method known as the back propagation training algorithm (Rumelhart et al., 1986).

The back propagation training algorithm is based on the simple idea that searching for a minimum of the error function (6) can be performed step by step in iterations and that at each step we should increment or decrement the weights in such a way as to decrease the error function. This can be done using, for example, a simple steepest descent rule

$$\Delta W = -\eta \left(\frac{\partial E}{\partial W} \right) \quad (7)$$

where η is a so-called learning constant and W is either one of two weights (a or b). Using (6), (2), and (3), the derivative in (7) can be expressed analytically through the derivative of the activation function φ and through the weight values at the previous iteration step. At the first step where we do not have weights from a previous training iteration, a weight initialization problem arises that is familiar to those who use various kinds of iterative schemes. Many publications have been devoted to weight initialization (e.g., Nguyen and Widrow, 1990; Wessels and Bernard, 1992).

A nonlinear error function (6) has multiple local minima. The back propagation algorithm converges to a local minimum, as does almost any algorithm available for solving the nonlinear optimization problem (NN training). Usually, multiple

initialization procedures are applied to avoid shallow local minima and to choose a local minimum with a sufficiently small error.

On the other hand, these local minima, which may have approximation error, give different solutions in terms of NN weights. These different NNs provide different interpolations and different derivatives. Thus, in such cases the approximation error may be not instrumental without using additional criteria for selecting solutions with good interpolation properties and derivatives.

2.3 Multiple NN Emulations for the Same Target Mapping and the NN Ensemble Approach

The existence of multiple solutions is a property of nonlinear models and nonlinear approximations. These models have many nonlinear parameters that could change in the process of generating solutions. These multiple solutions may be close in terms of a particular criterion used for obtaining the solutions. At the same time these models (NNs) may be different in terms of other criteria that provide complementary information about the target mapping. The availability of multiple solutions may lead to some inconveniences and uncertainties, e.g., the necessity of introducing an additional step to use additional criteria to select a single model. On the other hand, the availability of multiple models (NN emulations) providing complementary information about the target mapping opens the opportunity to use an ensemble approach that allows integration of the complementary information contained in the ensemble members into an ensemble that “knows” more about the target mapping than does any of the ensemble members (particular NN emulations).

The idea that an ensemble of learning models consisting of many members is capable of providing a better description of the system than any particular member model can be traced back to as early as the late 1950s and early to middle 1960s (Selfridge, 1958; Nilsson, 1965). Since the early 1990s, many different algorithms based on similar ideas have been developed for NN ensembles (Hansen and Salamon, 1990; Sharkey, 1996; Naftaly et al., 1997; Opitz and Maclin, 1999; Hsieh, 2001)

An ensemble of NNs consists of a set of members, i.e., individually trained NNs. They are combined when applied to new data to improve the generalization (interpolation) ability because previous research showed that an ensemble is often more accurate than any single ensemble member. Previous research also suggests that any mechanism that causes some randomness in the formation of NN members can be used to form a NN ensemble (Opitz and Maclin, 1999). For example, ensemble members can be created by training different members on different subsets of the training set (Opitz and Maclin, 1999), by training different members on different subdomains of the training domain, by training different members using NNs with different architectures (different numbers of hidden neurons) (Hashem, 1997), or by training different members using NNs with the same architecture but different initial conditions for the NN weights (Maclin and Shavlik, 1995; Hsieh, 2001).

In the context of our application, an approximation of a complex mapping (1), the members of the ensemble are separately trained approximating NNs, which provide different accuracies of approximation for the target mapping and different interpolation accuracies. We can expect that the ensemble average will provide a better approximation and interpolation than the individual members. Krasnopol'sky

(2007) also applied the NN ensemble technique to reduce the uncertainty of the NN Jacobian. Most of the previous work with NN ensembles has been done in the context of solving the classification (Hansen and Salamon, 1990; Sharkey, 1996; Opitz and Maclin, 1999) or the prediction of time series problems (Naftaly et al., 1997; Hsieh, 2001). The NN ensembles will be explored further in the Chapter 7 of this dissertation.

2.4 Application of NN Emulations to Development of Hybrid Atmospheric Models

A new concept of a complex HEM has been formulated and developed by Krasnopolsky et al. (2000b, 2002, 2005a) and Krasnopolsky and Fox-Rabinovitz (2006a, 2006b). The hybrid modeling approach considers the whole GCM or ENM as a system. Dynamics and parameterizations of physics, chemistry, etc., are considered to be the components of the system. Hybridization in this case is introduced at the level of components inside the system (ENM). For example, the entire LWR (or SWR) parameterization is emulated by a single NN as a single/elementary object or block. The NN emulation approach is based on the general fact that any parameterization of model physics can be considered as a continuous or almost continuous mapping (1).

Krasnopolsky and Fox-Rabinovitz (2006a, 2006b) formulated a developmental framework and test criteria that can be recommended for developing and testing the statistical learning components of HGCM, i.e., NN emulations of model physics components. The developmental process consists of three major steps.

The first step is problem analysis or analysis of the model component (target mapping (1), i.e., the original parameterization) to be approximated to determine the optimal structure and configuration of the NN emulations, the number of inputs and outputs, and the first guess of the functional complexity of the original parameterization that determines an initial number of hidden neurons k in one hidden layer of (2) and (3).

The second step is generation of representative data sets for training, validation, and testing. This is achieved by using for NN training the data that is simulated by running the original GCM, or in other words, a GCM with the original unmodified parameterization. When creating a representative data set, the original GCM must be run long enough to produce all possible atmospheric model simulated states, phenomena, etc. Here, because of the use of simulated data it is not a problem to generate the sufficiently representative (and even redundant) data sets required to create high-quality NN emulations. Using model-simulated data for NN training allows a high accuracy of emulation to be achieved because simulated data are almost free of the problems typical in empirical data (like a high level of observational noise, sparse spatial and temporal coverage, and poor representation of extreme events).

The third step is training the NN. Several different versions of NNs with different architectures, initialization, and training algorithms should be trained and validated. As for the NN architecture the number of hidden neurons k should be kept to the minimum number that provides a sufficient emulation accuracy to create the high-quality NN emulations required.

Testing the HGCM that uses the trained NN emulation consists of two major steps. The first step is testing the accuracy of the NN approximation against the original parameterization using the independent test data set. In the context of the hybrid approach the accuracy and improved computational performance of NN emulations and eventually the HGCM are always measured against the corresponding controls, namely, the original parameterization and its original GCM. Both the original parameterization and its NN emulation are complicated multidimensional mappings. Many different statistical metrics of the emulation accuracy should be calculated to assure that a sufficiently complete evaluation of the emulation accuracy is obtained. For example, total, level, and profile statistics have to be evaluated (see section 3.3.1). The second test step consists of a comprehensive comparison and analysis of parallel HGCM and GCM runs. For the parallel model simulations, all relevant model prognostic (i.e., time-dependent model variables) and diagnostic fields should be analyzed and carefully compared to assure that the integrity of the original GCM and its parameterization, with all its details and characteristic features, is precisely preserved when using a HGCM with NN emulation. This test step involving model simulations is crucially important. GCMs are essentially nonlinear complex systems; in such systems, small systematic and even random approximation errors can accumulate over time and produce a significant impact on the quality of the model results. Therefore the development and application framework of the new hybrid approach should be focused on obtaining a *high accuracy* in both NN emulations and HGCM simulations.

Chapter 3: Neural Network Emulation of Full Model Radiation: Decadal Simulations and Seasonal Predictions.

3.1 Brief Descriptions of GCMs

Two general circulations models of different levels of complexity were used in this study.

3.1.1 NCAR CAM

The NCAR (National Center for Atmospheric Research) CAM (Community Atmospheric Model) Version 2 is described in detail in the special issue of Journal of Climate, 1998, vol. 11, no. 6. We used the NCAR CAM v. 2 with the T42 (~3 degree) horizontal resolution and 26 vertical levels (T42L26) and with the climatological sea surface temperature forcing.

The calculation of model physics in this configuration of NCAR CAM takes about 70% of the total model computations. Evidently, this percentage is model and configuration dependent but full model radiation in general is the most time-consuming component of GCMs (e.g., Morcrette et al. 2007a, b). Such a situation is an important motivation for looking for new alternative numerical algorithms that provide faster calculations of model physics while carefully preserving their accuracy.

3.1.2 NCEP CFS and GFS

The operational NCEP CFS (Climate Forecasting System) is described in detail in Saha et al. (2006) and the references therein. The coupled NCEP CFS version used in our study incorporates: the NCEP GFS (Global Forecast System) 64-level

atmospheric model, the 40-level interactive MOM4 ocean model, the interactive Noah land model with four soil levels with improved treatment of snow and frozen soil, an interactive sea ice model with fractional ice cover and depth allowed, a sub-grid scale mountain blocking, a new seasonal climatological aerosol treatment, a historical CO₂ database from global observations collected by the World Meteorological Organization, a variable solar constant database, and historical stratospheric volcanic aerosol distributions (Sato et al., 1993).

The NCEP GFS model is a mature, state-of-the-art spectral atmospheric GCM (AGCM) used in operational medium-range weather forecasts. The operational GFS version has a variable horizontal spectral resolution of up to T574 or ~25 km. The hybrid sigma-pressure coordinate and a conservative finite-difference scheme are used in the vertical domain. The operational model is run with 64-layer vertical resolution between the surface and 0.27 hPa (about 60 km).

3.2 Long- and Short- Wave Radiation Parameterizations in GCMs

3.2.1 NCAR CAM

The function of the LWR parameterization in atmospheric GCMs is to calculate heating fluxes and rates produced by LWR processes. The complete description of NCAR CAM atmospheric LWR and SWR parameterizations is presented in Collins (2001, 2002).

The input vectors for the NCAR CAM-2 LWR parameterization include ten profiles (atmospheric temperature, humidity, ozone, CO₂, N₂O, CH₄, two CFC mixing ratios (the annual mean atmospheric mole fractions for the halocarbons), pressure, cloud

emissivity, and cloud cover) and one relevant surface characteristic (the upward LWR flux at the surface). The LWR parameterization output vectors consist of the profile of heating rates (HRs) and several radiation fluxes, including the outgoing LW radiation flux from the top layer of the model atmosphere (the outgoing LWR or OLR).

The input vectors for the SWR parameterization include twenty one vertical profiles (specific humidity, ozone concentration, pressure, cloud cover, layer liquid water path, liquid effective drop size, ice effective drop size, fractional ice content within cloud., aerosol mass mixing ratios, etc.), solar zenith angle and surface albedo for four different bands. The SWR parameterization output vectors consist of a vertical profile of heating rates (HRs) and several radiation fluxes. The NN emulations of the SWR parameterization have 173 inputs and 33 outputs.

It is noteworthy that the number of NN inputs is less than the number of input profiles multiplied by the number of the vertical layers plus the number of relevant single level characteristics. Many input variables (e.g., all gases) have zero or constant values in upper vertical layers. These constant values are not included as NN inputs because NN does not need constants inputs.

3.2.2 NCEP CFS and GFS

NCEP CFS's and GFS's radiation components contain a GCM version (v2.3) of the Rapid Radiative Transfer Model (RRTM) for LWR (hereafter referred to as RRTMG-LW) developed at AER Inc. (e.g. Mlawer et al., 1997; Iacono et al., 2000), and a

SWR based on Chou's parameterization scheme (Hou et al., 2002; Chou and Suarez, 1999). In the coupled CFS and standalone GFS used in this study the SWR of the operational versions of the models has been replaced by a GCM version (v2.3) of the AER's RRTM SWR (hereafter referred to as RRTMG-SW) (e.g. Clough et al., 2005) to improve the accuracy of SWR calculation.

The RRTMG-LW in the CFS model employs a computationally efficient correlated-k method for radiative transfer calculations. It contains 16 spectral bands with various number of quadrature points (g-points) in each of the bands that sums up to a total of 140 g-points (e.g., Mlawer et al., 1997, Iacono et al., 2000). Active gas absorbers include H₂O, O₃, CO₂, CH₄, N₂O, O₂, and four types of halocarbons (CFCs). A maximum-random cloud overlapping scheme is used for cloudy sky radiative transfer, and a climatological aerosol scheme provides the global distribution of aerosol optical depth. In this study, a one-hour frequency of radiation calculation is applied to both SWR and LWR. In the current version of the LWR parameterization the level of atmospheric CO₂ and its time dependence is presented by its global mean value that increased from 350 to 380 ppmv during the period of model integration used in this study (1990 to 2006).

Beside the RRTMG-LW, which is a faster member of the RRTM LWR family, we have also experimented with another version of the RRTM LWR (hereafter as RRTMF-LW) in this study. The RRTMF-LW is based on AER's RRTM-LW v3.0. It uses a full 16 g-points in each of the 16 spectral bands that add to a total of 256 vs. the reduced total of 140 in the faster RRTMG-LW. Unlike the diffusivity approach

(one zenith angle of about 53°) in the faster RRTMG-LW, the RRTMF-LW uses multi-angle radiance integration over a hemisphere to yield better accuracy (we set it at 3 angles in the study). As a result, the RRTMF-LW is about five times slower than the RRTMG-LW in exchange for improved accuracy (Mlawer et al., 1997).

The SWR parameterization used in the CFS is a modified version of AER's RRTMG-SW (v2.3) (Clough et al., 2005). It contains 14 spectral bands with various numbers of g-points in each of the bands to a total of 112. RRTMG-SW uses a fast two-stream radiative transfer scheme, and supports sophisticated absorption and scattering processes by clouds, aerosols, and absorbing gases (H_2O , O_3 , CO_2 , CH_4 , N_2O , O_2). Thus, in the current version of the SWR parameterization the level of atmospheric CO_2 and its time dependence is presented by the entire 3-D CO_2 field that changes with time in accordance with the change of the mean CO_2 level that increased from 350 to 380 ppmv during the period of model integration used in this study (1990 to 2006).

Although both RRTMG-LW and RRTMG-SW are built with fast computation schemes designed for GCM applications, they still represent the most time-consuming physics in the NCEP CFS model. The percentage of the total model computation time used by model physics and by radiation (LWR and SWR) vary depending largely on the model horizontal and vertical resolution, the time step, the frequency of radiative calculations, and the computing environment (e.g. the number of processors and threads). For example, in the CFS configuration at the T126L64 resolution, with the new RRTMG-LW and RRTMG-SW both called every hour, the

portion of the radiation computation time is about 57% of the total AGCM model computation time.

3.3 NN Emulations for Full Model Radiation

3.3.1 Bulk Approximation Error Statistics

To ensure a high quality of representation of the LWR and SWR radiation processes, the accuracy of the NN emulations has been carefully investigated. Our NN emulations have been validated against the original NCAR CAM and NCEP CFS LWR and SWR parameterizations. To calculate the error statistics presented in the following figures and tables of this section, the original parameterizations and their NN emulations have been applied to a validation data set. Two sets of the corresponding HR profiles have been generated for both LWR and SWR. Total and level bias (or mean error), total and level RMSE, profile RMSE or PRMSE, and σ_{PRMSE} have been calculated. Some of these statistics presented in tables of this section have been calculated as follows. The outputs of the original parameterization and the NN emulations can be represented as: $Y(i,j)$ and $Y_{NN}(i,j)$, respectively, where $i = (lat, lon)$, $i=1, \dots, N$ is the horizontal location of a vertical profile, N is the number of horizontal grid points, and $j = 1, \dots, L$ is the vertical index where L is the number of vertical levels.

The mean difference, B (bias or a systematic error of approximation), between the original parameterization and its NN emulation, is calculated as follows:

$$B = \frac{1}{N \times L} \sum_{i=1}^N \sum_{j=1}^L [Y(i,j) - Y_{NN}(i,j)] \quad (8)$$

The root mean square error has been calculated for each i^{th} profile:

$$prmse(i) = \sqrt{\frac{1}{L} \sum_{j=1}^L [Y(i, j) - Y_{NN}(i, j)]^2} \quad (9)$$

This error can be used to calculate mean profile root mean square error, $PRMSE$, and its standard deviation, σ_{PRMSE} :

$$\begin{aligned} PRMSE &= \frac{1}{N} \sum_{i=1}^N prmse(i) \\ \sigma_{PRMSE} &= \sqrt{\frac{1}{N-1} \sum_{i=1}^N [prmse(i) - PRMSE]^2} \end{aligned} \quad (10)$$

3.3.2 NCAR CAM

The NCAR CAM-2 (T42L26) was run for two years to generate representative data sets. The representative data set adequately samples the atmospheric state variability. The first year of simulation was divided into two independent parts, each containing input/output vector combinations. The first part was used for training and the second for tests (control of overfitting, control of NN architecture, etc.). The second year of simulation was used to create a validation data set completely independent of both the training and test data sets. The third part or the validation set was used for validations only. All approximation statistics presented in this section are calculated using this independent validation data set. The accuracy of the NN run, i.e., biases and rmse, are calculated against the control run.

Table 1 shows bulk validation statistics for the accuracy of approximation and computational performance for the best (in terms of accuracy and performance)

developed NN emulations: NN 50 ($k = 50$ hidden neurons in eq. (2)) for the LWR emulation and NN 55 ($k = 55$ hidden neurons in eq.(2)) for the SWR emulation.

The NN emulations developed for LWR and SWR are highly accurate. They have practically zero bias and a quite small PRMSE. Zonal mean differences between the NN emulation and the original parameterization for radiative fluxes at the top of the atmosphere and at the surface have also been produced. The differences appear to be uniformly small for all latitudes, mostly within $\pm 0.5 \text{ W/m}^2$ and do not exceed $\pm 1 \text{ W/m}^2$.

Table 1. Statistics estimating the accuracy of HRs (in K/day) calculations, and computational performance for NCAR CAM-2 LWR and SWR using NN emulation vs. the original parameterization.

	Bias (K/day)	PRMSE (K/day)	σ_{PRMSE} (K/day)	Performance (times faster)
LWR NN 50	$3. \cdot 10^{-4}$	0.28	0.20	150
SWR NN 55	$-4. \cdot 10^{-3}$	0.15	0.12	20

The NN emulations using 50 neurons in the hidden layer provide, if run *separately* at every model physics time step (1 hour), a speed-up of roughly 150 times for LWR and 20 times for SWR as compared with the original LWR and SWR, respectively. It is noteworthy that the main reason for the smaller performance gain for NN SWR vs. NN LWR is that the original CAM SWR parameterization is simpler and about 10 times faster than the original CAM LWR.

Using NN emulations *simultaneously* for LWR and SWR or for the full model radiation, results in an overall significant, 13-fold acceleration of calculations for the entire/full model radiation block. It is worth clarifying, for a better understanding of the overall speed-up, that for the usual control run the original LWR (including time-consuming optical properties calculations) is calculated *less frequently*, only every 12 hours or twice a day, and only computationally inexpensive heating rates and radiative fluxes are calculated every hour. Notice that all other inputs, including cloud cover, which is represented by a vertical profile of cloud fraction, are updated hourly. For the model run using NN emulations, LWR (including both optical properties and heating rates and radiative fluxes) is calculated *more frequently*, every hour, consistent with SWR and other model physics calculations. We also performed an additional costly control run with the original LWR calculated every hour, as it is done in the LWR NN run, for a limited period (10 years). The results of the two control runs appeared to be very close. The difference between them is significantly less than the difference between each of them and the LWR NN run. Because of that we decided to validate the 40 year full radiation NN run against the usual control run.

3.3.3 NCEP CFS

The NCEP CFS (T126L64) has been run for seventeen years to generate representative training data sets. The representative data set samples the atmospheric state variability adequately, i.e., it represents all possible states produced by the model as fully as possible (including the states introduced due to time dependent CO₂ concentration). All inputs and outputs of original LWR and SWR parameterizations

have been saved for two days per month, i.e., for one day at the beginning and one day in the middle of the month, every three hours (eight times per day) to cover the annual and diurnal cycles. From the three-hour global data set three hundred events (the set of input and output profiles) have been randomly selected. The obtained data set was divided into three independent parts, each containing about 200,000 input/output vector combinations. The first part has been used for training, the second one for tests (control of overfitting, control of NN architecture, etc.), and the third part (an independent validation data set) was used for validation of trained NN only. All approximation statistics presented in this section are calculated using the independent validation data set. The accuracy of the NN emulation, i.e., biases, rmse, etc. are calculated against the control (the original parameterization).

Table 2. Statistics estimating the accuracy of HRs (in K/day) calculations and the computational performance for NCEP CFS (T126L64) LWR and SWR using NN emulation vs. the original parameterization. For comparison, NCAR CAM (T42L26) LWR and SWR statistics are also shown. Total statistics show the bias, RMSE, PRMSE, and σ_{PRMSE} for the entire 3-D HR fields. Layer (for the top and bottom layers) statistics show the bias and RMSE for one horizontal layer (the top or bottom layer). Also, the changes in statistics due to applying the balancing procedure (see Appendix1) are shown for RRTWG LWR and SWR NN emulations. The NN complexity N_C and speedup η (how many times NN emulation is faster than the original parameterization) are shown. RRTMG and RRTMF are different versions of the radiation code developed by AER Inc.

Statistics Types	Statistics	LWR				SWR		
		NCAR CAM	NCEP CFS			NCAR CAM	NCEP CFS	
			RRTM G	Change due to Balancing	RRTM F		RRTM G	Change due to Balancing
Total Error Statistics	Bias	$3. \cdot 10^{-4}$	$2. \cdot 10^{-3}$	$6. \cdot 10^{-4}$	$7. \cdot 10^{-4}$	$-4. \cdot 10^{-3}$	$5. \cdot 10^{-3}$	$-3. \cdot 10^{-3}$
	RMSE	0.34	0.49	$1. \cdot 10^{-4}$	0.42	0.19	0.20	$-5. \cdot 10^{-3}$
	PRMSE	0.28	0.39	$3. \cdot 10^{-4}$	0.30	0.15	0.16	$-5. \cdot 10^{-3}$
	σ_{PRMSE}	0.2	0.31	$1. \cdot 10^{-4}$	0.30	0.12	0.12	$1. \cdot 10^{-3}$
Bottom Layer Error Statistics	Bias	$-2. \cdot 10^{-3}$	$-1. \cdot 10^{-2}$	$-6. \cdot 10^{-4}$	$6. \cdot 10^{-3}$	$-5. \cdot 10^{-3}$	$9. \cdot 10^{-3}$	$-8. \cdot 10^{-3}$
	RMSE	0.86	0.64	$1. \cdot 10^{-5}$	0.67	0.43	0.22	-0.01
Top Layer Error Statistics	Bias	$-1. \cdot 10^{-3}$	$-9. \cdot 10^{-3}$	$6. \cdot 10^{-4}$	$2. \cdot 10^{-3}$	$2. \cdot 10^{-3}$	$1.3 \cdot 10^{-2}$	$4. \cdot 10^{-3}$
	RMSE	0.06	0.1782	$4. \cdot 10^{-3}$	0.09	0.17	0.21	$1. \cdot 10^{-3}$
NN Complexity	N_c See eq. (2)	12,733	33,294	-	93,969	11,418	45,173	-
Speedup, η	Times	150	12	-	21	20	45	-

Table 2 shows bulk validation statistics for the accuracy of approximation of heating rates (HR) and the computational performance for the best (in terms of both the accuracy and performance) developed NN emulations for the NCEP CFS LWR and

the SWR. Total statistics show the bias, RMSE, PRMSE, and σ_{PRMSE} (for definitions see section 3.3.1) for the entire 3-D HR fields. Also, layer statistics for the top and bottom atmospheric layers are included to illustrate the accuracy of NN emulations in the areas of the increased non-linearity (Morcrette et al. 2008). Although the two models (as well as their embedded radiation parameterizations) are different, comparisons between NCAR CAM (with 26 vertical layers) and NCEP CFS (with 64 vertical layers) allow us to observe a general dependence of the NN accuracy on the model vertical resolution (see also error profiles shown in Fig.1).

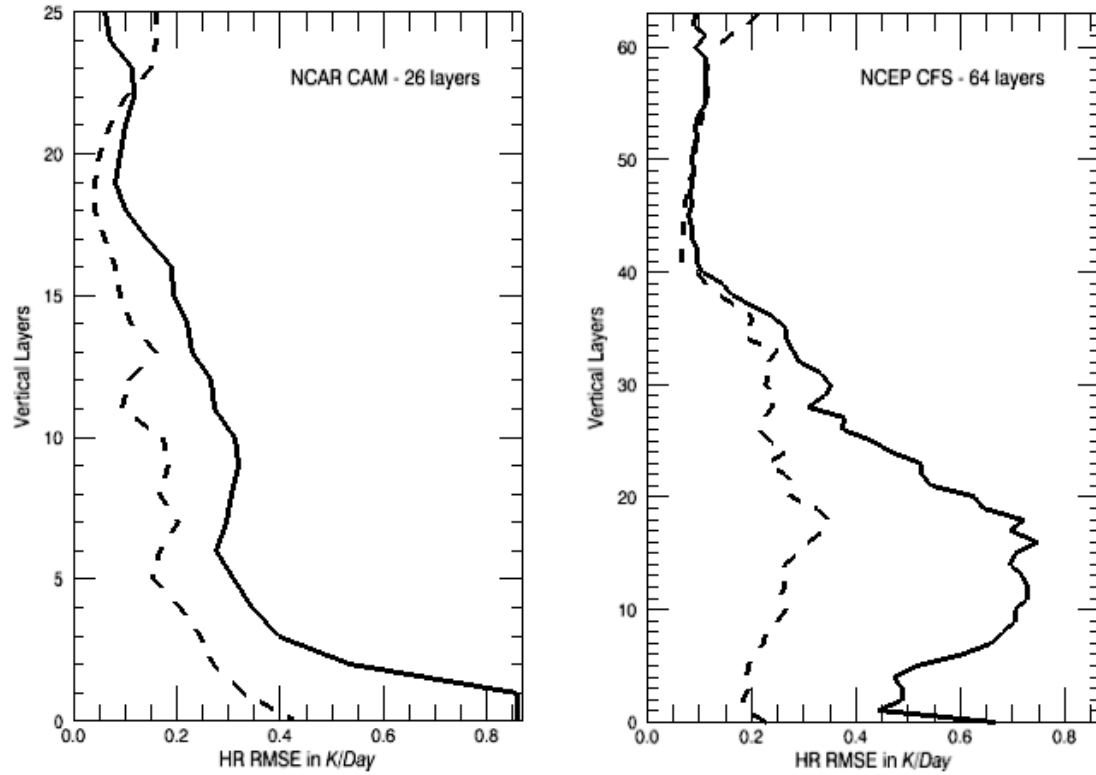


Fig.1 Vertical distributions of NN emulation errors for two models: NCAR CAM (26 vertical layers) and NCEP CFS (64 vertical layers). Solid line corresponds to LWR

and dashed line to SWR. The errors and their vertical distributions are similar, i.e. in the same bulk part, for both models.

As can be concluded from Table 1 and Fig.1, NN emulations for both LWR and SWR handle very well the nonlinearity at the top of the atmosphere where biases and RMSEs are very small with RMSEs being even smaller than the total RMSE. At the bottom layer, the non-linearity does not cause significant increases in biases; the RMSEs increase about two times, and as compared with the total RMSE, remain sufficiently small. It is noteworthy that in the context of our discussion “sufficiently small errors” means that the errors of such a magnitude have almost negligible impacts on model behavior. Only validation of NN emulations in parallel model runs allows us to make final conclusion about the sufficient smallness of the approximation errors.

It terms of presented statistics, there are practically no differences between NCAR CAM with 26 vertical layers and NCEP CFS with 64 vertical layers. As shown in Fig.1, the entire vertical distributions of errors (for both LWR and SWR) are similar for these two models. Thus, the accuracy of our NN emulation approach does not depend significantly on vertical resolution of the model. It does depend on the vertical location of the atmospheric layer. The layer RMSE increases near the surface for both models.

Also, the NN complexity N_C and speedup η (how many times NN emulation is faster than the original parameterization) are shown in Table 2. These characteristics complement our discussion on the dependence of the speedup on vertical resolution (see the end of the section 2.2.1) For the LWR parameterization, we see a significant decrease of the speedup for NCEP CFS with 64 vertical layers vs. NCAR CAM with 26 vertical layers although the LWR NN emulation for NCEP CFS is still 12 times faster than the original parameterization. For the SWR parameterization the opposite tendency is observed; that is, the speedup for NCEP CFS SWR NN is more than two times higher than that of NCAR CAM SWR NN.

These seemingly contradictory speedups for LWR and SWR emulations can be explained by the interplay of the two main contributing factors: the physical complexity of the radiation calculation itself (the number of treated species, spectral bands, parameterization schemes, etc.), and the dependence of the particular numerical scheme implemented in the radiative transfer on the number of vertical model layers. The results presented in Table 2 illustrate the fact that the numerical scheme implemented in the NCEP CFS RRTMG-LW parameterization is significantly more efficient (linear with respect to the number of vertical levels L) than that of the original NCAR CAM LWR parameterization (quadratic with respect to L). Thus a smaller speedup factor is produced by the NN emulation for NCEP LWR than that for NCAR CAM. The NCEP CFS's RRTMG-SW includes more spectral bands and g-points and uses more complex treatment for a larger variety of absorbing/scattering species; thus NN shows a larger speedup value η than that of

NCAR CAM. In any case, our NN emulation approach is significantly less dependent (in terms of both the accuracy and speed-up) on the increase of vertical resolution than the NN based LWR parameterization NeuroFlux for which at vertical resolution of 60 layers and more, both accuracy and speed-up could not be achieved simultaneously (Morcrette et al. 2008). For our NN emulation approach, for the model with 64 vertical layers, the desired accuracy of the NN emulation could be achieved simultaneously with a significant speed up of 12 times for the LWR and of 45 times for the SWR parameterizations.

Using NN emulations *simultaneously* for LWR and SWR or for the full model radiation results in an overall significant, about 20 – 25% speedup of NCEP CFS climate simulations when both LWR and SWR are calculated every hour. The speedup η provided by NN emulations (see Table 2) can be also used for more frequent calculations of model radiation.

3.4 Validation of Parallel Decadal Model Simulations

3.4.1 NCAR CAM

. The results of multi-decadal climate simulations performed with NN emulations for both LWR and SWR, i.e., for the full model radiation, have been validated against the parallel control NCAR CAM simulation using the original LWR and SWR. Below we estimate closeness of the results for these parallel 50-year climate simulations. Note that the first 10 years of simulations are not included in the validation to avoid the impact of spin-up effects, so that years 11-50 are used for the validation. The

spin-up is done for the original NCAR CAM; it is not related to the use of NN emulations. We will analyze below the differences between the parallel runs in terms of time and spatial (global) means as well as temporal characteristics.

Table 3 presents comparisons between the parallel control and NN emulation runs in terms of the time (40-year) and global mean characteristics and the differences between the results of the parallel runs. Basically, the differences, in terms of mean, rms, minimum and maximum characteristics, between the parallel runs, are small.

More specifically, there are negligible mean differences (bias), 0.02 hPa and -0.1 K, in sea level pressure and 2-meter temperature, respectively, between the NN and control runs. For these fields, rmse, minimum and maximum differences are also small. Other time and global mean differences presented in Table 3, including such sensitive fields as total precipitation, total cloud amount, cloud amounts for high, low, and mid clouds, total grid-box cloud liquid and ice water paths, top of model net long-wave flux and cloud forcing, also show a close similarity, in terms of all presented difference characteristics, between the parallel simulations for these fields. These differences are within typical observational and reanalysis errors/uncertainties. Note that minimum and maximum differences in Table 3 are not averaged in space and time but rather are instantaneous grid point values obtained for the entire 40-year simulation.

Let us discuss the differences between the parallel simulations in terms of spatial and temporal characteristics. Zonal and time mean heating (or cooling) rates for LWR and SWR are presented in Figs. 2 and 3, respectively.

Table 3. Time (40-year) and global means for model diagnostics from NCAR CAM-2 control climate simulations with the original LWR and SWR, simulation with NN emulations for the full radiation using NN 50 (LWR) and NN 55 (SWR), and their differences. SLP – sea level pressure; T2M – temperature at 2 m ; U-200 - 200 hPa zonal wind; TPR - total precipitation rate; TCA - total cloud amount; HLCA - high-level cloud amount; LLCA - low-level cloud amount; MLCA - mid-level cloud amount; TGCLWP - total grid-box cloud liquid water path; TGCIWP - total grid-box cloud ice water path; TOMNLW - top of model net long-wave flux; TOMLWC - top of model long-wave cloud forcing.

Field	Control	NN Full Radiation	Mean Difference	RMS Difference	Min Difference	Max Difference
SLP (hPa)	1011.48	1011.50	0.02	0.52	-2.04	1.57
T2M (K)	287.37	287.27	- 0.1	0.26	-1.64	0.78
U-200 (m/s)	16.21	16.29	0.08	0.86	-2.31	3.95
TPR (mm/day)	2.86	2.89	0.03	0.2	-1.84	1.19
TCA (%)	60.71	61.12	0.41	1.42	-7.50	5.76
HLCA(%)	43.05	43.29	0.24	1.63	-7.52	8.01
LLCA(%)	31.67	31.93	0.26	1.06	-5.20	4.78
MLCA(%)	19.11	19.14	0.03	0.81	-4.86	4.39
TGCLWP (g/m²)	60.23	60.59	0.36	3.02	-19.43	14.95
TGCIWP (g/m²)	8.82	8.83	0.01	0.39	-1.69	1.45
TOMNLW (W/m²)	234.48	234.54	0.06	2.32	-8.37	11.56
TOMLWC (W/m²)	29.33	29.07	-0.26	2.45	-15.59	7.64

The HR patterns (the upper panels) are practically indistinguishable and their differences (the bottom panels) are small. It confirms that the NN emulations for LWR and SWR are very close to their original parameterizations throughout the model simulations. It is noteworthy that the HR differences in SWR and especially in LWR are a bit larger at the surface because HRs are larger there (Figs. 2 and 3). For the zonal means it is not easy to distinguish between the ocean and land. However, the differences seem to be larger over the mountainous Antarctica region (60° S to 90° S) as well as over the Northern Hemisphere mid-latitudes where the major mountains are located (such as those in Europe, Asia and North America).

Fig. 4 shows a very close similarity in zonal and time mean 2-meter temperature for the parallel simulations (the upper panel) where their differences are within the -0.6 K to 0.3 K range (the bottom panel).

ANN

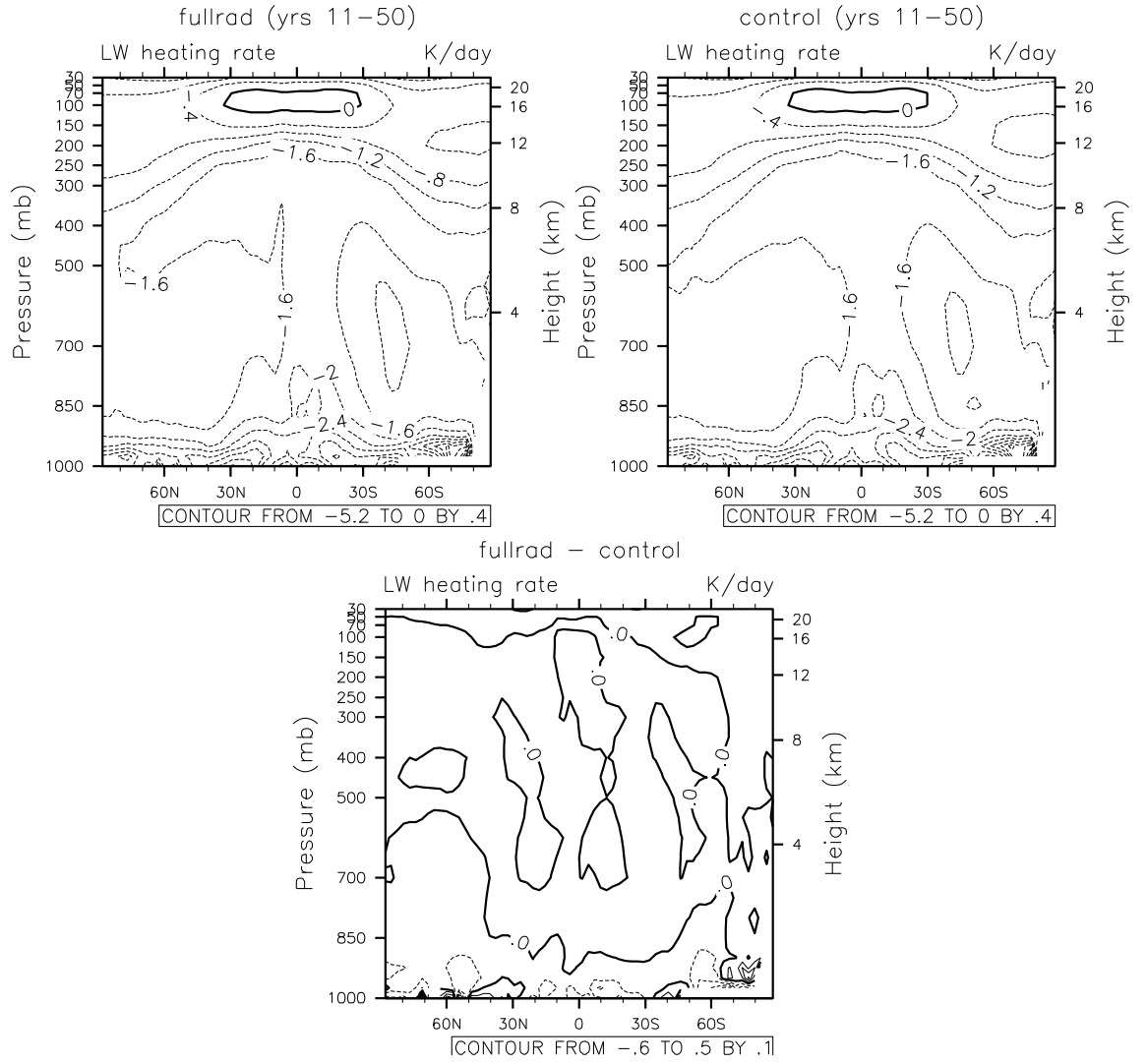


Fig. 2 Zonal and time mean LWR Heating Rates, in K/day, for the NN LWR run (the upper left panel), the control run (the upper right panel) and their difference (the bottom panel).

ANN

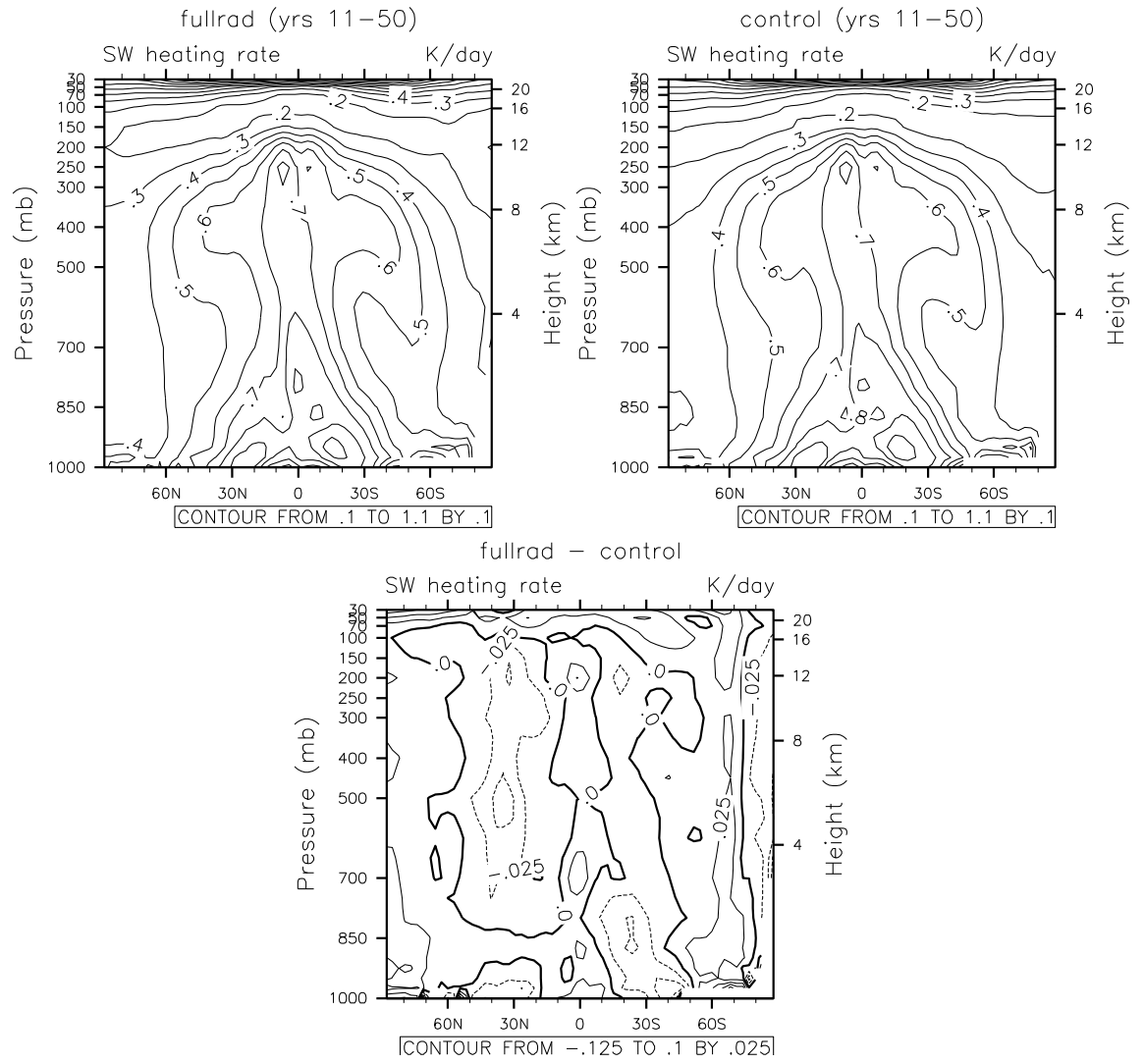


Fig. 3 Same as in Fig. 2 but for the SWR.

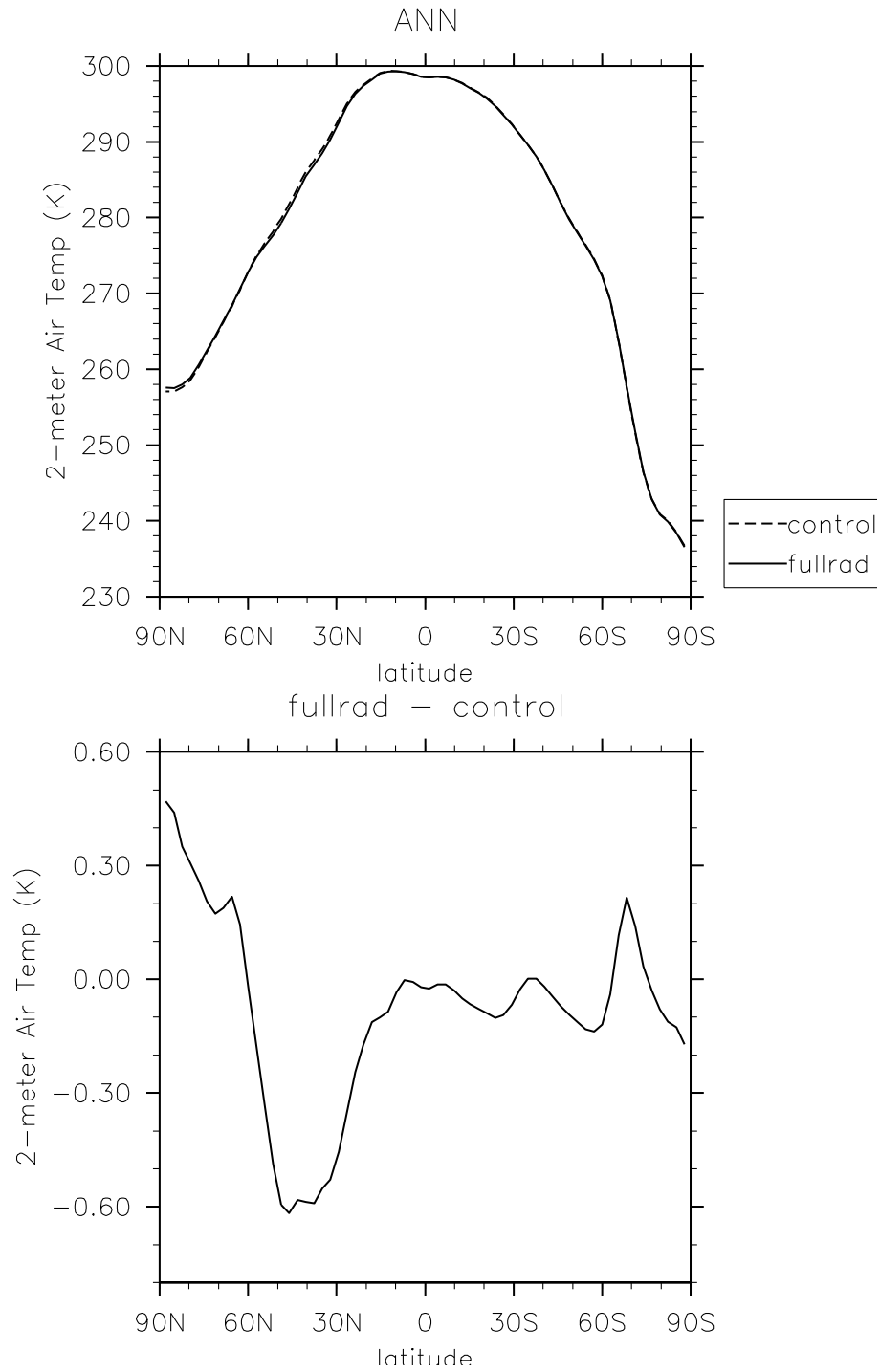


Fig. 4 Zonal and time mean 2-meter temperature, in K, for the full radiation NN and control runs (the upper panel) and their difference (the bottom panel).

ANN

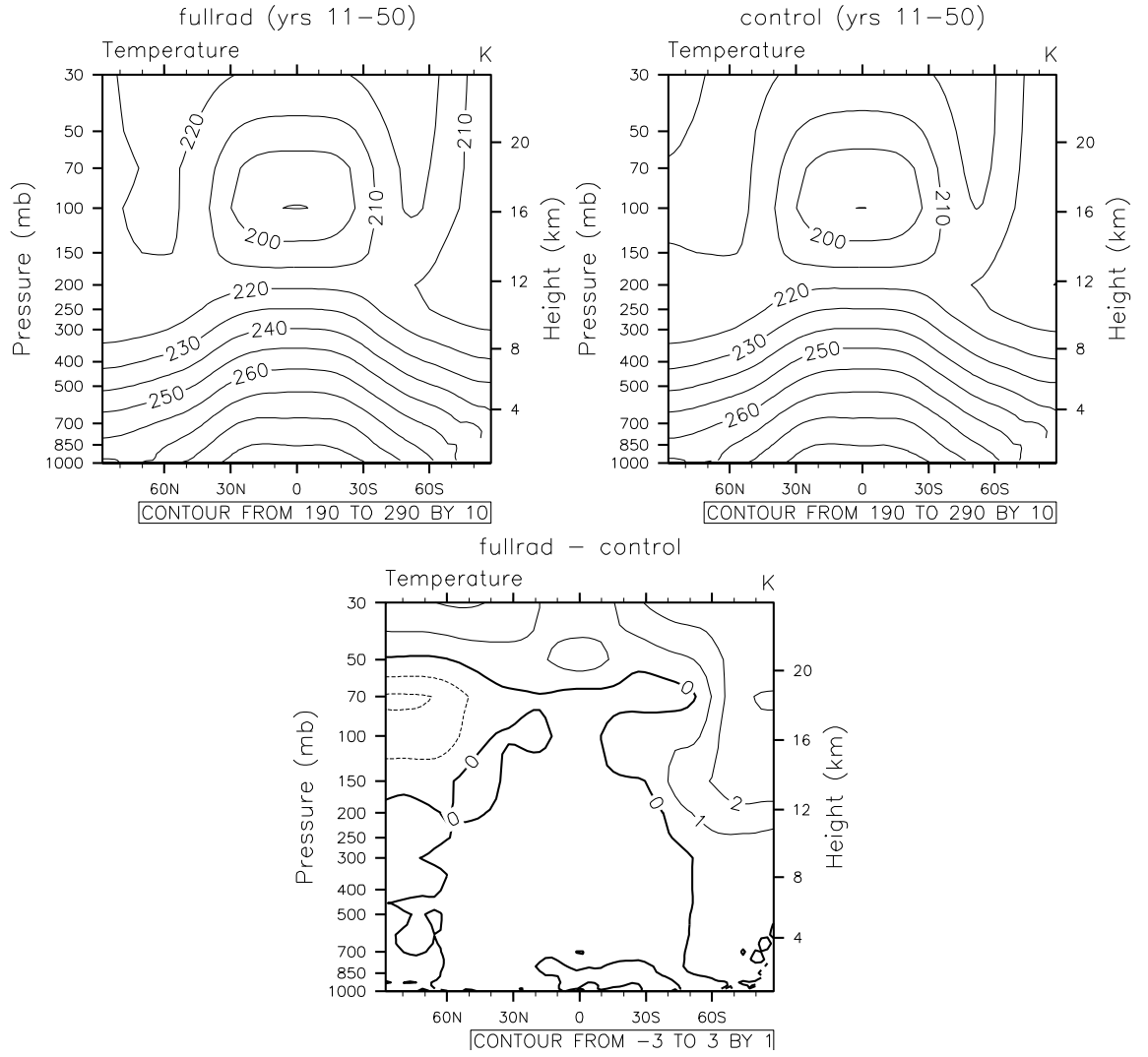


Fig. 5 Zonal mean vertical distribution of time mean temperature, in K, for the full radiation NN run (the upper left panel), the control run (the upper right panel), and their difference or bias (the bottom panel).

The zonal and time mean vertical distributions of temperature for the parallel runs (Fig. 4) are close to each other and their difference or mean bias is practically zero, with minimum and maximal biases within ~ 2 -2.5 K by magnitude. This larger zonal bias occurs in the stratosphere mostly over the Southern polar domain. However, it is comparable with typical observational and/or reanalysis errors/uncertainties (just as a reference) and also comparable with the differences between the NCEP and ECMWF reanalysis.

Close similarities have also been obtained for the results of parallel runs in terms of time mean spatial fields such as 850 hPa temperature presented in Fig. 5. The horizontal fields presented in the upper and middle panels are close to each other. For the difference field (the bottom panel), bias is negligible (-0.06 K), RMSE is small (0.34 K), and minimum and maximum values (~ -1.6 K and ~ 0.9 K) are well within observational or reanalysis errors/uncertainties.

In addition to global distributions such as shown in Fig. 5 it is important to assess the differences between the parallel simulations at a local (station) level, an example of which is presented in Fig. 6. The vertical distributions of time mean temperature are very close for both runs at the local level as well (Fig. 7).

Now we compare the results of the parallel simulations in terms of temporal characteristics. Fig. 8 shows the winter-summer differences for time mean temperature at 850 hPa. Their patterns are practically indistinguishable and the minimum and maximum values are very close. The global mean time series for time mean temperature at 850 hPa presented in Fig. 9 are very similar throughout the entire decadal simulations for the parallel runs, with only occasional small differences

(within 0.5 K) that are well below the observation and reanalysis errors. The annual cycle for global mean temperature at 850 hPa is presented in Fig. 10. It shows very small differences between the runs, with the maximum within 0.2 K for January. The precipitation annual cycles shown on Fig. 11 are very close for both runs (the upper panels) and their differences or bias (the bottom panel) is quite small. Close similarity has also been obtained for other model prognostic and diagnostic fields in term of their spatial and temporal characteristics.

The results obtained confirm the profound similarity in parallel climate simulations, which justifies the possibility of using efficient neural network emulations of full model radiation for decadal and longer climate simulations as well as for weather prediction models. The methodology developed can be applied to other LWR and SWR schemes used in a variety of models, process studies, and other applications.

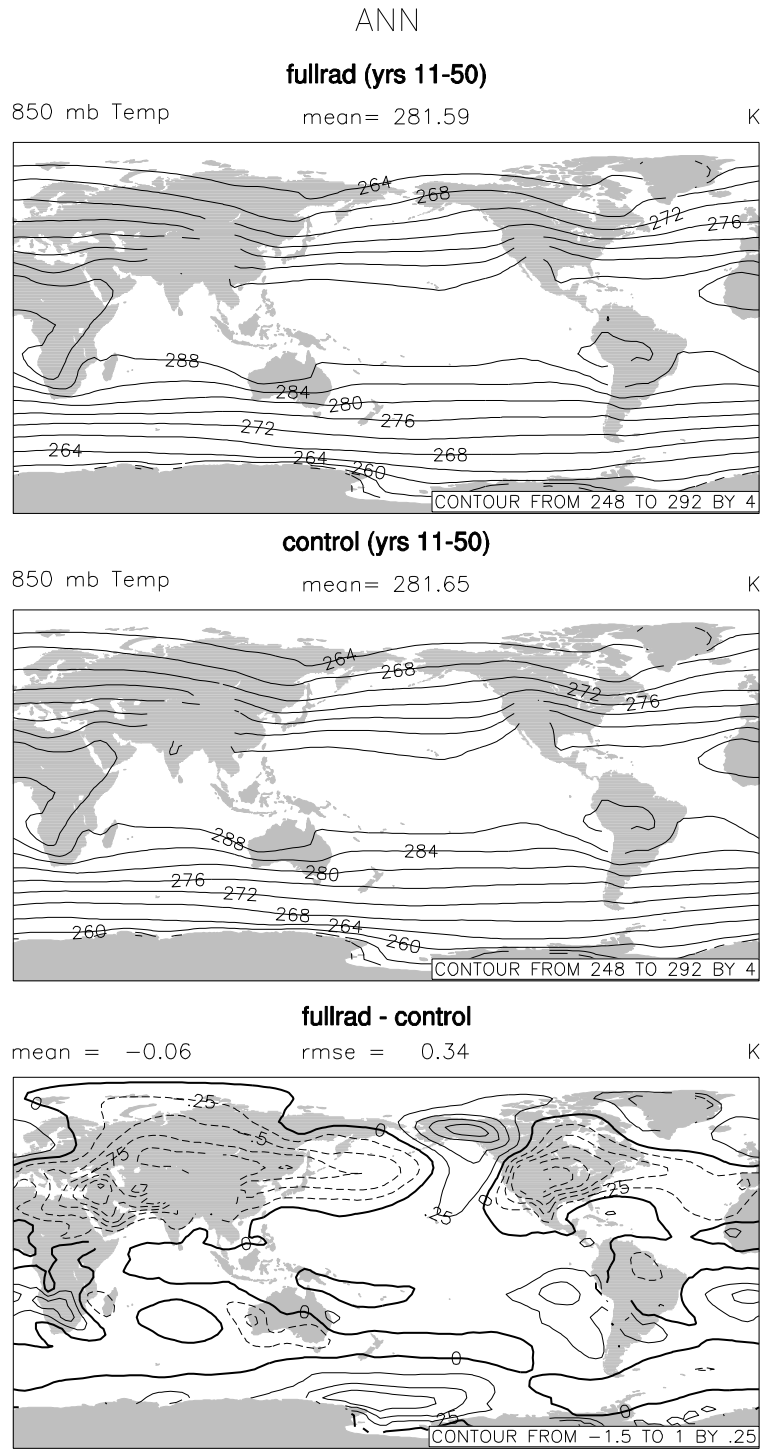


Fig. 6 Time mean temperature at 850 hPa, in K, for the full radiation NN run (the upper panel), the control run (the middle panel), and their difference (the bottom panel).

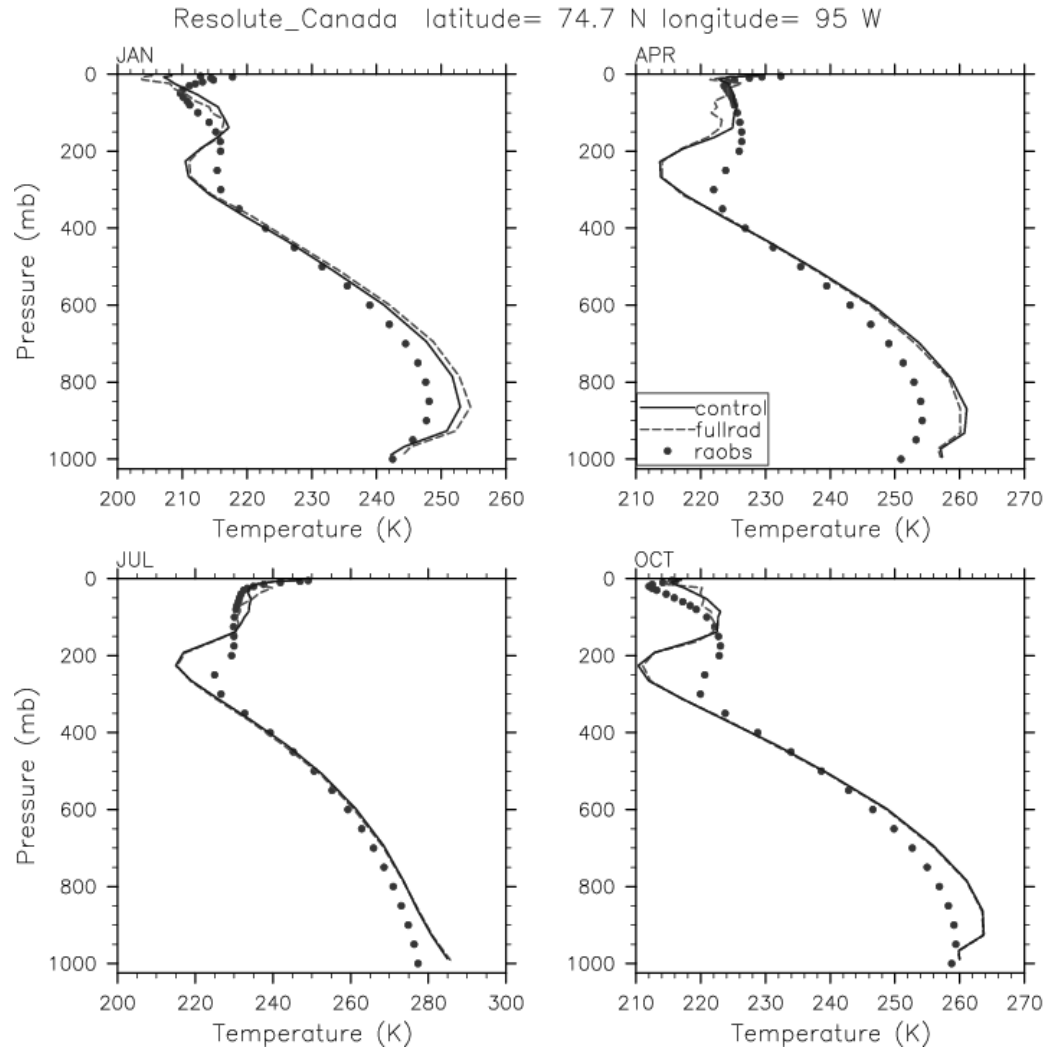


Fig. 7 Vertical profile of time mean temperature, in K, at the Resolute, Canada station for the full radiation NN run (the dashed line), the control run (the solid line), and observations (the dotted line).

DJF-JJA

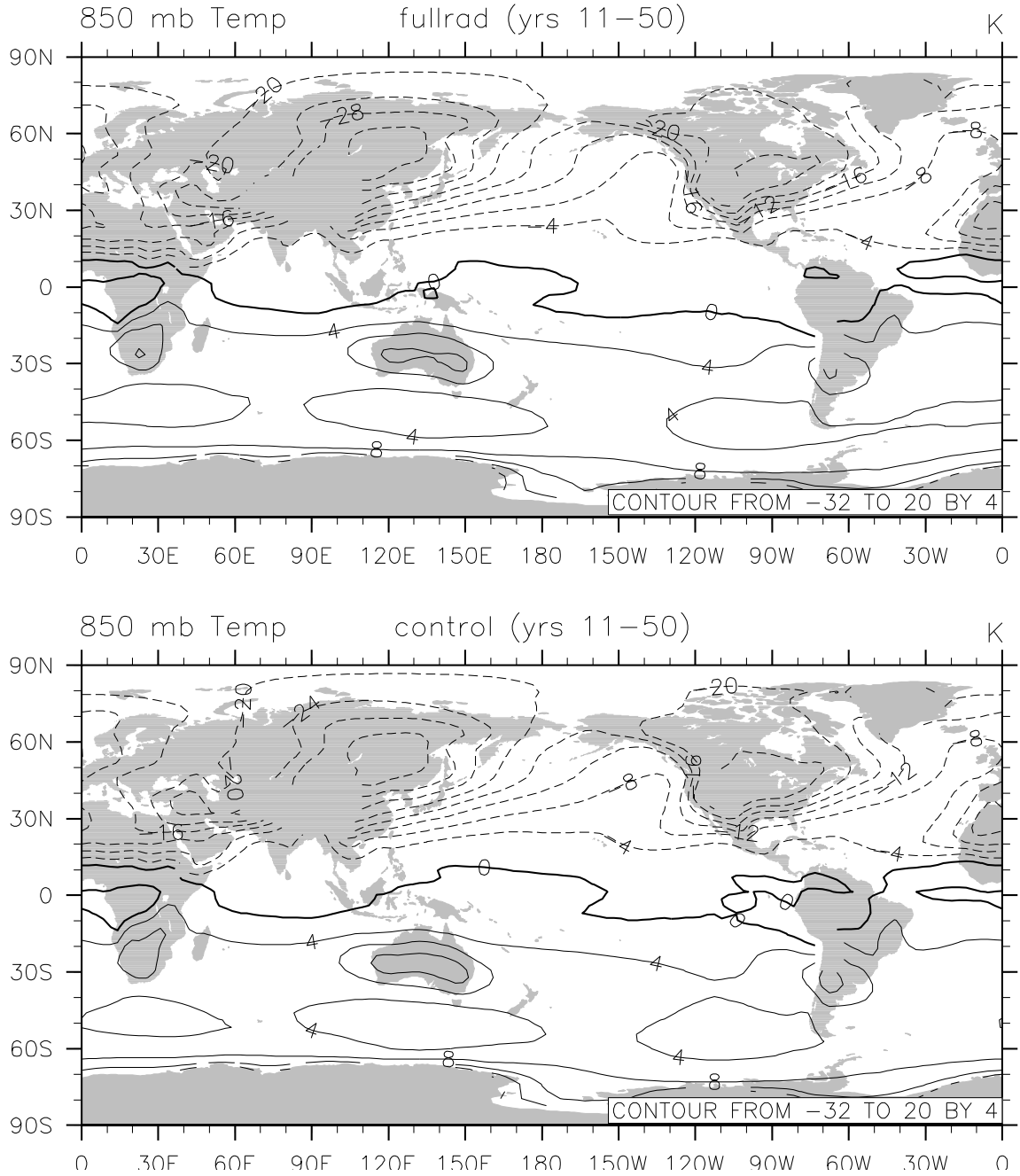


Fig. 8 Winter-summer difference for time mean temperature at 850 hPa, in K, for the full radiation NN run (the upper panel), and the control run (the bottom panel).

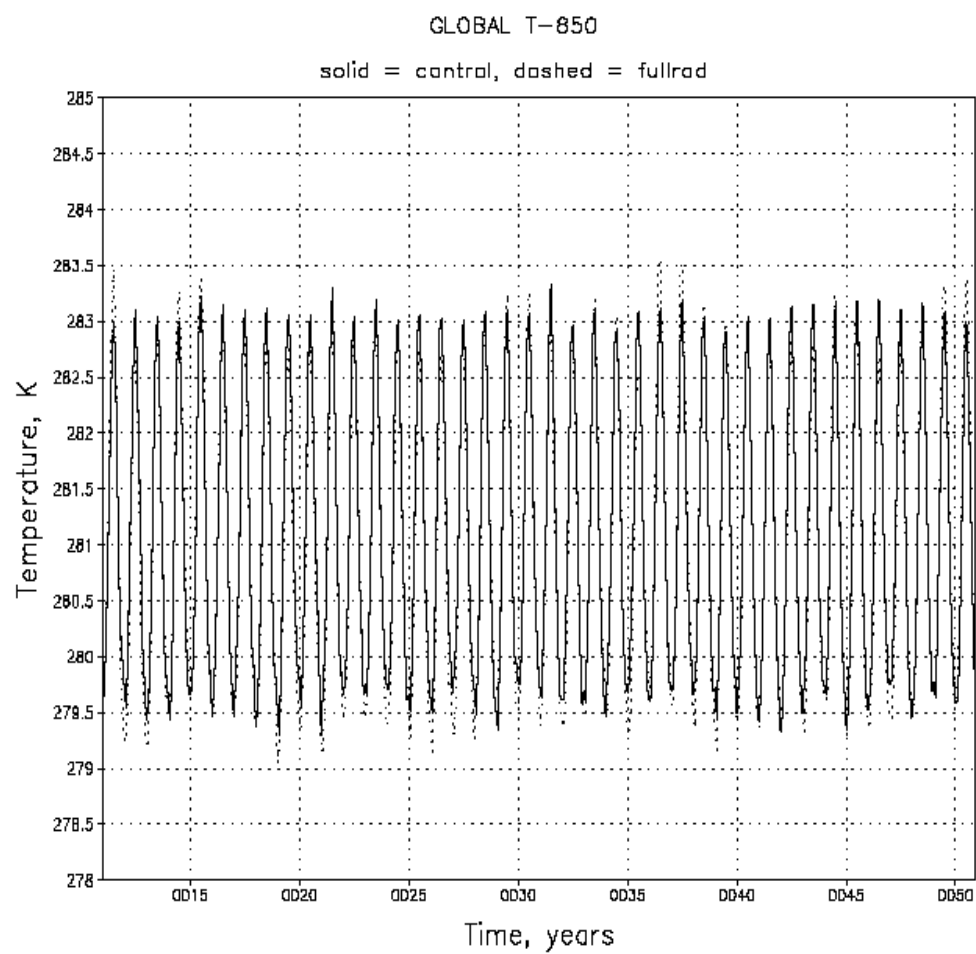


Fig. 9 Global mean time series for time mean temperature at 850 hPa, in K, for the full radiation NN run (the dotted line), and the control run (the solid line).

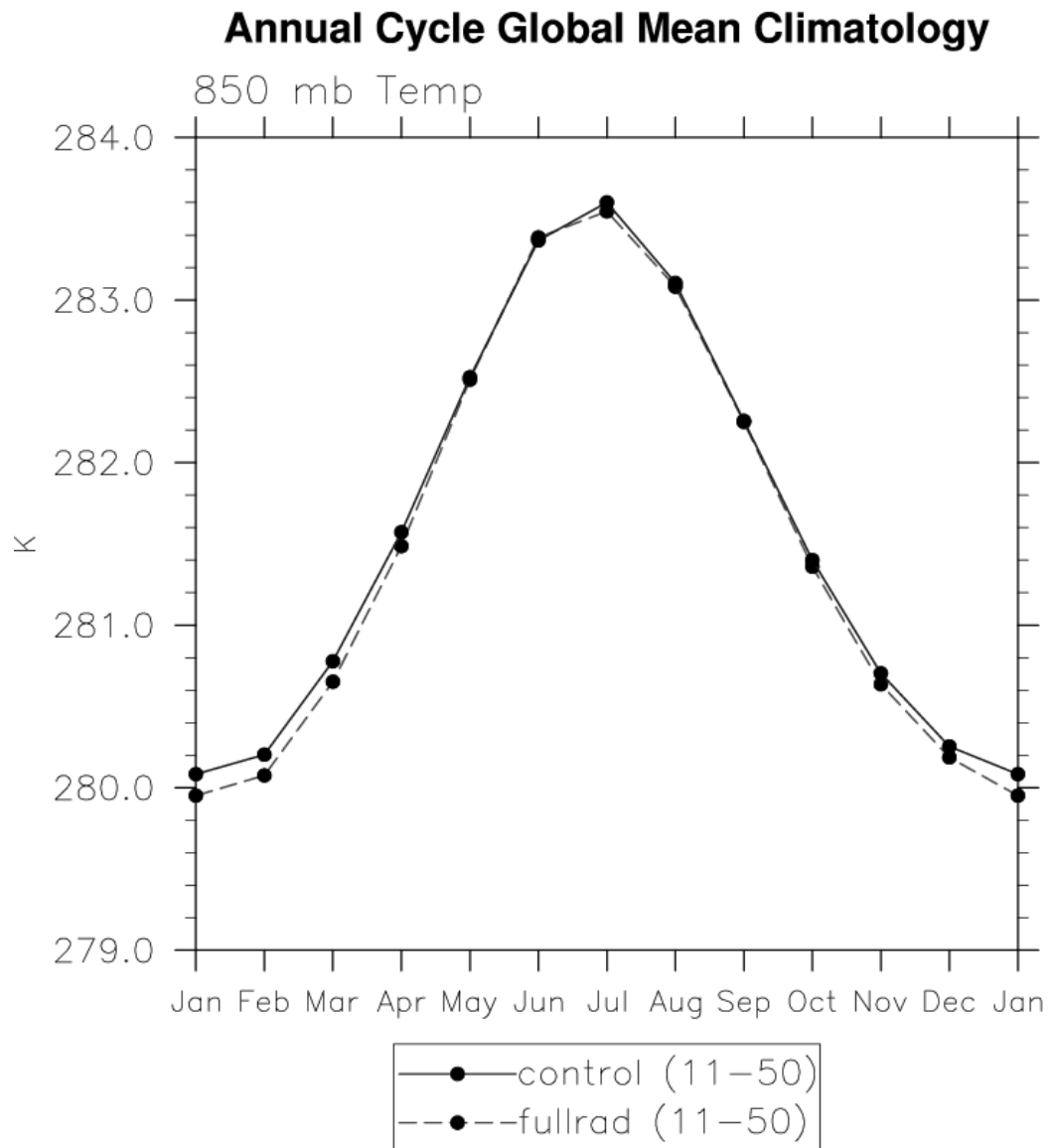


Fig. 10 Annual cycle for global mean temperature at 850 hPa, in K, for the full radiation NN run (the dashed line), and the control run (the solid line).

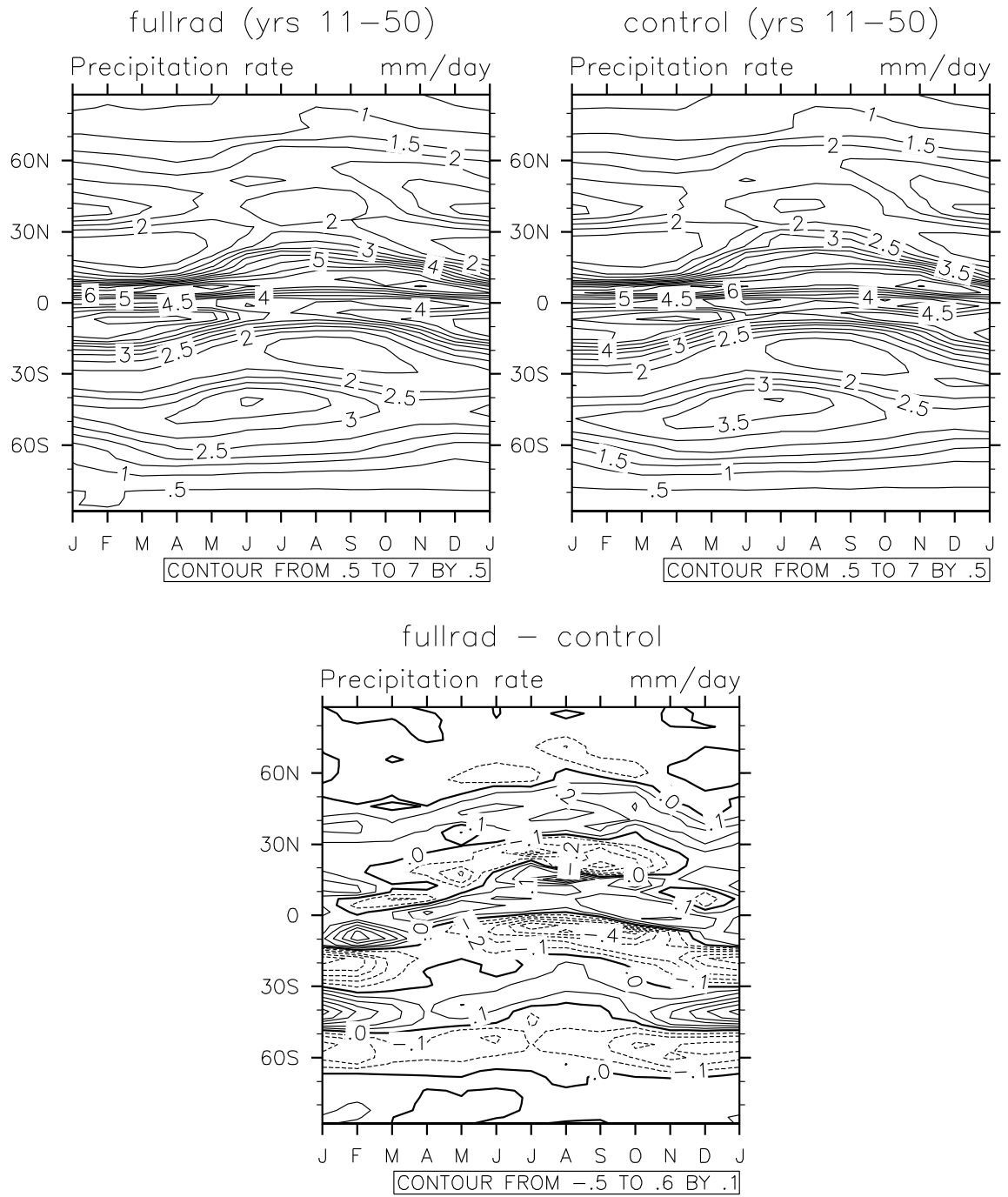


Fig. 11 Annual cycle for precipitation, in mm/day, for the full radiation NN run (the upper left panel), the control run (the upper right panel), and their difference or bias (the bottom panel).

3.4.2 NCEP CFS

As in the previous section 3.4.1 on the results of decadal NCAR CAM climate simulations we show in this section the differences between the decadal and seasonal parallel runs for NCEP CFS. To evaluate the NN induced changes, we compare them with such commonly used measures as observation errors or uncertainties of reanalysis. We show that the differences are smaller than these quantitative measures.

In order to emphasize how small the changes introduced by the use of NN emulations are, we also find it appropriate to use a measure derived from the model itself, namely, the *model's internal variability*. Because a GCM is an essentially nonlinear system, it may produce something like a “butterfly effect”, that is a significant reaction/response even to small perturbations in the model or in the model computational environment (e.g. routine changes in computer hardware, operational system, compilers, libraries, etc.). Any, even infinitesimal change in model formulation, initial conditions or computational environment makes two model integrations diverge, with the effect that after the deterministic predictability is lost (which takes just weeks for the atmosphere, although longer for the ocean), the timing and location of weather patterns becomes essentially independent for the two integrations. Hence the two runs provide, in essence, two independent samples of the model's climatology, and their difference represents the model's internal variability.

Thus, we can state that the approximation error of NN emulation is negligible and, therefore, NN's accuracy is sufficient for the use in the model if the

differences/changes introduced in the model results by using the NN emulation are of the same order of magnitude as the aforementioned model's internal variability.

To estimate the model's internal variability, we ran two control runs with the original NCEP CFS model configuration, i.e., without NNs. The first run was performed before and the second run after the routine changes (introduced quasi-regularly by system administrators) of the version of the FORTRAN compiler and libraries. Small differences between these two runs (which are similar to those due to changes in a computer operation system and/or in hardware (Moorthi 2009)) are shown below together with the differences between the parallel NN and control runs for comparison purposes, as an additional measure of the NN emulation accuracy. Presenting model's internal variability helps us to better evaluate the differences in climate simulations caused by using NN emulations for model radiation and to emphasize how small these differences are.

3.4.2.1 Climate Simulations

The results of 17-year (1990-2006) climate simulations performed with NN emulations for both LWR and SWR, i.e., for the full model radiation, have been validated against the parallel control NCEP CFS simulation using the original LWR and SWR. We analyze the differences between the parallel runs in terms of spatial (global) means as well as temporal characteristics.

Let us discuss first the differences between the parallel simulations in terms of spatial and temporal radiation characteristics. The differences between the NN radiation and

control runs and the differences between two control runs for zonal and time mean LWR and SWR fluxes are presented in Fig. 12. The upper row of Fig. 12 shows the differences for zonal and time mean top of atmosphere upward long (left panel) and short (right panel) wave fluxes (in W/m^2) for winter. The lower row of Fig. 12 shows the differences for zonal and time mean downward (left panel) and upward (right panel) surface long wave fluxes (in W/m^2). For the fluxes presented in Fig. 12, both the differences between the NN radiation and control runs and the differences between two control runs are small and similar by magnitude. They do not exceed $2\text{-}3 \text{ W/m}^2$, i.e., they are within observational errors and uncertainties of reanalysis (e.g. Kalnay et al. 1996, Kistler et al. 2001). The similarity of the differences by magnitude means that both the differences between the NN radiation and control runs are comparable with the model's internal variability. The HR differences are also very close in magnitude to (and do not exceed) the model's internal variability.

Let us discuss now prognostic and diagnostic characteristics such as SST, precipitation, different types of clouds, and time series that are potentially sensitive to changes in the model resulting from using NN emulations. Close similarities have also been obtained for these results of parallel runs in terms of time mean spatial fields, which are presented in Figs. 13 to 17. These figures contain two columns: the left column shows results for winter (December-January-February) and the right column for summer (June-July-August). The upper row panels (a) and (d) show fields produced by the full radiation NN run, the middle row panels (b) and (e) show mean errors/bias or the difference between the full radiation NN run and the control run

(CTL), NN-CTL, and the lower row panels (c) and (f) show the differences between two control runs (i.e., model's internal variability), CTL1-CTL, presented for comparison. Notice that spatial (global) and time mean errors/biases and RSMEs are shown in the panel titles for NN-CTL and CTL1-CTL.

The 17-year (1990-2006) time-mean SST distributions and bias/differences for the full radiation NN run vs. the control run and the differences between two control runs (model's internal variability) are presented for summer and winter in Fig. 13. The SST bias and RMSE for NN-CTL are very small; they are not larger than those of the model's internal variability, CTL1-CTL. The time and global mean errors/biases are near zero and RMSEs are just a small fraction of K. The results for other two seasons (spring and fall) are similar.

Fig. 14 shows the 17-year (1990-2006) time-mean distributions and bias/differences for total precipitation (PRATE) for the parallel full radiation NN and control runs for summer and winter, respectively. The PRATE bias is quite limited and occurs mostly in the tropics; it is also very close in magnitude (as well as RMSE) and pattern to the model's internal variability. The results for other seasons are similar.

Figs. 15 to 17 show comparisons for the parallel full radiation NN and control runs for different types of clouds. They present the 17-year (1990-2006) time-mean distributions and bias/differences of total clouds (Fig. 15), convective precipitation clouds (Fig. 16), and boundary layer clouds (Fig. 17) for summer and for winter. Clouds are very sensitive to any changes in the model and, therefore, provide a suitable and sensitive estimate of the accuracy of NN emulations.

For all types of clouds shown in Figs. 15 through 17, the cloud patterns and bias/differences for parallel total radiation NN and control runs are very close for both seasons presented. The situation is similar for other seasons and types of clouds (such as low, mid, and upper level clouds). The bias is very small and occurs mostly in the tropics. It has the same magnitude (as well as RMSE) and pattern as the differences between two control runs or model's internal variability shown for comparison. For all presented clouds the time and global mean errors/biases are near zero, just $\sim 0 - 1\%$, and RMSEs are just $\sim 1 - 2.5 \%$.

Let us compare now the results of the parallel NN and control runs in terms of temporal characteristics. The global mean time series for monthly means of the total precipitable water (PWAT), with the seasonal cycle subtracted, are presented in Fig. 18. The figure shows the 17-year (1990-2006) time series for the parallel full radiation NN run (the dash-dotted line) and for two control runs (the solid line for CTL and the dotted line for CTL1). The time series for PWAT presented in Fig. 18 for the parallel full radiation NN and the control run, CTL, show an overall similarity for the entire 17-year (1990-2006) period. The differences between two control runs are similar but marginally larger. The total global and time means for PWAT are very close for the parallel runs: 25.48 mm/day for the control (CTL) run, 25.62 for another control (CTL1) run, and 25.64 mm/day for the NN run.

Fig. 19 shows the 17-year (1990-2006) time series for the Nino3.4 index for the parallel full radiation NN and the two control runs. The Nino3.4 index is calculated

over the small area in the equatorial Pacific Ocean shown by the black rectangle in Figs. 13 – 17. The upper panel shows the Nino3.4 index calculated from reanalysis (CDAS), the control runs (the old control – the second panel from the top, and the new control – the second panel from the bottom) and the full radiation NN run (the bottom panel). The time series for the Nino3.4 index are affected by a quite limited SST anomaly sampling for the relatively small area and are very sensitive to any changes in the model or in its computational environment as can be seen from Fig. 19. The explanation for the different details of the Nino3.4 time series is that timing and magnitude of ENSO events is “chaotic” and subject to different phases of internal variability in the different runs. As can be seen from the standard deviation values included in Fig. 19, the model overestimates the ENSO variability compared with CDAS reanalysis. The overall dissimilarity of the indexes or their deviation from CDAS is not larger than that of the two control runs from CDAS and from each other.

Fig. 20 shows the 17-year (1990-2006) time series for global mean temperature at 850 hPa for the parallel full radiation NN and the two control runs. All three time series are close to each other; the differences do not exceed 0.5 K. The small differences between the full radiation NN and control runs are of the same magnitude as those of between two control runs.

The time-mean simulated products presented in Figs. 13-17 as well as other model simulated products show that biases and RMSE for the full radiation NN run are small, i.e., are overall within the observational errors or uncertainties of reanalysis, and are of a similar magnitude as the model’s internal variability.

Close similarity has also been obtained for other model prognostic and diagnostic fields in term of their spatial and temporal characteristics. Summarizing, from the obtained validation results, we can conclude that the differences between decadal climate simulations produced by the parallel full radiation NN and control runs are overall within or less than the observation errors and uncertainties of reanalysis (e.g. Kalnay et al., 1996). Moreover, these differences (both in terms of bias and RMSE) are of a similar magnitude as the model's internal variability or the differences between two control runs, which are regularly introduced in climate models by routine changes in computer environment (like changes in hardware, operational system, and/or compilers).

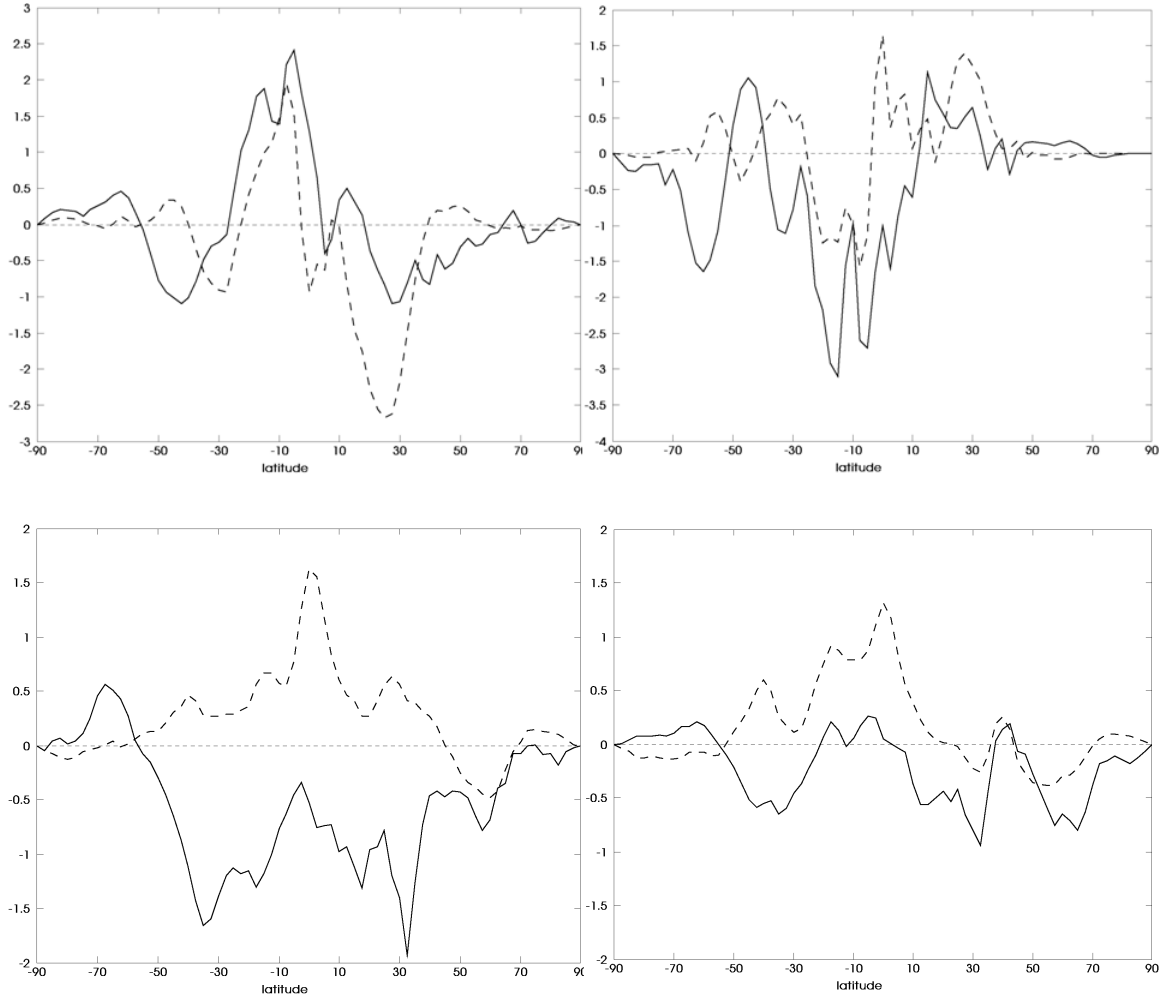


Fig. 12 The upper row: zonal and time mean Top of Atmosphere Upward Long (left panel) and Short (right panel) Wave Fluxes (in W per m^2) for the winter. The solid line – the difference (the full radiation NN run – the control (CTL)), the dash line – the differences between two control runs presented for comparison. The lower row: zonal and time annual mean downward (left panel) and upward (right panel) Surface Long Wave Flux (in W/m^2). The fluxes' differences are multiplied by $\cos(lat)$ to equalize the areas.

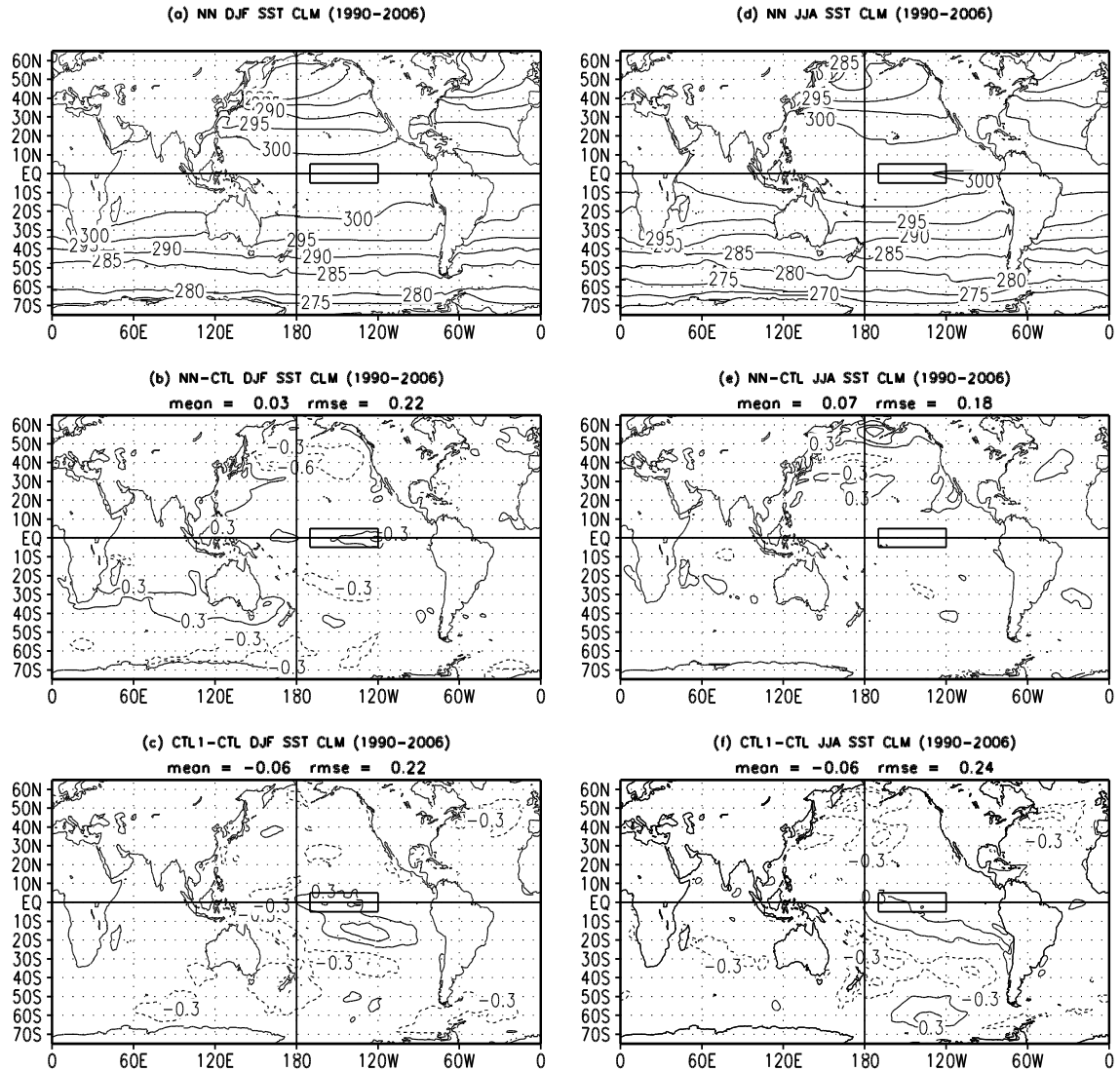


Fig. 13 The 17-year (1990-2006) time-mean (NN run) SST distributions and bias/differences for winter (DJF: December-January-February, left column) and for summer (JJA: June-July-August, right column) for the full radiation NN run vs. the control run. The upper row panels show full radiation NN runs. The middle row panels show bias or the difference (full radiation NN run – CTL). The lower row panels show the differences between two control runs shown for comparison. The contour intervals for the SST fields are 5° K and for the SST bias and difference are 0.3° K. Numbers above the figures in the middle and lower rows show the global bias and RMSE.

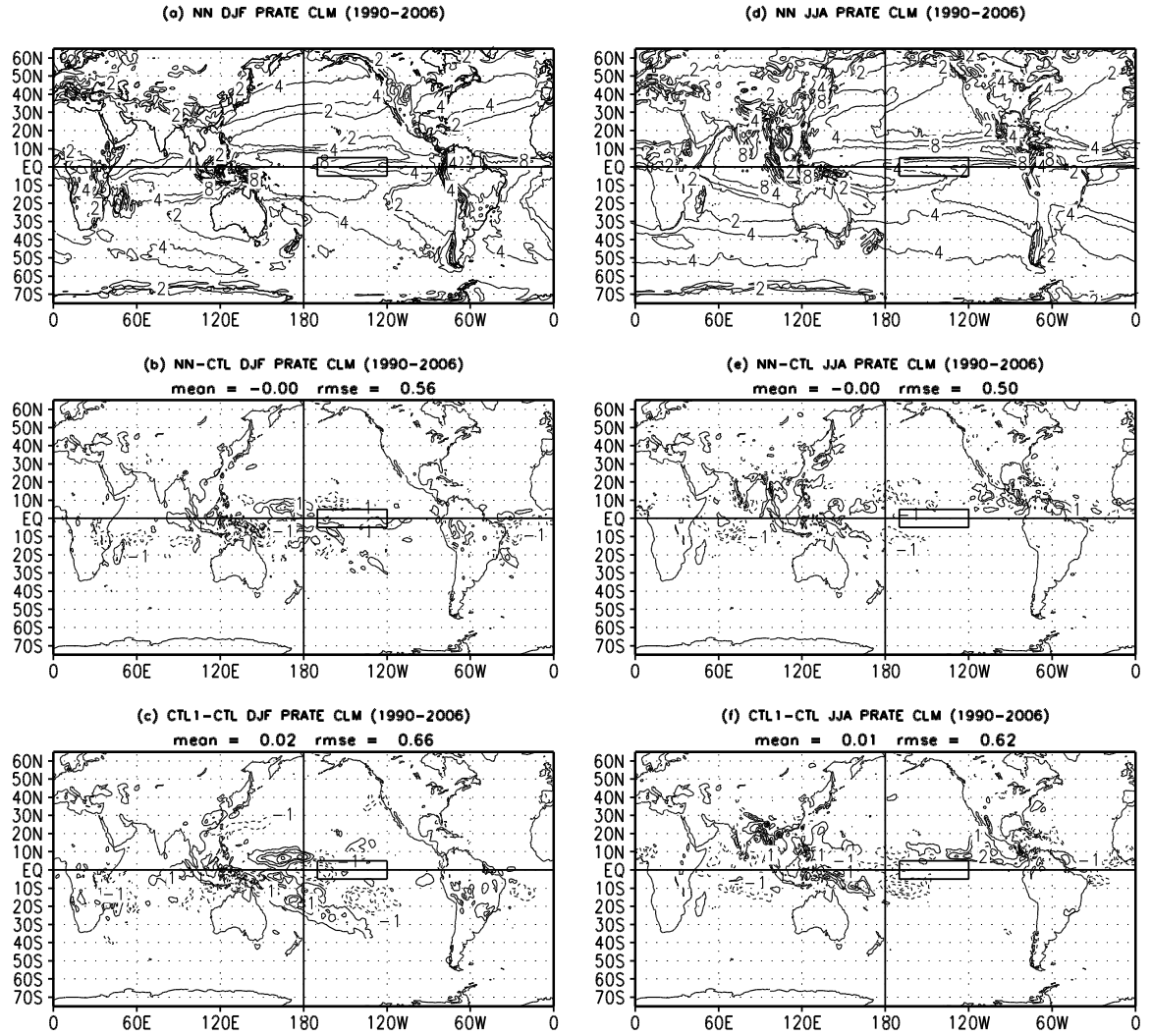


Fig. 14. The same as in Fig. 13 but for total precipitation (PRATE). The contour levels for the PRATE fields are 2, 4, 8, 16 and 32 mm/day. The contour intervals for the PRATE differences (the bottom panels) are 1 mm/day with 0 mm/day contour skipped for clarity.

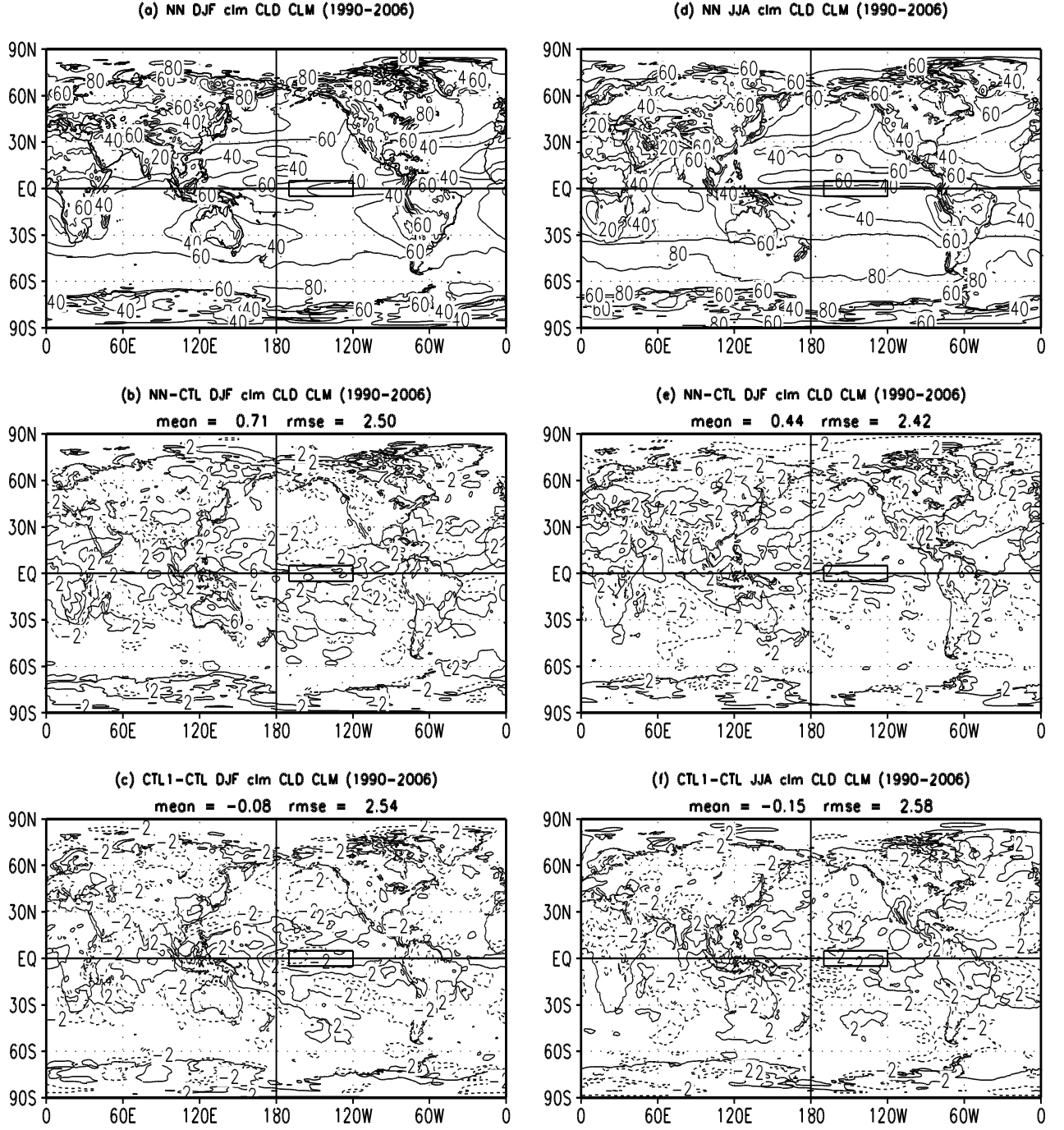


Fig. 15 The same as in Fig. 14 but for total clouds. The contour intervals for the cloud fields are 20% and for the differences – 4% with 0 % contour skipped for clarity.

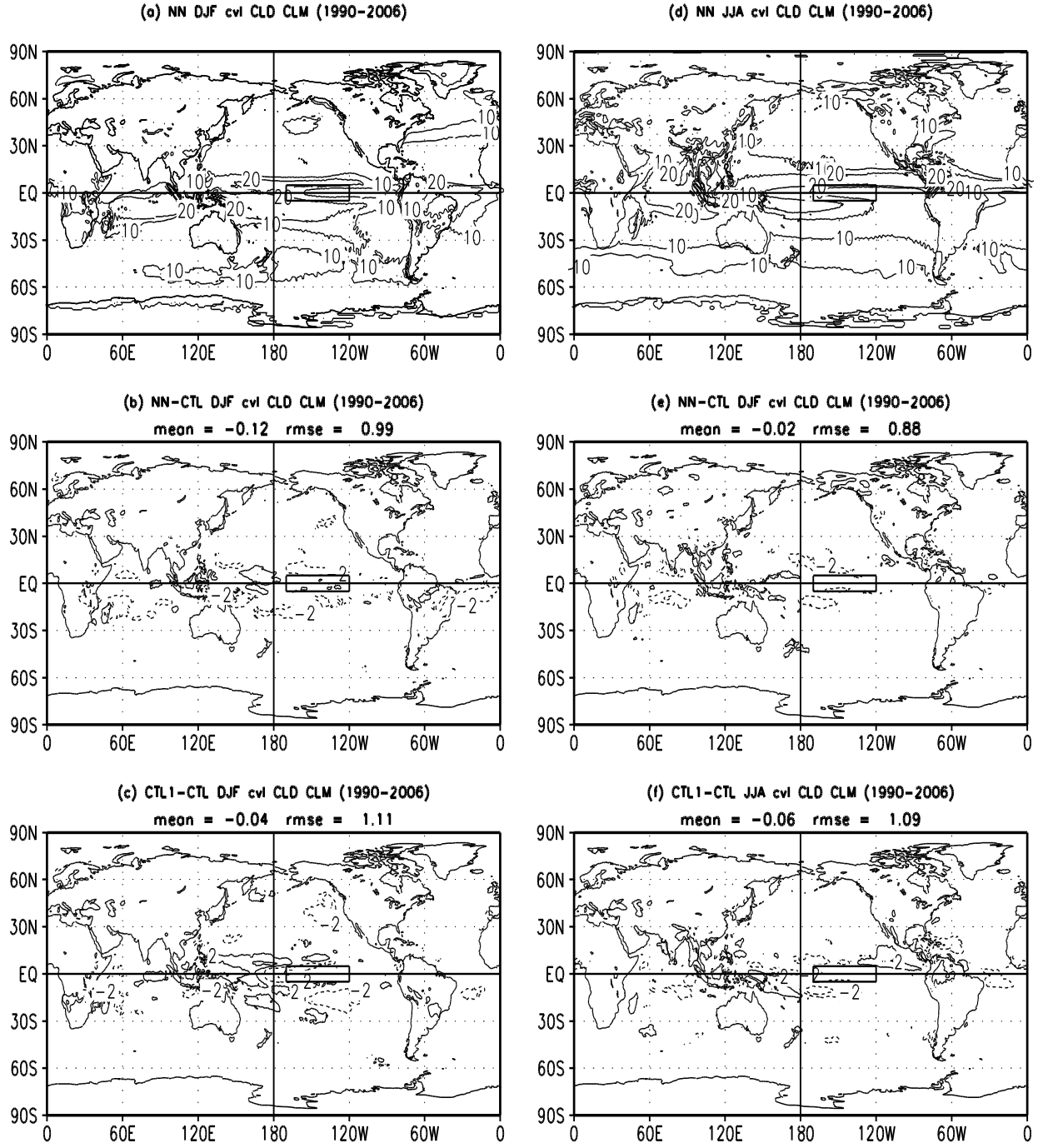


Fig. 16 The same as in Fig. 15 but for convective precipitation clouds. The contour intervals for the cloud fields are 10% and for the differences – 4%.

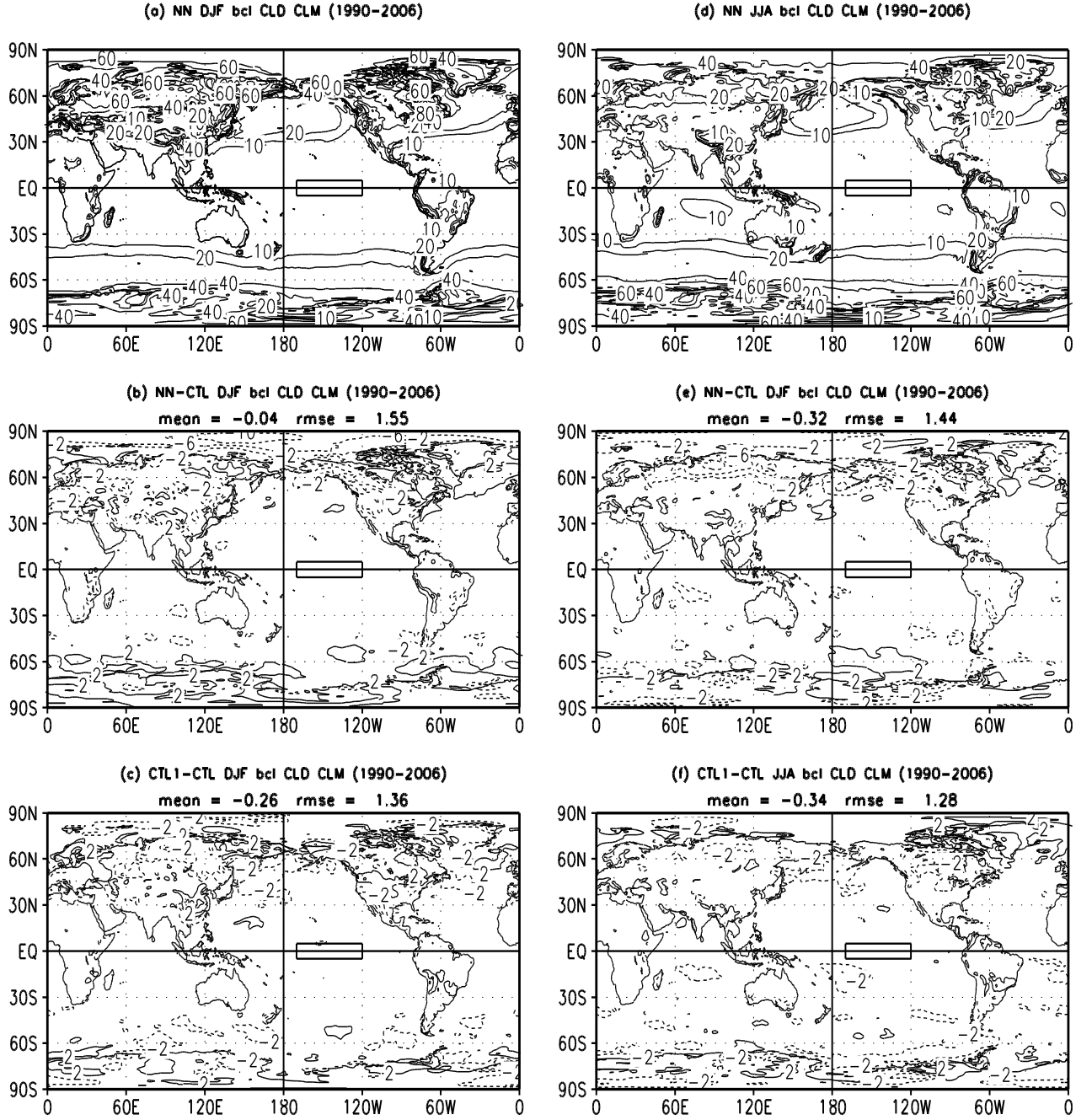


Fig. 17 The same as in Fig. 15 but for boundary layer clouds. The contour levels for the cloud fields are 10, 20, 40, 60, 80 and 100 % and for the differences - 4%.

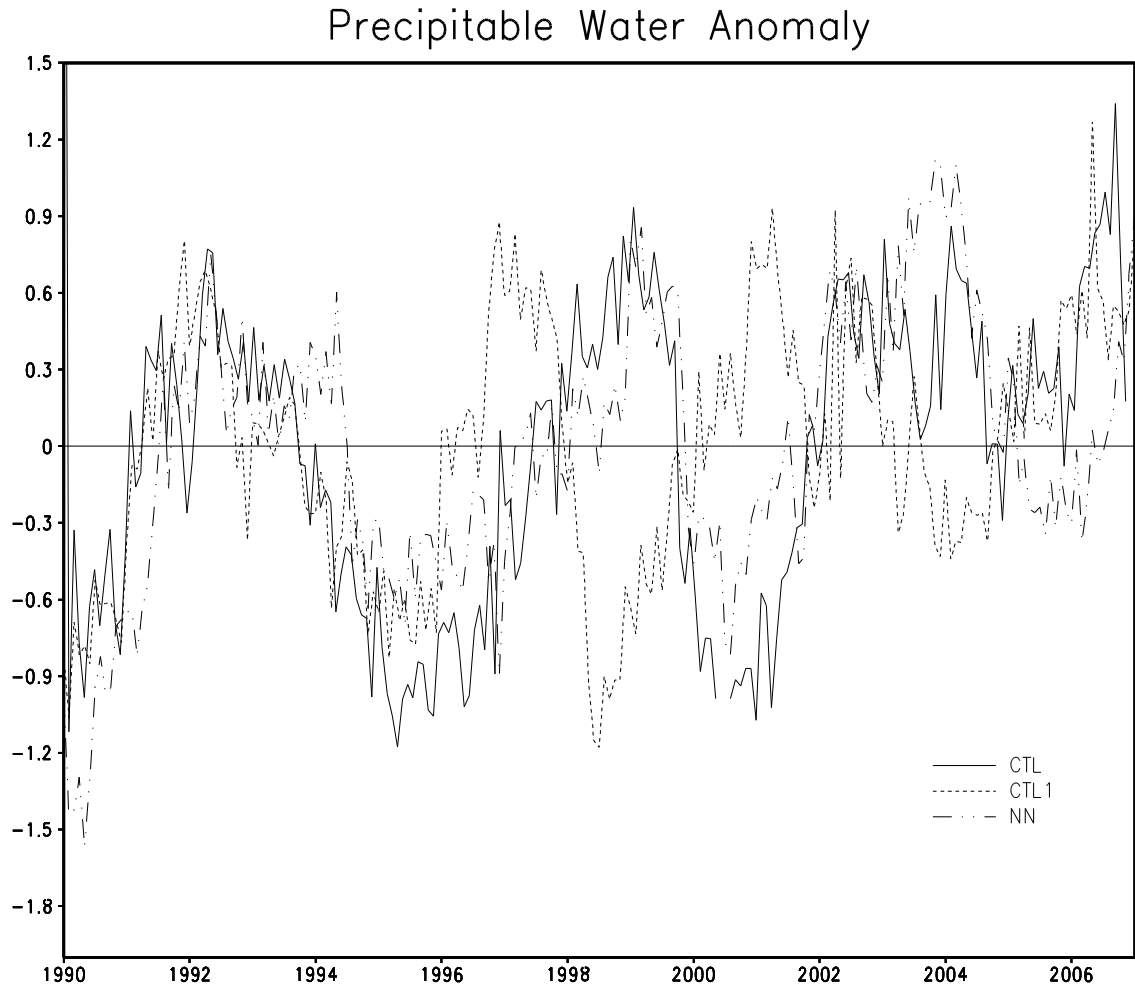


Fig. 18 The 17-year (1990-2006) time series of the total precipitable water anomaly (PWAT), with the seasonal cycle subtracted, for the full radiation NN run (dash-dotted line) and for two control runs, CTL (solid line) and CTL1 (dotted line). The mean PWAT anomaly is 25.48 mm/day for the control run (CTL), 25.62 for another control run (CTL1), and 25.64 mm/day for the NN run.

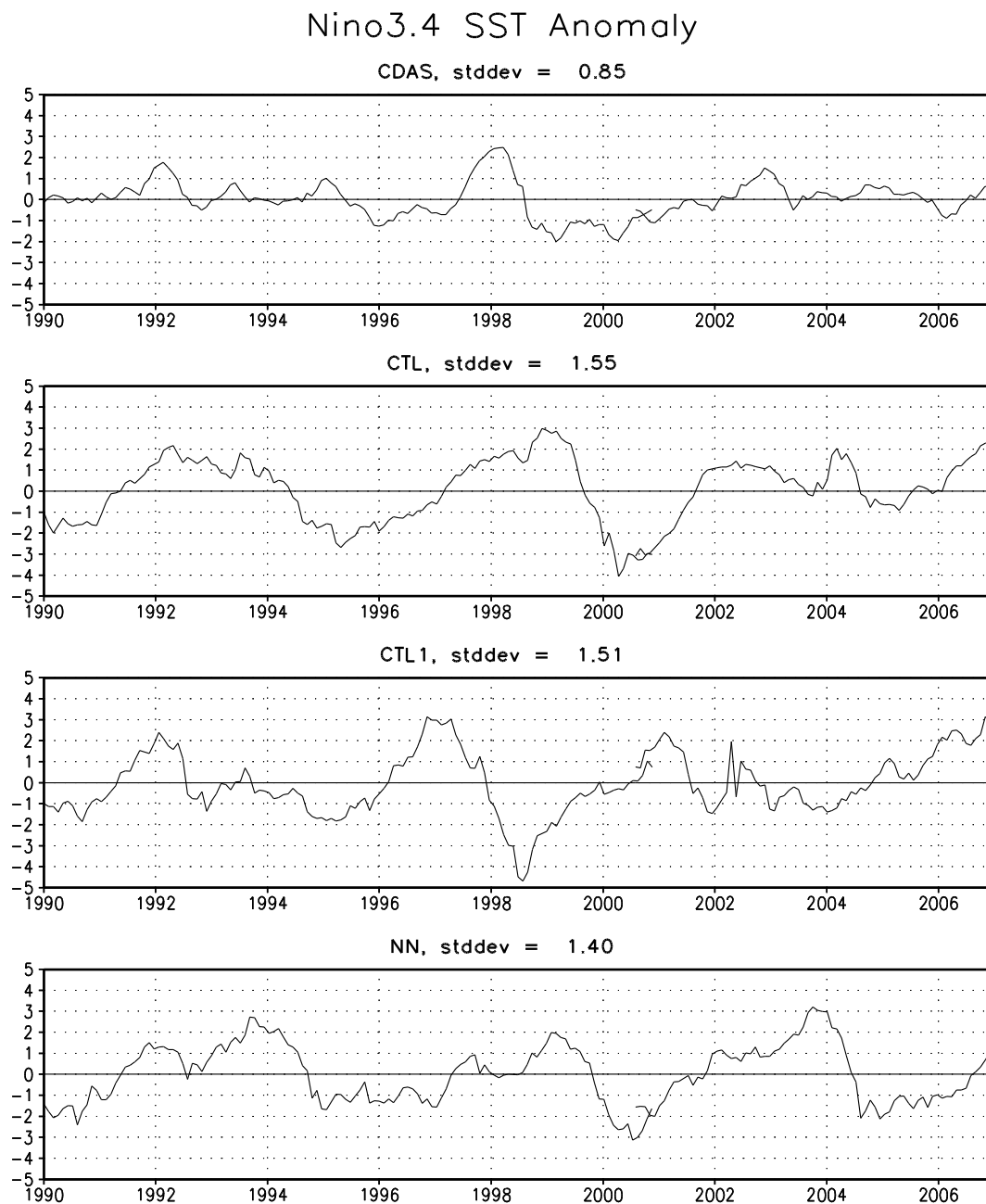


Fig. 19 The 17-year (1990-2006) time series for the Nino3.4 index for the reanalysis (CDAS) (the upper panel), and for the parallel full radiation NN (the bottom panel) and two control runs (the middle panels) described. The Nino3.4 index is calculated over the area in the Pacific Ocean shown by a rectangle in Figs. 13 – 17.

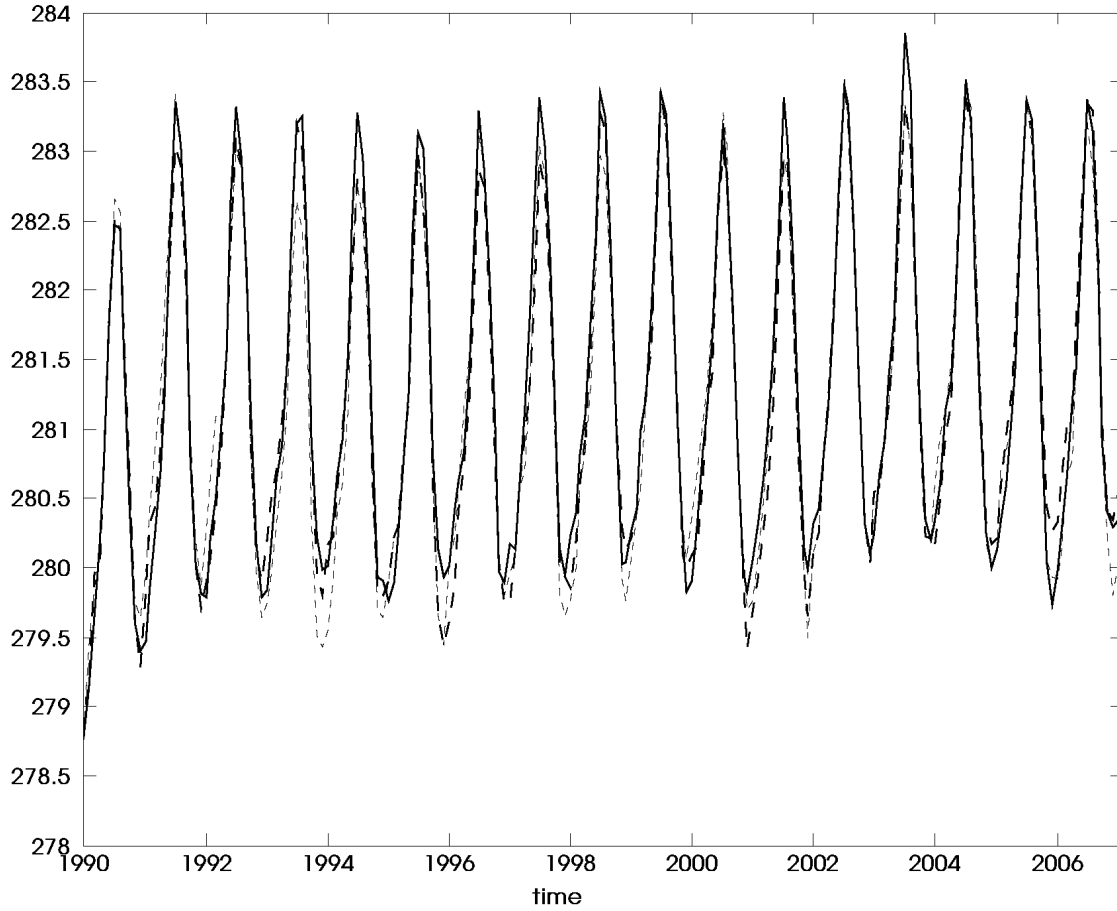


Fig. 20 The 17-year (1990-2006) time series for global mean temperature at 850 hPa (in K) for the parallel full radiation NN (solid line) and the old control (large-dashed line) and new control (short-dashed line) runs.

3.4.2.2 Seasonal predictions

We performed similar validation for seasonal predictions for 1990. Basically, the results are similar to those presented above. Fig. 21 shows biases or differences between the NN and control runs (NN-CTL) and differences between two control runs (CTL1-CTL) for seasonal predictions of SST, total clouds (clm CLD), total precipitation (PRATE), and convective clouds (cvl CLD).

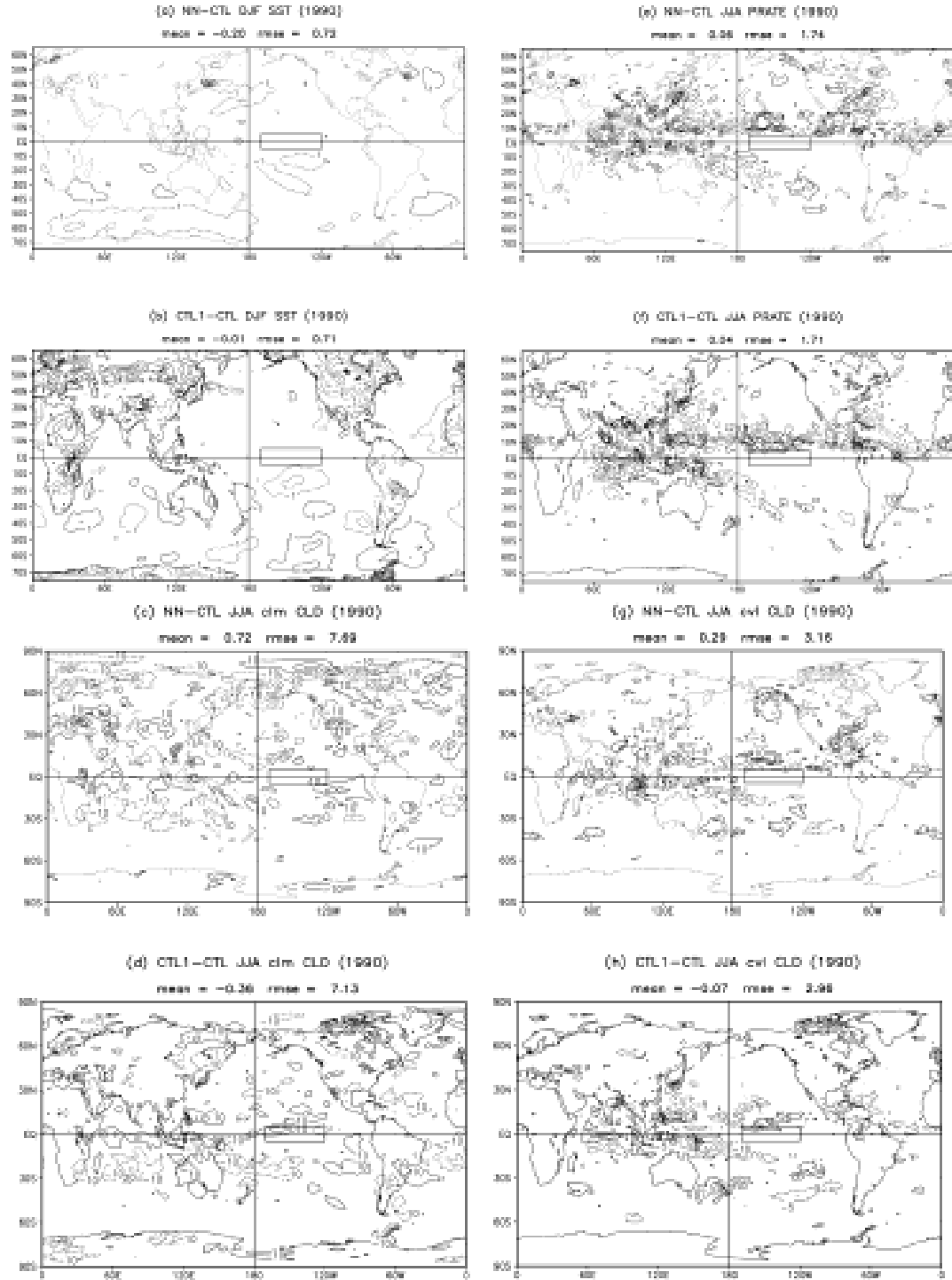


Fig. 21 Biases or differences between the NN and control runs (NN-CTL) and differences between two control runs (CTL1-CTL) for seasonal predictions of 1990 for: winter (DJF) SST - (a) and (b) panels, summer (JJA) total clouds (clm CLD) -

(c) and (d) panels, total precipitation (PRATE) – (e) and (f) panels, and convective clouds (cvl CLD) – (g) and (h) panels. The contour intervals for the SST fields are 1° K, for PRATE – 2 mm/day, for total clouds – 10%, and for convective precipitation clouds – 5%.

All the patterns for the control and NN runs (not shown) are quite close to each other. The differences between seasonal predictions produced by the parallel full radiation NN and control runs are slightly larger than the differences for climate simulations shown above in section 3.4.2.1. It is partly due to a smaller sample used for seasonal predictions. However, the differences/biases are still comparable with the observation errors and uncertainties of reanalysis. The differences do not increase significantly from season one to season four. For the seasonal predictions, biases or differences between the NN and control runs (NN-CTL) are close by magnitude and do not exceed the differences between the two control runs (CTL1-CTL) or model's internal variability. The time and global mean biases and RMSEs (shown in panel titles) are also quite small for NN-CTL and comparable with those of CTL1-CTL.

The examples of seasonal predictions show overall reasonable results. We realize that for practical implementation seasonal predictions should be produced in an ensemble mode (typically, including several tens of ensemble members), to reduce the impacts of internal variability and to estimate forecast uncertainty. However, this kind of

testing is supposed to be done by the NCEP/EMC implementation group and goes beyond the scope of this study.

3.5 Discussion

The study has shown the practical possibility of using highly efficient NN emulations for the full model radiation block for decadal and seasonal climate simulations. A very high accuracy and increased speed of NN emulations for the NCAR CAM and NCEP CFS full radiation (LWR and SWR) has been achieved. The systematic errors introduced by NN emulations of full model radiation are negligible and do not accumulate during the decadal model simulation. The random errors of NN emulations are also small. Almost identical results have been obtained for the parallel climate and seasonal simulations. These results show the potential of developing efficient NN emulations for model physics components and the entire model physics.

Comparison of the results obtained for NCAR CAM and NCEP CFS show that our NN emulation approach works for the high resolution (T126L64) NCEP CFS as well as for the lower resolution (T42L26) NCAR CAM. The NN emulation approach has already been applied to both LWR and SWR parameterizations and tested in different models with different dynamical cores and with different resolutions. It is significantly less dependent (in terms of both the accuracy and speed-up of calculations) on the increase of vertical resolution than the NN approach introduced by Chevallier et al. (1998) for developing a NN based LWR parameterization

NeuroFlux. At vertical resolution of 60 layers and more, both accuracy and rapidity of NeuroFlux cannot be achieved simultaneously (Morcrette et al. 2008). As we demonstrated in this study, our NN emulation approach can achieve simultaneously both the desired high accuracy and significant speed-up at vertical resolution of 60 layers and more.

Applying the NN emulation approach, which allows us to achieve such a significant speed-up with preservation of the accuracy and functional integrity of model physics, may create some challenges that can be resolved using the tremendous flexibility of statistical learning techniques and of the NN technique in particular. Because NN emulations are statistical approximations, there exists a small probability of larger approximation errors or outliers. The major reason for obtaining larger errors is high dimensionality n of the input space of the mapping (1), which reaches several hundreds for NCEP CFS and may reach thousands for future models with significantly higher vertical resolution. It is difficult to sample uniformly a domain in such a high dimensional space. Far corners of the domain may remain underrepresented in the training set. During the NN run, if input vectors belonging to these underrepresented far corners of the domain are encountered, they may cause larger errors in the NN outputs. These larger errors can be successfully controlled using a compound parameterization technique with a quality control procedure for removing larger errors and/or using the NN ensemble approach with NN emulations which will be discussed in Chapters 6 and 7 of this dissertation respectively.

Because model vertical resolution determines the NN emulation architecture, i.e., the number of inputs and outputs, every time the vertical resolution of the model is changed (which is usually done quite rarely), the NN emulation needs to be retrained. It is noteworthy that NN retraining can be done routinely and takes a limited time and effort once the practical framework for a specific model is developed.

In some applications of the developed NN emulation (in a data assimilation system or for an error and sensitivity analysis) not only NN emulation but also its first derivatives (NN Jacobian) are used. High accuracy of NN emulation does not automatically guarantee the accuracy of the NN Jacobian. An approach that allows to calculate accurately the NN Jacobian was developed by Krasnopolsky (2007b).

Chapter 4: Neural Network Emulation of Full Model Radiation: Short-to-Medium Range Eight Days Forecasts with NCEP GFS.

In this chapter we will explore the applicability of the NN emulation approach to short- to-medium range forecasts.

4.1 Brief Description of the High Resolution Version of NCEP GFS

In the in new set up the NCEP GFS configuration is run at the T574L64 resolution, uses prescribed time variable CO₂, prescribed time variable aerosols and a new set of radiation parameterizations. The RRTMG-LW is used in the GFS and CFS model as a LWR parameterization. It is based on the AER's RRTM-LW v2.3 (Mlawer et al., 1997, Iacono et al., 2000). The SWR parameterization used in GFS and CFS is a modified version of AER's RRTMG-SW (v2.3) (Clough et al., 2005). Although both RRTMG-LW and RRTMG-SW are built with fast computation schemes designed for GCM applications, they still represent the most time-consuming part of model physics in the NCEP GFS model. The percentage of the total model computation time used by model physics and by radiation (LWR and SWR) vary largely depending on the model horizontal and vertical resolution, the time step, the frequency of radiative calculations, and the computing environment (e.g. the number of processors and threads). In our set up the portion of the radiation computation time is about 15-18% of the total model computation time.

4.2 NN Emulations for Full Model Radiation

In the previous chapter when developing NN emulations for the radiation parameterizations of NCAR CAM, we followed a straightforward approach in selecting the emulating NN architecture. Inputs and outputs of the emulating NN have been selected to be identical to the inputs and outputs of the radiation parameterization to be emulated. For CAM, which has 26 vertical layers, the LWR emulating NN has 220 inputs and 33 outputs and the SWR emulating NN has 451 inputs and 33 outputs. If we followed this procedure for the NCEP model, which has 64 vertical layers, the LWR emulating NN would have 585 inputs (7 profiles of 64 components each + 2 profiles of 65 components each + 6 scalar variables), and the SWR emulating NN would have 3,277 inputs (49 profiles of 64 components each + 2 profiles of 65 components each + 10 scalar variables). However, even for this straightforward approach, the number of NN inputs is less than the number of input profiles multiplied by the number of vertical layers plus the number of relevant single level characteristics. Many input variables have zero or constant values for the upper (e.g., water vapor) or lower (e.g., ozone) vertical layers, and for some gases the entire volume mixing ratio profile is a constant (obtained from climatological data). To improve the accuracy of the approximation, these constant inputs should not be used for NN training. Constant inputs (zero or nonzero) do not contribute to the functional input/output relationship and should not be used as inputs and/or outputs for NN emulations. Moreover, if they were used, they would introduce an additional noise

(an approximation error). Thus, 92 such constant inputs have been removed (see Table 4), which, in addition, significantly reduced the emulating NN dimensionalities. In terms of reducing the number of NN inputs, some input profiles contain a lot of redundancy that, if properly identified, can be used to reduce the input dimensionality (Krasnopolsky et al. 2009). Some profiles depend on the vertical coordinate very smoothly. Autocorrelation functions (ACF) for vertical profiles of some model variables are shown in Fig. 22. ACF of a profile shows the correlation between adjacent components of the profile (between values of the corresponding variables at the adjacent model levels). Slowly decreasing ACF (like those for pressure and temperature shown in black and red in Fig. 22) shows that the adjacent components of the profile are highly correlated and that redundant information is introduced if all of them are used as inputs for the emulating NN. For such profiles a sampling can be applied to reduce the redundancy and dimensionality of the NN input vector. For these profiles every other or even every third level can be selected as NN input. For some other profiles (e.g., cloud fraction and cloud liquid path shown in pink and brown in Fig. 22) the corresponding ACFs decrease very quickly, which means that the redundancy for these variables is insignificant and the sampling should not be applied. In the case of LWR and SWR NN emulation, for the pressure and temperature profiles we applied samplings shown in Table 4. This procedure allowed us to eliminate 50 redundant NN inputs without any significant reduction in the approximation accuracy (see Table 5).

Table 4. Inputs and outputs of LWR and SWR NNs developed for GFS

LWR NN Inputs			SWR NN Inputs	
NN input	NN input ##	Levels	NN input ##	Levels
$\cos\left(\frac{month \cdot \pi}{6}\right)$	1	-	1	-
$\sin\left(\frac{month \cdot \pi}{6}\right)$	2	-	2	-
$\cos(Lon)$	3	-	3	-
$\sin(Lon)$	4	-	4	-
Lat	5	-	5	-
Interface pressure	6:28	1,2:2:42,43	6:28	1,2:2:42,43
Interface temperature	29:64	1,2,3:2:61,62:65	29:64	1,2,3:2:61,62:65
Layer H ₂ O mixing ratio	65:104	1:40	65:104	1:40
Layer O ₃ mixing ratio	105:138	31:64	105:138	31:64
Layer total cloud fraction	139:186	1:48	139:185	1:47
Surface emissivity	187	-	-	-
Cos of zenith angle	-	-	186	-
Surface albedo	-	-	187:190	1:4
Total Inputs	187		190	
LWR NN Outputs			SWR NN Outputs	
NN output	NN output ##	Levels	NN output ##	Levels
Layer Heating Rates	1:64	1:64	1:64	1:64
Total sky upward flux at toa	65	-	65	-
Clear sky upward flux at toa	66	-	66	-
Total sky downward flux at toa			67	-
Total sky upward flux at sfc	67	-	68	-
Clear sky upward flux at sfc	68	-	69	-
Total sky downward flux at sfc	69	-	70	-
Clear sky downward flux at sfc	70	-	71	-
Total sky downward uv-b flux at sfc	-	-	72	-
Clear sky downward uv-b flux at sfc	-	-	73	-
Total Outputs	70		73	

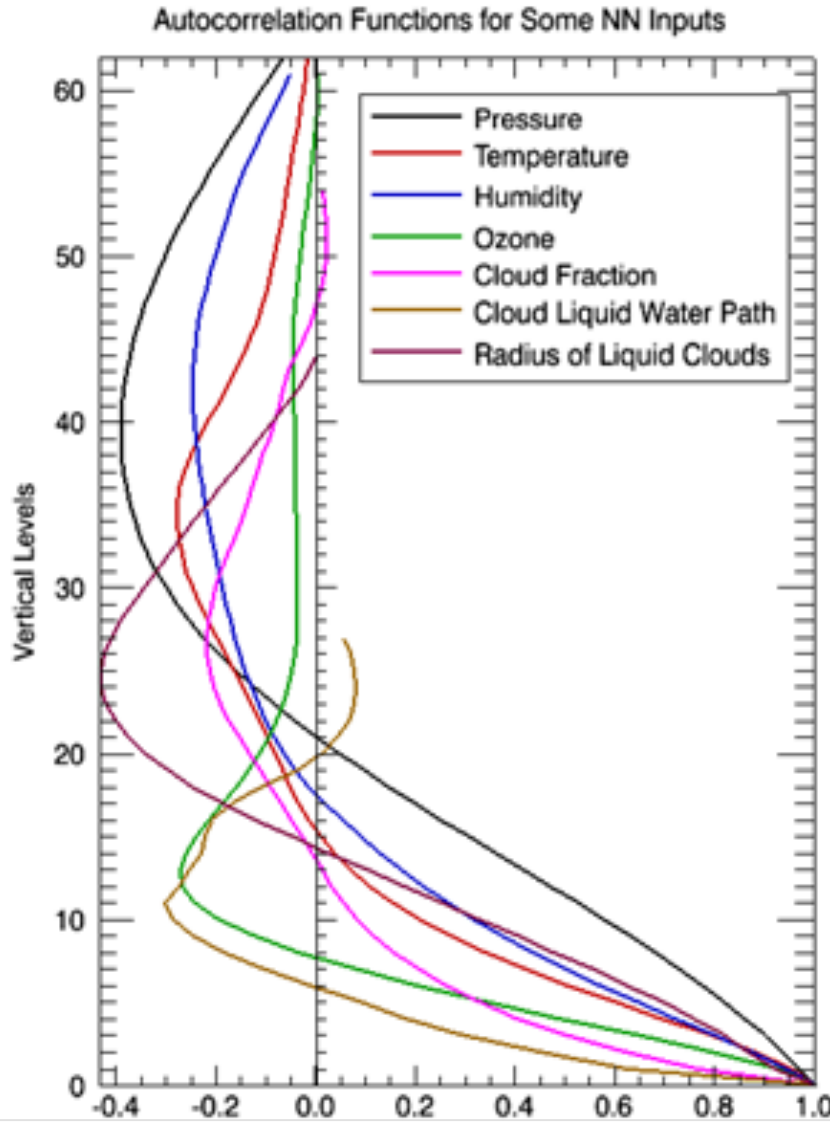


Fig. 22 Autocorrelation function for several NN input profiles. The horizontal axis shows the correlation and the vertical axis – the lag in vertical levels. Curves have different length for different input parameter profiles because the profiles have different number of nonzero components.

Also, for SWR, 2688 inputs describing the optical depth, single scattering albedo, and asymmetry parameters of 14 aerosol species were substituted by five inputs: $\cos(\tau)$,

$\sin(\tau)$, $\cos(\text{lon})$, $\sin(\text{lon})$, and lat , where lon is the longitude, lat is the latitude, and $\tau = \frac{2 \cdot \pi}{T} \cdot q$, where q is the month of the year, and $T = 12$. Such a substitution is possible because in NCEP CFS and GFS aerosol model aerosols are calculated using the specific humidity profiles and 3-D lookup tables composed of climatological monthly data, different for different months of the year. It means that in terms of functional input/output dependences, the aerosol characteristics are the functions of lat , lon , τ , and the specific humidity only. Since the profile of the specific humidity has been already included in NN SWR inputs, only the five aforementioned additional variables have to be included to allow NN to completely emulate the contribution of aerosols into SWR. Thus, *the SWR emulating NN emulates both the aerosol model and SWR*.

In this work we generalized this approach to reduce even more the size of the SWR and LWR emulating NNs. Both SWR and LWR use, in addition to the cloud fraction profile, four other cloud characteristic profiles: layer cloud liquid water path (LCLWP), layer mean effective radius for liquid droplet (LMERLD), layer cloud ice water path (LCIWP), and mean effective radius for ice cloud (MERIC). These four profiles are calculated in microphysics block using models (equations) that use the specific humidity and atmospheric temperature profiles (Moorthi et al. 2001). Since the profiles of the specific humidity and atmospheric temperature have been already included as inputs in NNs emulating SWR and LWR, and the four aforementioned profiles are correlated with the cloud fraction profile, the emulating NNs are capable of emulating the part of microphysics that calculate these four additional profiles also. Thus, we excluded these four profiles from SWR and LWR NN inputs (totally 256

inputs have been eliminated). As a result, the developed LWR emulating NN emulates actually, in addition to LWR parameterization, the cloud microphysics calculations of LCLWP, LMERLD, LCIWP, and MERIC. The developed SWR NN emulation emulates actually SWR parameterization, cloud microphysics, and aerosol model.

Table 5. Statistics estimating the accuracy of HRs (in K/day) calculations and the computational performance for NCEP GFS (T574L64) LWR and SWR using NN emulation vs. the original parameterization. For comparison, NCEP CFS (T126L64) LWR and SWR statistics are also shown. Total statistics show the mean error or bias, RMSE, PRMSE, and σ_{PRMSE} for the entire 3-D HR fields. Layer (for the top and bottom layers) statistics show the mean error and RMSE for one horizontal layer (the top or bottom layer). The NN complexity N_C and average speedup η are shown.

	Statistics	GFS RRTMG-LW	CFS RRTMG-LW	GFS RRTMG-SW	CFS RRTMG-SW
Total Error Statistics	Bias, eq. (A1)	$8. \cdot 10^{-3}$	$2. \cdot 10^{-3}$	$-7. \cdot 10^{-3}$	$5. \cdot 10^{-3}$
	RMSE, eq. (A1)	0.52	0.49	0.26	0.20
	PRMSE, eq. (A1)	0.38	0.39	0.18	0.16
	σ_{PRMSE}, eq. (1)	0.36	0.31	0.19	0.12
Bottom Layer Error Statistics	Bias, eq. (1)	$2. \cdot 10^{-2}$	$-1. \cdot 10^{-2}$	$-3. \cdot 10^{-2}$	$9. \cdot 10^{-3}$
	RMSE, eq. (1)	0.55	0.64	0.20	0.22
Top Layer Error Statistics	Bias, eq. (1)	$5. \cdot 10^{-2}$	$-9 \cdot 10^{-3}$	$-1. \cdot 10^{-3}$	$1. \cdot 10^{-2}$
	RMSE, eq. (1)	0.13	0.18	0.13	0.21
NN Complexity	N_C See eq. (1)	25,870	33,294	26,473	45,173

Speedup, η	<i>Times</i>	20	16	100	80
-----------------------------------	---------------------	----	----	-----	----

The GFS model with the original LWR and SWR parameterizations have been used to simulate data for NN training and test. The data set was composed of 24 ten day forecasts started each first and fifteenth day of the month during one year (2010). All inputs and outputs of the original LWR and SWR parameterization have been saved over the globe eight times per day (every three hours) during each day of each ten day forecast. Thus 1920 global data sets have been generated for NCEP GFS. About 300 data records have been randomly selected from each global data set. The collected set of about 600,000 input/output radiation vectors was divided into three independent parts, each containing about 200,000 input/output vector combinations (records). The first part was used for training, the second – for validation (control of overfitting, control of a NN architecture, etc.), and the third part – for testing the approximation quality only.

A number of NNs has been trained using the training set described above. The developed NN emulations use from 60 to 150 neurons in one hidden layer and have the same inputs and outputs presented in Table 4. Then bulk validation statistics for the accuracy of approximation and computational performance for the developed NNs emulations have been estimated on independent data set. LWR and SWR NNs with 100 hidden neurons have been selected for testing in GFS. The accuracy of the selected NN emulations has been estimated against the original GFS radiation parameterizations; the statistics are presented in Table 5. For these NN emulations, bias is negligible (about 10^{-3} K/day) and RMSE is limited (about 0.3 – 0.5 K/day).

Obtaining very small NN emulation biases is important to ensure non-accumulating errors in the course of model integrations using NN emulations. The NN emulations developed for GFS are smaller and faster than those developed previously for CFS (see Chapter 3 for description). However, as seen in Table 5, they are as accurate as CFS LWR and SWR NNs. The developed highly accurate NN emulations for LWR and SWR, in terms of code-by-code comparison at each model time step when LWR and SWR are calculated, are about 20 and 100 times faster than the original/control NCEP GFS LWR and SWR respectively.

4.3 Validation of Parallel Eight Day Model Forecasts

As the next step, the developed LWR and SWR NN emulations were validated in GFS model integrations. The LWR and SWR emulations with 100 neurons have been selected for an initial validation because they seem to be acceptable in terms of both their accuracy and minimal complexity (see the previous section). Several 8-day forecasts have been run using the GFS model (T574L64, 2011 version). Here we present results for three runs performed from August 1 to August 8, 2010. This period was selected for a validation run because there were several atmospheric events during this time:

1. Tropical storm Colin moved through the western Atlantic to Bermuda from August 2 to 8, 2010.
2. In the Eastern Pacific tropical storm Estelle moved from the coast of Mexico to about 400 miles southwest of Baja California from Aug 6 to 10, 2010.
3. Tropical storm Domeng occurred in the Western Pacific to the north east of

the Philippines from August 3 to 5, 2010.

Three parallel runs have been performed:

1. Control run labeled as PRNNCTL (black) in the figures, which uses the original radiation codes;
2. Run labeled as PRNNFULLGFS (red) that uses LWR NN and SWR NN developed for GFS using GFS (T574L64) simulated data (see above); and
3. Run labeled as PRNNFULLCFS (green) that uses LWR NN and SWR NN developed for CFS (see Chapter 3) using CFS data simulated by an old (T126L64) version of atmospheric model (see below).

It is noteworthy that NCEP CFS used for development and validation of CFS NN radiation (see Chapter 3) incorporated: the NCEP GFS (model version from 2006) atmospheric model with 126 spectral components and 64 vertical levels (T126L64) coupled with the 40-level interactive MOM4 ocean model, the interactive Noah land model with four soil levels with improved treatment of snow and frozen soil, an interactive sea ice model with fractional ice cover and depth allowed, a sub-grid scale mountain blocking, and a seasonal climatological aerosol treatment. The CFS NN radiation has been developed based on data set accumulated during 17 years of continuous CFS run. The GFS NN radiation presented in this chapter has been developed based on 24 ten day forecasts produced by high resolution T574L64 (model version from 2011) GFS (uncoupled atmospheric part of CFS), which, in addition to much higher spectral resolution, incorporates many changes in physics and other model components as compared with the version of CFS, for which the

CFS NN radiation has been developed. Thus, the CFS NN radiation has not been trained and validated for a significantly different model environment of the current version of GFS. In one of the GFS runs presented below we use this old CFS NN radiation in the new T574L64 GFS. Comparing results of this run with the control run and with the GFS run using new GFS NN radiation developed for the current version of GFS allows us to evaluate robustness of the NN emulation approach with respect to the changes in the model.

Figures 23 to 31 present various statistics (anomaly correlations, biases, and RMS errors) routinely used for evaluation and comparison of GFS runs. The comparisons of anomaly correlations, biases, and RMS errors have been performed for instantaneous model prognostic and diagnostic fields produced at each day of the 8-day forecasts. The NN radiation and control runs are very close in terms of calculated statistics.

For example, Fig. 23 shows the anomaly correlation calculated at 850 mb and 500 mb (left and right column respectively) for the temperature field. The upper and lower rows show results for the northern and southern hemispheres correspondingly. The middle row shows results for the tropics. Fig. 24 shows the anomaly correlation calculated for the geopotential height field for the 500 mb level and Fig. 25 shows the anomaly correlation calculated for the surface pressure field for the northern hemisphere (upper row), tropics (medium row), and southern hemisphere (lower row) correspondingly.

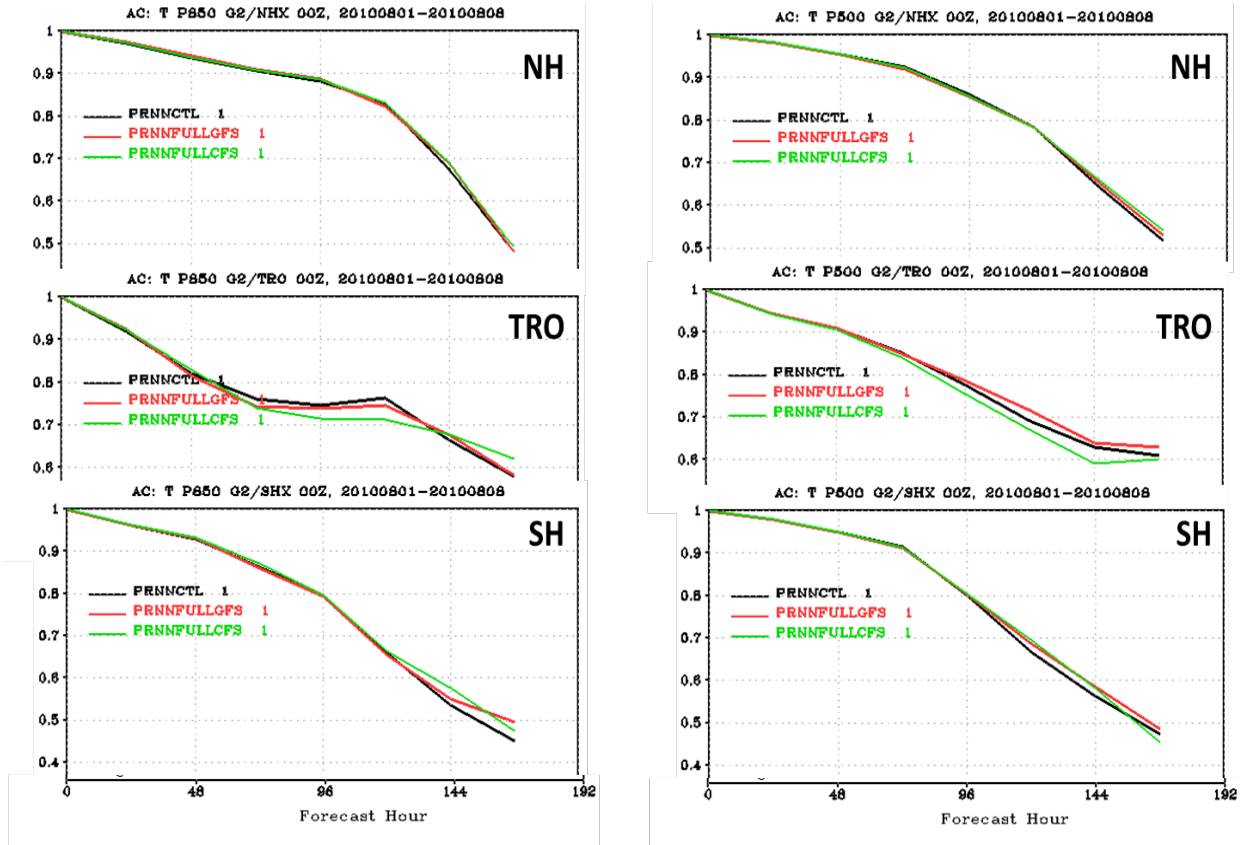


Fig. 23 Anomaly correlation at 850 mb (left column) and 500 mb (right column) for the northern hemisphere (upper row), tropics (medium row), and southern hemisphere (lower row) calculated for temperature fields. Black line – control run with the original LWR and SWR (PRNNCTL); green line – run with NN SWR and LWR developed for CFS (PRNNCFS); and red line – run with NN SWR and LWR developed for the current version of GFS (PRNNGFS).

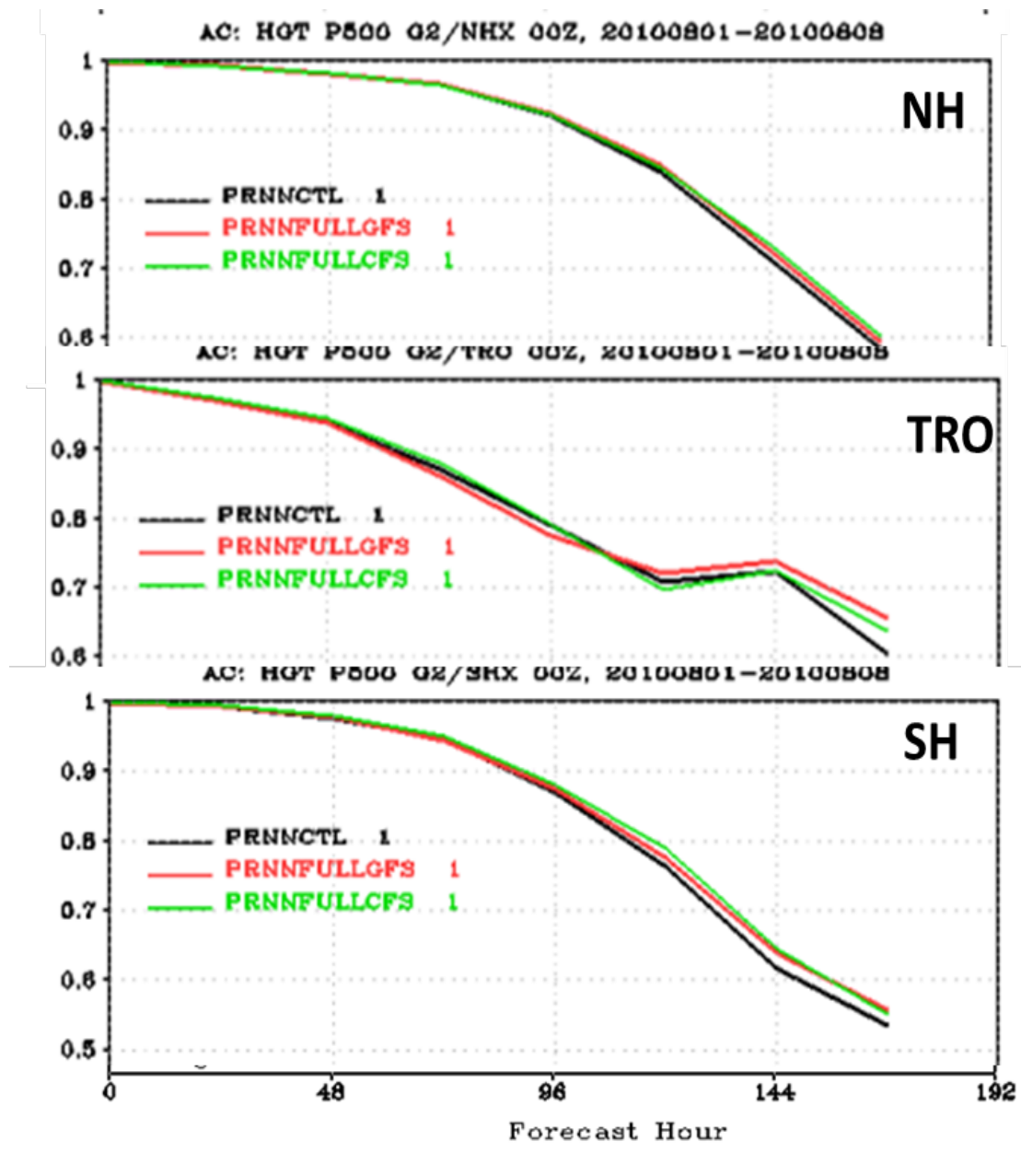


Fig. 24 Anomaly correlation at 500 mb for the northern hemisphere (upper row), tropics (medium row), and southern hemisphere (lower row) calculated for geopotential height fields. Black line – control run with the original LWR and SWR (PRNNCTL); green line – run with NN SWR and LWR developed for CFS

(PRNNCFS); and red line – run with NN SWR and LWR developed for the current version of GFS (PRNNGFS).

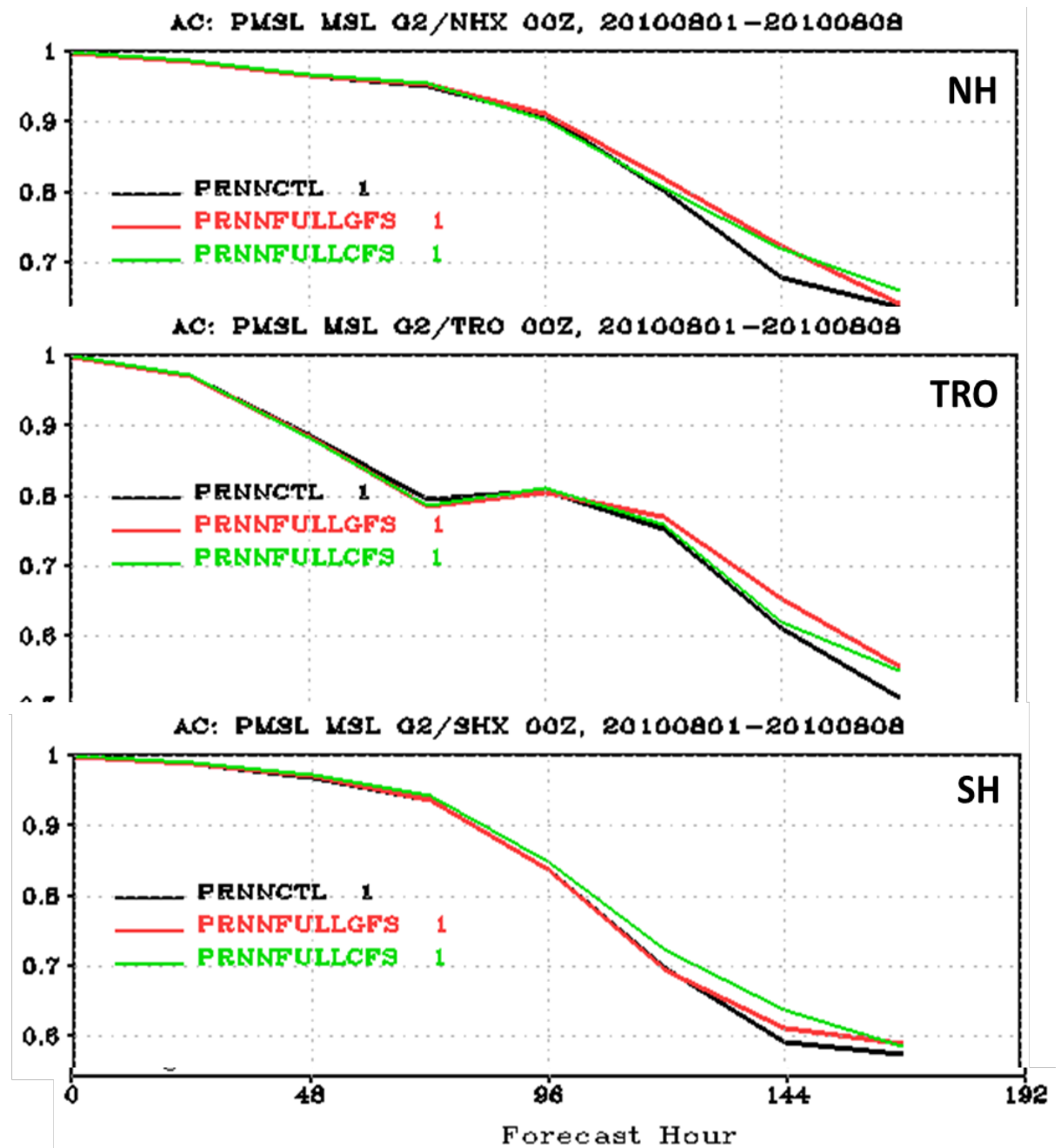


Fig. 25 Anomaly correlation calculated for the surface layer pressure fields. See caption to Fig. 24.

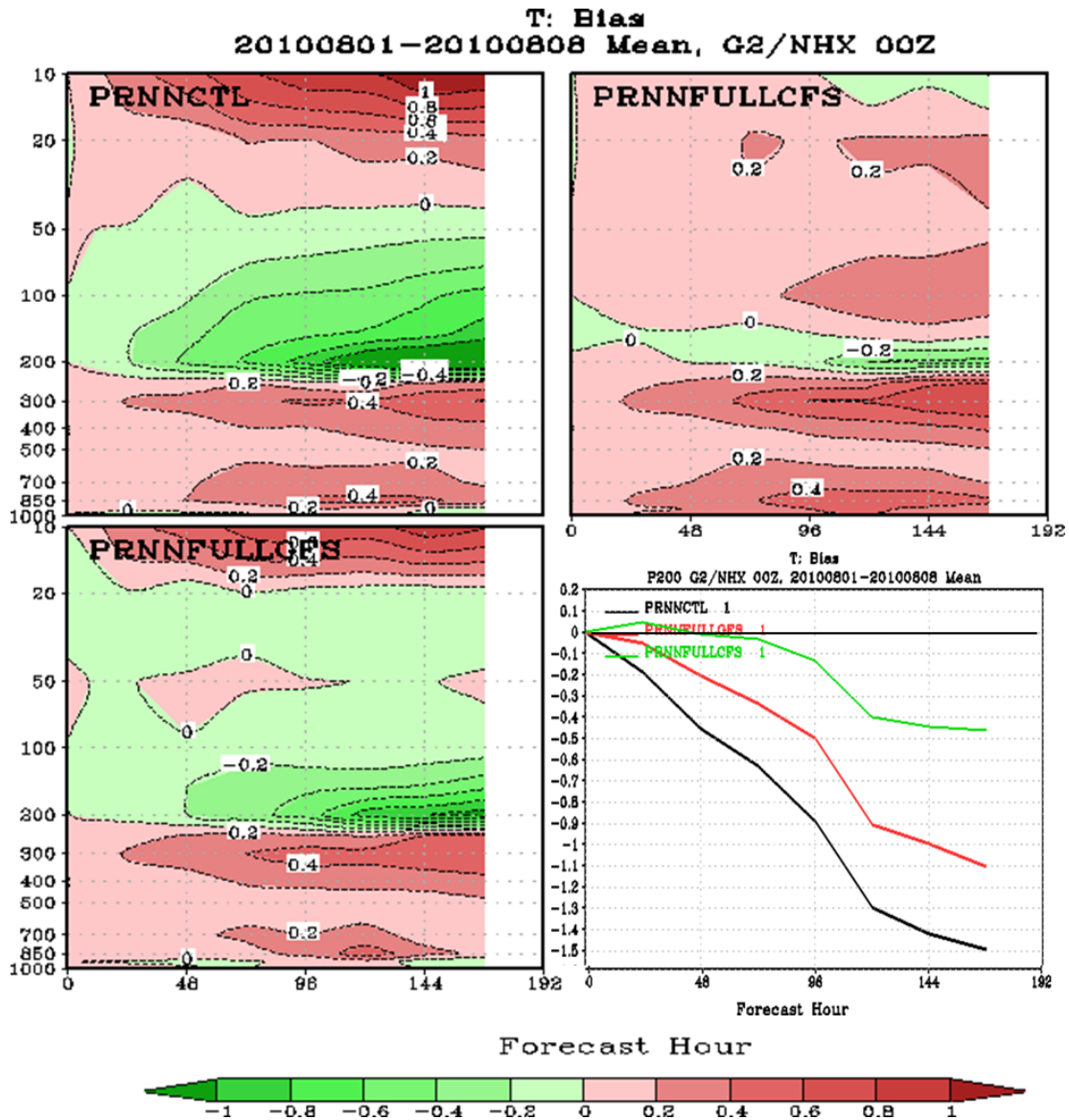


Fig.26 Temperature bias calculated for the northern hemisphere as a function of the forecast time (horizontal axis) and height in mb (vertical axis) for the control run, PRNNCTL (upper left), for PRNNCFSS (upper right), and for PRNNGFS (lower left). Lower right shows the bias at 200 mb level: black line – control run (PRNNCTL); green line – PRNNCFSS run; and red line – PRNNGFS run.

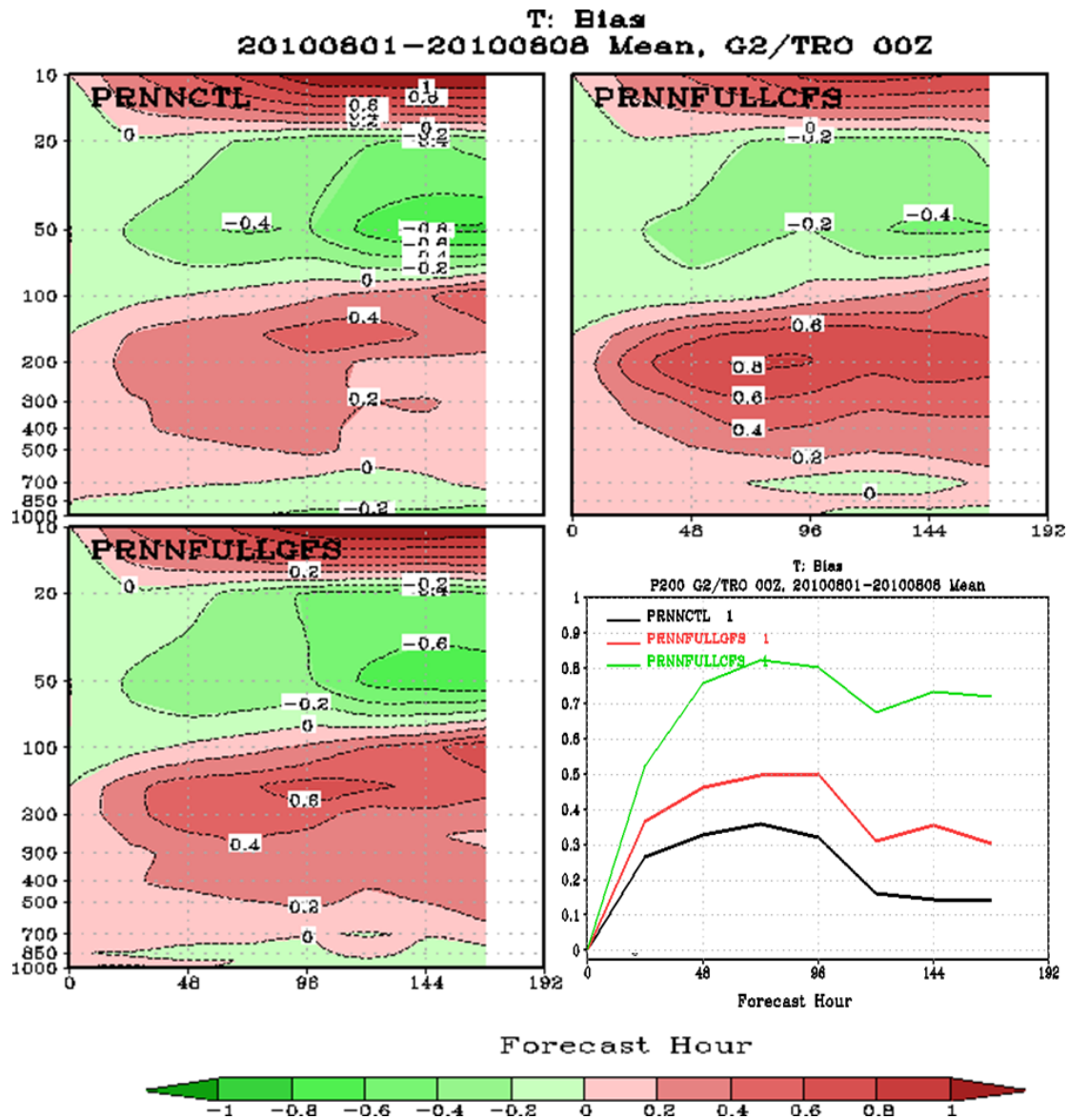


Fig.27 The same as in Fig. 26 calculated for the tropics.

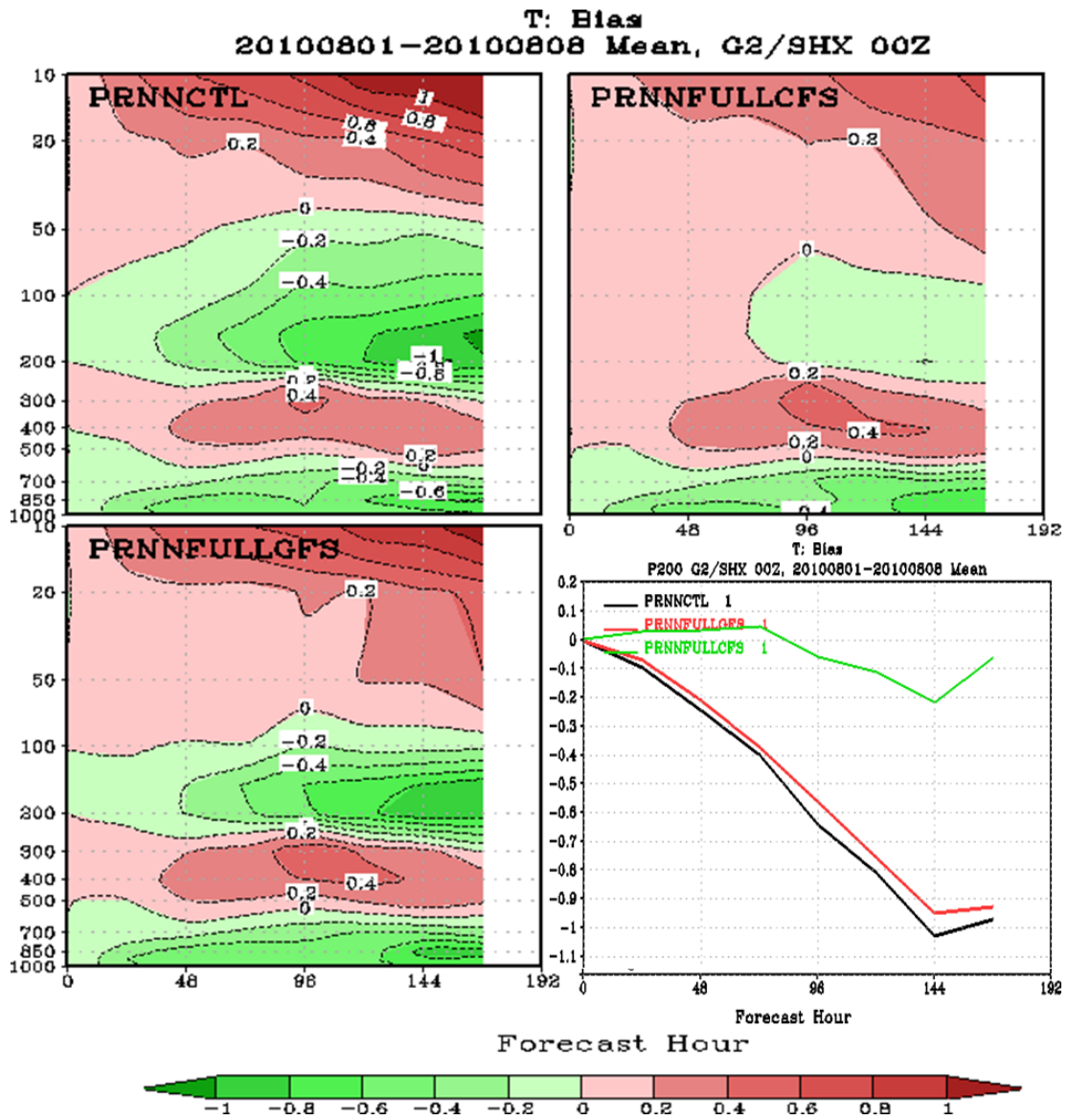


Fig.28 The same as in Fig. 26 calculated for the southern hemisphere.

The differences between NN runs and control run increase from day one to day eight but are remaining small. As could be expected, the *PRNNGFS* run (red) closer follows the control run (black) than the *PRNNCFS* run (green).

The CFS LWR and SWR NNs have been transplanted into new significantly different version of GFS. Relatively small difference between the *PRNNGFS* (red) and *PRNNCFS* (green) runs demonstrates the robustness of the NN emulation approach with respect to changes in the model environment.

Figs. 26 to 28 show the temperature bias calculated for the northern hemisphere (Fig. 26), for the tropics (Fig. 27), and the northern hemisphere (Fig. 28) as functions of the forecast time (horizontal axis) and height in *mb* (vertical axis). The upper left panel shows bias for the control run, the upper right – for *PRNNCFS* run, and the lower left – for *PRNNGFS* run. The lower right panel shows the cross-section of the three other panels at 200 mb.

Figs. 29 to 31 show the vector wind RMS errors and differences calculated for the northern hemisphere (Fig. 29), for the tropics (Fig 30), and the southern hemisphere (Fig. 31) as functions of the forecast time (horizontal axis) and height in *mb* (vertical axis). The upper left panel shows the RMS error for the control run, the upper right – the RMS differences for (*PRNNCFS* – control) run, and the lower left – for (*PRNNGFS* – control) run. The lower right panel shows the cross-section of the RMS errors for the three runs at 200 mb.

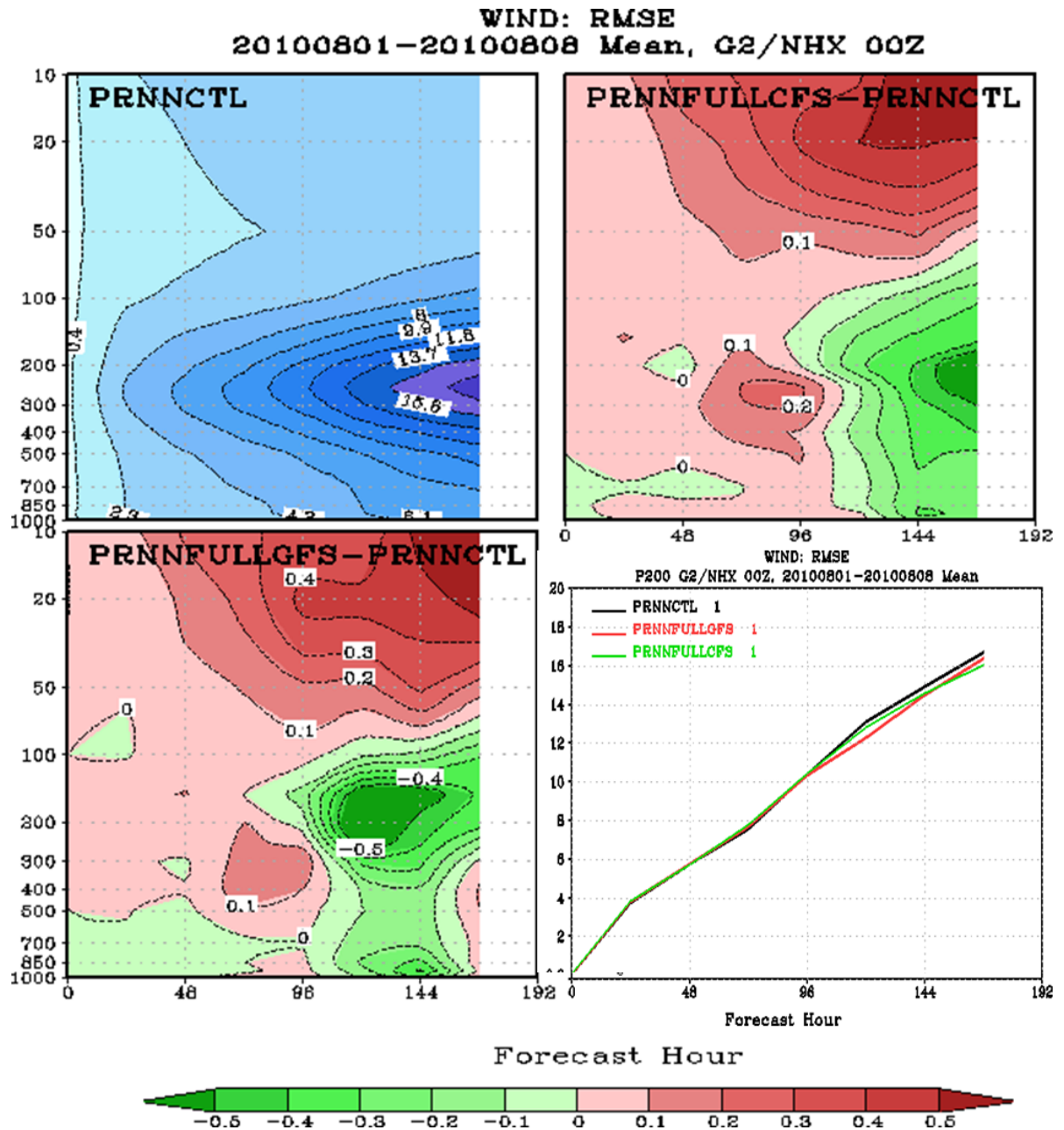


Fig.29 Vector wind RMSE calculated for the northern hemisphere as a function of the forecast time (horizontal axis) and height in mb (vertical axis) for the control run, PRNNCTL (upper left), for (PRNNCFS – PRNNCTL) (upper right), and for (PRNNGFS – PRNNCTL) (lower left). Lower right shows the RMSE at 200 mb level: black line – control run (PRNNCTL); green line – PRNNCFS run; and red line – PRNNGFS run.

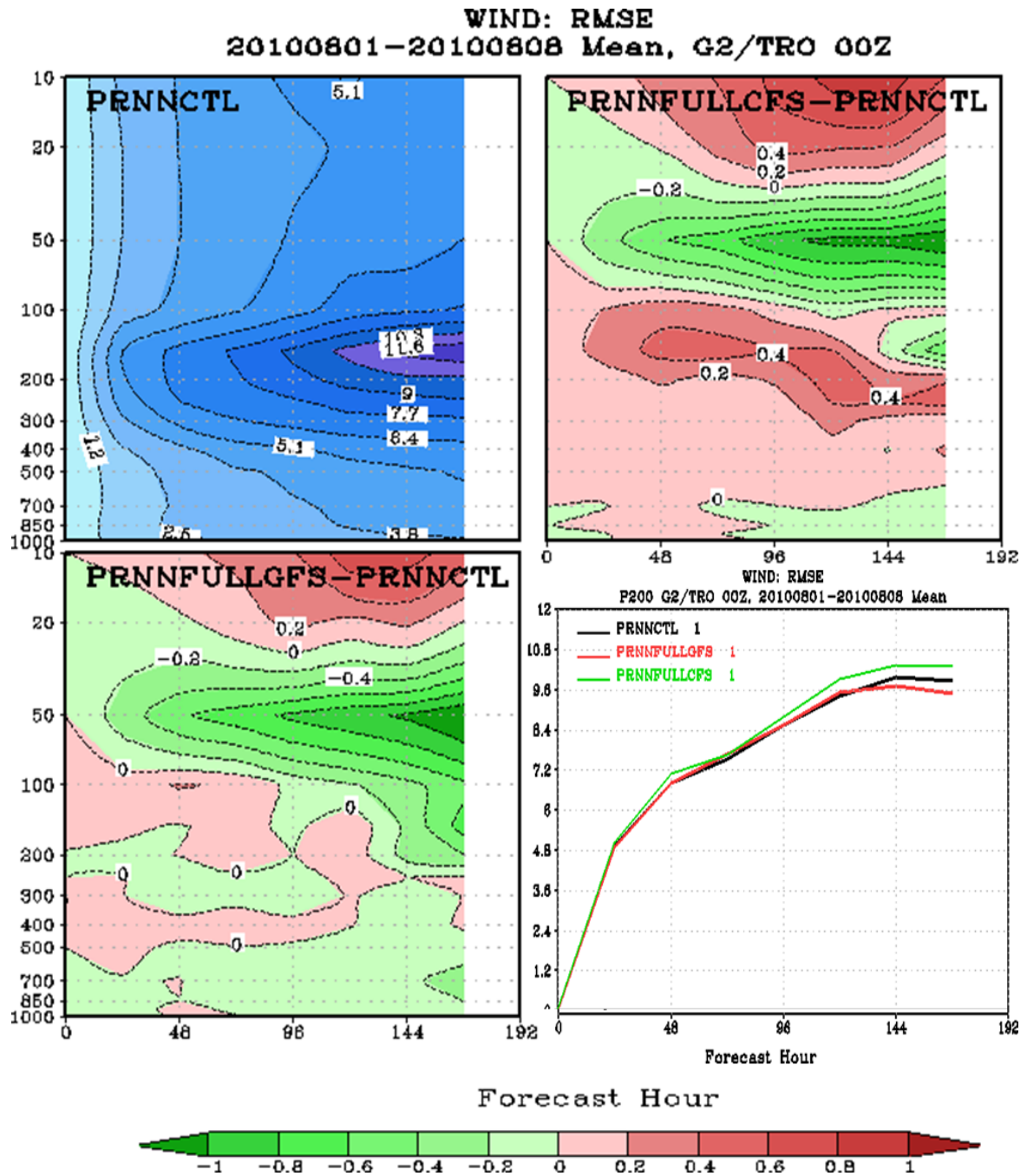


Fig.30 The same as in Fig. 29 calculated for the tropics.

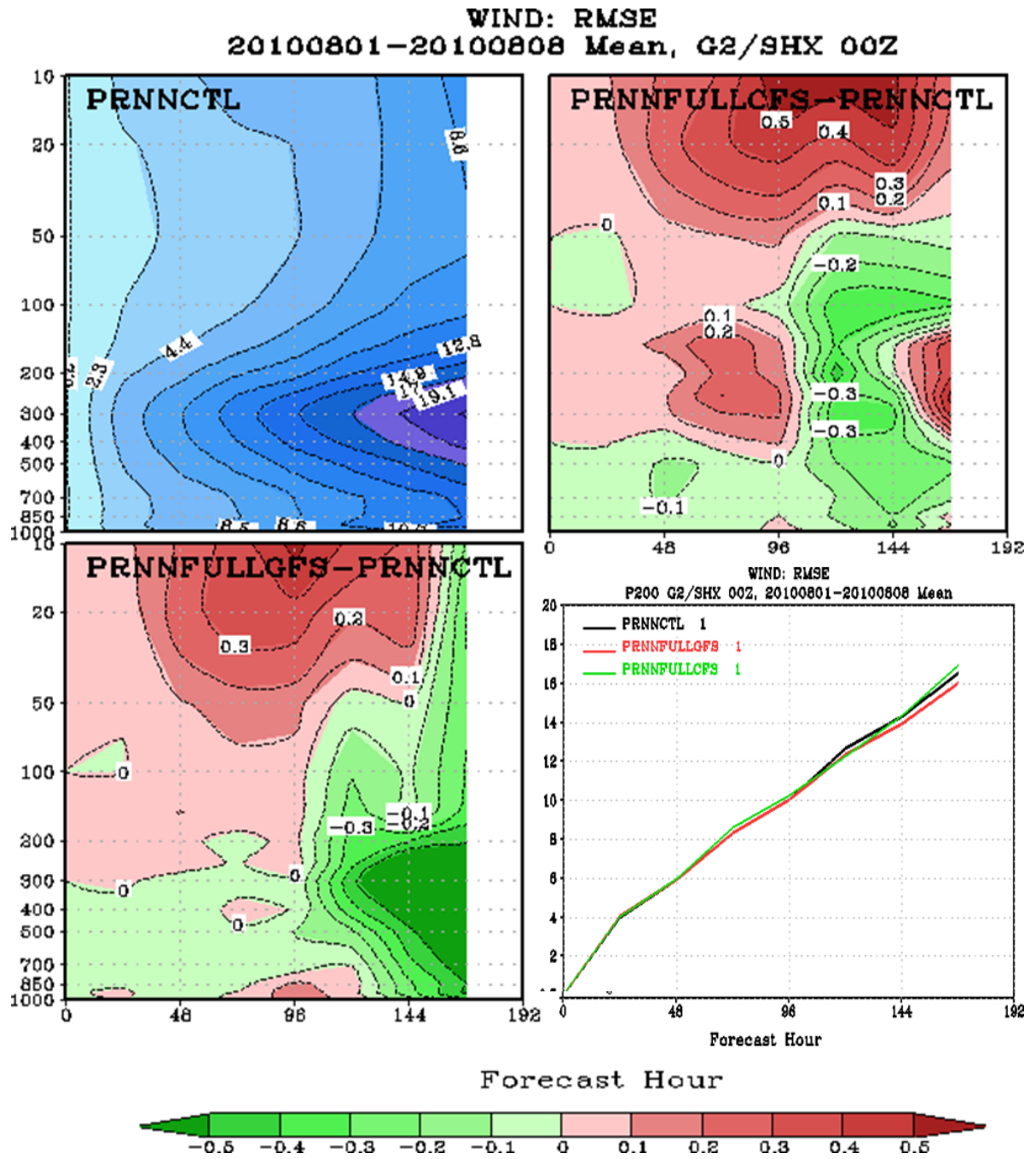


Fig.31 The same as in Fig. 29 calculated for the southern hemisphere.

4.4 Discussion

The results presented in this section show that the developed NN radiation is very accurate; the *PRNNGFS* run closely follows the control run, *PRNNCTL*. The differences between the *PRNNGFS* and control forecasts increase slowly with the forecast time (Figs. 23 to 31); however, in many cases *PRNNGFS* demonstrates slightly better results (higher anomaly correlation, lower bias and RMSE) at longer forecast times.

The additional *PRNNCFS* run allowed us to evaluate the robustness of the NN radiation and the NN emulation approach in general with respect to the changes in the model. The comparison of three runs (control, *PRNNGFS*, and *PRNNCFS*) demonstrated small differences between them, which shows high level of robustness of the developed NN radiation with respect to changes in the model environment. It shows that the developed NN radiation (CFS NN radiation) survived the transplantation from an old version of coupled model (CFS) to the newest version of uncoupled GFS. It also survived about 5 years of constant model evolution resulted in many changes in other than radiation physics parts of the model. After all these changes were made for the model, *PRNNCFS* still produces reasonable results comparable with those of *PRNNGFS*. This is a very important practical result, which shows that the NN radiation does not require frequent updates and may work in the model, if it is not changed very significantly, for many years without retraining. Of course, when the original radiation parameterization or the vertical resolution of the

model is changed, the NN radiation has to be retrained.

In addition to high accuracy, the developed NN GFS radiation is very fast. The high speed of NN radiation calculations can be used in several different ways:

1. The original radiation LW and SW parameterization can be simply substituted by the NN radiation and the NN radiation can be calculated in the model at the same frequency (once per hour). This is the least efficient use of the NN radiation, which provides a significant speedup of the total model integration of about 15-18%.
2. The GFS NN LWR is 20 times faster and NN SWR is 100 times faster than the original parameterization. This very significant speed up can be used to calculate radiation more frequently than once per hour (actually it can be calculated at each integration time step). Such a run will take as much time as the current run (with once per hour frequency of radiation calculations). In this case, the model run with the NN radiation would be many times faster than that with original parameterization calculated with the same frequency. Also in this case, in addition to a significant speedup, improvements in the quality of the forecast could be expected due to improvements in the radiation-cloud interaction.
3. The developed NN emulation approach can be used to emulate more advanced and time consuming radiation parameterization, which currently cannot be afforded in GFS. For example, NN emulation could be developed for the newest RRTM-McICA (Monte Carlo independent column approximation)

radiation, the most sophisticated but slowest version of RRTM radiation. In this case, use of the NN emulation approach could lead to improvements in model radiation physics and in the quality of the forecast.

It is noteworthy that in addition to the speedup of radiation calculations, the use of the NN radiation provides an additional significant advantage as compared to the use of the original parameterizations, namely it helps to achieve a significantly better load balance (Krasnopolsky et al 2010). The radiative transfer calculations take different time under different cloud conditions because of the different complexity of cloud-radiation interaction. For a more complex cloud-radiation interaction (deep convection) the calculation of the original LWR and SWR parameterizations takes ~22% and ~57% more time respectively than for clear sky conditions. Obviously, the time of the NN radiation calculations does not depend on the cloud conditions.

Thus the results presented in this section demonstrate the first successful step in evaluating NN radiation in GFS. Building upon these initial results, the further steps (by NCEP/EMC) may include:

1. More comprehensive tests in a longer series of 10-day forecasts.
2. Evaluation of the NN radiation in parallel runs with more frequent radiation calculations.

Also refinement of NN emulations for the GFS model based on longer training set, implementation of the concept of a compound parameterization including a quality

control procedure, and the NN ensemble approach (see chapters 6 and 7) will be introduced.

Chapter 5: Investigation of Other Approaches to Emulation of Model Physics: Tree Approximation of the Long Wave Radiation Parameterization in the NCAR CAM

5.1. Motivation

While artificial neural networks can be considered the current state-of-the-art black box methodology for a wide range of high-dimensional approximation problems, and justifiably so, they may not necessarily be the best solution for the application considered in this dissertation. The accuracy of neural network emulations depends on the number of layers and hidden neurons employed. While it is known that neural networks are universal approximators, i.e., they can approximate any continuous functions to any predetermined accuracy (see for instance Hornik (1989) and DeVore et al (1997)), this is achieved only by allowing the number of neurons to increase arbitrarily. Also, the learning of the network parameters (weights) requires the solution of a large, non-linear optimization problem, which is prone to deliver sub-optimal solutions.

While the application of neural networks to the problem of learning parameterizations has produced excellent results, it is not without limitations. Foremost among these is that the neurons have large support (a half space in \mathfrak{R}^n) and their superposition is a complex nonorthogonal expansion. This makes capturing of local or multiscale phenomena difficult. In our use of NN emulations in parameterization computation, we also notice that NNs can sometimes exhibit larger errors (with small probability) even for a good sampling. Thus, although both the systematic and the random errors are very small, there exists a very small probability of larger errors which needs to be

avoided (an approach to solution of this problem in terms of neural networks is described in the Chapter 6).

Additionally, the approximation is trained by a data set that consists of evaluations of the original parameterization gained during a reference run of the climate model. The inputs of this training data set, therefore, cover the physical states observed during a certain time period of climate history. However, the domain in which the parameterization is to be evaluated may change with time as in the case of climate change. In such situations the approximation may be forced to extrapolate beyond its generalization ability, which may lead to large errors. In this case it could become necessary to re-train the emulation in order to adapt it to the new environment.

This brings into question whether NNs are the ultimate SLT solution to numerically emulating parameterizations. Indeed, since our goal is to capture subtle multiscale phenomena, a more application oriented, responsive, and adaptive learning method could be useful. Working against us is the fact that our learning problems lie in high space dimension where computational issues are compelling.

High dimensional problems occur in many other learning settings and much attention has been given to the area. The main approaches are kernel methods, support vector machines (a special case being NN), and nearest neighbor algorithms (see Shakhnarovich et al., 2006 for general references). Each approach has tradeoffs. The advantage of NNs and kernel methods is that they can be implemented in high dimensions without meshing or splitting up of the domain of the approximated mapping into sub-domains. Their disadvantage is twofold. Firstly, they are not local by which we mean the building blocks are not locally supported functions. This

means that local features are reflected in many or all terms of the function expansions. This is analogous to Fourier methods. (One of the great advances of the last decades in image processing has been to replace Fourier methods by the more local wavelet methods.) A second disadvantage is that they do not include adaptivity. For example, if the function F to be learned is known to have regions of smoothness and regions of singularities then the methods do not adapt to such features. Nearest neighbor methods have a local flavor but they are typically not implemented adaptively. In other words, the rules for identifying nearest neighbors do not take into account the variability in the underlying function (which will be reflected in the data). Nevertheless this approach would be closest to the methods we wish to employ. The severe obstructions encountered when dealing with problems in high spatial dimensions is often called the *curse of dimensionality*.

In this chapter we will use an alternative to neural networks within the class of non-parametric approximation methods. We restrict ourselves to basic design decisions and discuss the features of two common statistical learning paradigms, (approximate) nearest neighbors and regression trees.

5.2 Description of Algorithms

In order to keep this chapter self-contained we give a concise description of the non-parametric algorithms that will be used in the following numerical experiments.

5.2.1 Approximate Nearest Neighbors

Non-parametric learning methods typically try to partition the input space and then use simple local models like piecewise constants to approximate the data. In the case

of nearest neighbor methods, the input space is implicitly partitioned by the way the training data is distributed: the approximation is constant for query points that have the same set of nearest neighbors. Unfortunately in high dimensions there are no fast algorithms which could answer the question “what are the nearest neighbors to a given query point x ?” Therefore one must be content with approximate answers to this question. Here, assuming that all the training data is available beforehand, the input domain is recursively partitioned depending on the distribution of the input points.

The k-nearest neighbor method works as follows: one defines a metric $\|\cdot\|$ on the input space and given a query point x finds a permutation $i \rightarrow i_n$ in of the training data such that

$$\|x^{i_1} - x\| \leq \|x^{i_2} - x\| \leq \dots \leq \|x^{i_k} - x\| \leq \|x^{i_p} - x\|$$

for all $p > k$. Then, one averages the function values corresponding to these nearest neighbors

$$\tilde{f}(x) = \sum_{n=1}^k y^{i_n}.$$

to define an approximation of $f(x)$. Unfortunately, it is well known that in very high dimensions it is not possible to design fast algorithms that provide the permutation sought. Instead one relaxes the search and is content with an algorithm that returns points \bar{x}^{i_n} such that

$$\|\bar{x}^{i_n} - x\| < (1 + \varepsilon)\|x^{i_n} - x\|.$$

There are algorithms based on *kd*-trees or *bd*-trees that provide a fast answer to this relaxed problem, if ε is chosen large enough. An introduction to this topics can be found in Wendland (2005).

The central point in the above description is, of course, how to define the metric $\| \cdot \|$ on the input space. This is a non-trivial task because the input vectors include several physical quantities measured in different units and are varying over several orders of magnitude. A trivial method to equilibrate the various input quantities is to compute the maximum and minimum of each parameter in the training data set and to then scale this each component of the input vector individually to the interval $[0, 1]$. Then, one uses the standard Euclidian norm on $[0,1]^d$ to measure the distances. Another self-evident idea is to scale the variables belonging to the same profile with the same factors. Numerical experiments showed that the second type of scaling yields better results. Therefore, we use this scaling in the following experiments. Adaptive nearest neighbor methods try to learn a problem dependent metric from the data, but we have not pursued this approach any further, because the data seems to be too sparse to define local metrics reliably for this application.

5.2.2 Regression Trees

Regression trees follow a more adaptive approach and also use the y -values in order to define the domain partition. Here, starting with the entire input domain, the cells in the partition are recursively subdivided such that the residual of the resulting approximation is minimized in each step. Obviously, due to their recursive definition, none of these techniques is available for incremental learning without modification: a

new data point could theoretically change the decision how to perform the first split in the tree, which would require relearning the tree from the very beginning.

The most basic algorithm for the generation of regression trees is the Classification and Regression Tree (CART) algorithm intensively analyzed in Breiman et al (1984).

It can be summarized as follows: we initialize the partition $P = \{\Omega\}$ where

$\Omega = \prod_{i=1}^d [a_i, b_i]$ is a hyper-rectangle that contains all the training points and d is the

dimension of the input space. Then, each hyper-rectangle in the partition that contains more than a given number m of data points is recursively subdivided along a hyperplane $x_i = c$, where $i \in \{1, \dots, d\}$ and $c \in [a_i, b_i]$ is chosen such that the RMSE of the best piecewise constant approximation on the refined partition is minimized. That is, the regression function assumes the average value of all the points in a given hyper-rectangle of the partition. This is a reasonably efficient algorithm that can be used for a wide range of classification and regression problems. It also has the advantage that it is independent of the scaling of the data.

The Random Forests algorithm by Breiman (2001) averages the response of a given number T of CART trees. Hereby, before each subdivision step in the generation of the CART trees, the algorithm chooses a random subset of size P of the input parameters (typically about one third of all input parameters) along which the cell is allowed to be subdivided. This ensures that each single CART trees generates different partitions of the domain. Random Forest approximations are relatively smooth due to the averaging process. Random Forests are generally considered to be one of the best available black-box non-parametric regression methods.

5.3 Numerical Experiments

In this section we present the results of three groups of numerical experiments. In the first two subsections, the sets of training data and test data each contain 196,608 data samples collected during a reference climate simulation for the years 1961–1962 using the original parameterization. In the first subsection we compare the regression trees with the benchmark neural network approximation. Note that this comparison tends to overestimate the advantages of the neural network. First of all, it does not reflect the training time, which is about a week for the neural network, but only between a few seconds and less than a few hours for the tree-based methods. Second, whereas the neural network would profit only slightly from taking more training data (its accuracy is basically limited by the number of neurons), the non-parametric methods benefits significantly from allowing more data, and limiting the training data size is artificial and unnecessary. Nevertheless, we perform the comparison in this form, because it's the same training data we will use for the experiment in the subsection 5.4 where we have to comply with memory limitations. As it turns out, nearest neighbor methods and sparse occupancy trees do not deliver competitive accuracy, if applied naively, but their performance can be enhanced by dimension reduction.

In Figure 32 we see the RMSE profiles (left) and the bias profiles (right) for the following methods:

1. The benchmark neural network emulation (blue line)

2. The approximate nearest neighbor approximation (ANN, with $k = 5$, $\varepsilon = 1$, red line), where we used the profile-wise input scaling.
3. A single vector-valued regression tree (CARTV, cyan line).
4. One regression tree for each output component individually (CARTC, magenta line).
5. A vector-valued Random Forest approximation (RFV, $T = 20$, $P = 80$, green line), and
6. An approximation where we compute a Random Forest (RFC, $T = 20$, $P = 80$, black line) for each component individually.

In Table 6 we also give the total RMSEs and bias for all these methods.

Method	RMSE (J/kg/s)	Bias (J/ks/s)
Neural Net	$3.94836 \cdot 10^{-3}$	$3.11643 \cdot 10^{-6}$
ANN	$7.26535 \cdot 10^{-3}$	$2.70421 \cdot 10^{-5}$
CARTV	$8.17753 \cdot 10^{-3}$	$1.27022 \cdot 10^{-5}$
CARTC	$5.54573 \cdot 10^{-3}$	$1.26559 \cdot 10^{-6}$
RFV(20,80)	$4.75692 \cdot 10^{-3}$	$6.08371 \cdot 10^{-6}$
RFC(20,80)	$3.27711 \cdot 10^{-3}$	$3.99269 \cdot 10^{-7}$

Table 6: Total RMSE and absolute value of total bias for emulation with neural network, approximate nearest neighbors, CART and Random Forests applied to the whole vector or component-wise. Training and test data each consist of 196,608 evaluations of the original parameterization.

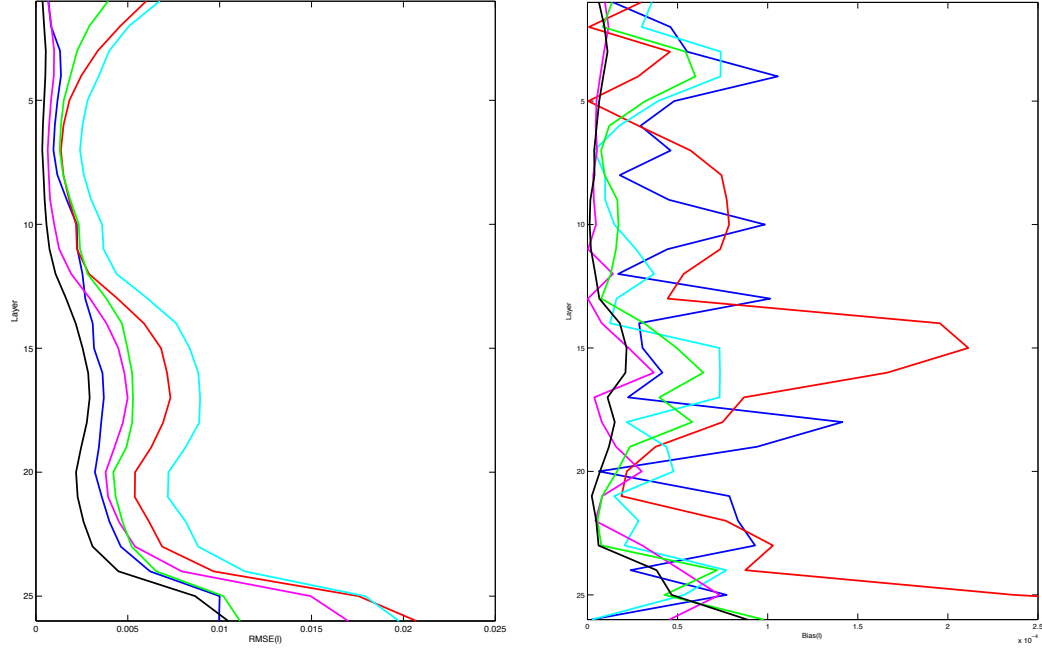


Figure 32: Comparison of neural network, approximate nearest neighbor, and several regression tree emulations. Left: layer-wise root mean square errors. Right: layer-wise absolute values of Bias.

Some major observations to be taken from Figure 32 and Table 6 can be summarized as follows:

1. Nearest Neighbors do not deliver competitive accuracy, if applied directly to the 220-dimensional input data. It is however surprising that the vector-valued CART does not yield a better result, even though it generates an adaptive partition of the input domain. One needs to use ensembles of regression trees to achieve good approximation accuracy.
2. It is possible to improve on the neural network emulation with moderate computational effort. The generation of regression trees is cheap, so even the generation of the 520 trees for RFC(20,80) takes only a few hours (7h on a

single 2.2 Ghz AMD-Opteron processor) on a standard PC. However, due to its storage requirements ($26 \cdot 20 = 520$ trees have to be computed) this result is not of great practical interest. The two practical competitors are the CARTC and the RFV emulations, which use 26 trees (one for each component) or 20 trees, respectively. RFV seems to be a little bit more accurate, but CARTC has a lower bias. To demonstrate the latter point we show in Figure 33 scatterplots for both emulations. The component-wise CART approximation clearly has a higher variance in layer 26, but delivers good, unbiased approximation in layer 9.

3. Notice that except for the somewhat inaccurate nearest neighbor approximation, the neural network approximation exhibits the most biased approximation.

Finally, in Figure 34 we show the emulated heating rates for three representative profiles in order to compare the vector-valued random forest and the component-wise CART approximation. In general CART very accurately follows the profile of the original parameterization. However, in extraordinary cases it can overshoot, which is most noticeable in the third graph. The random forest approximation has the tendency to flatten out the original profiles but does not produce extreme outliers.

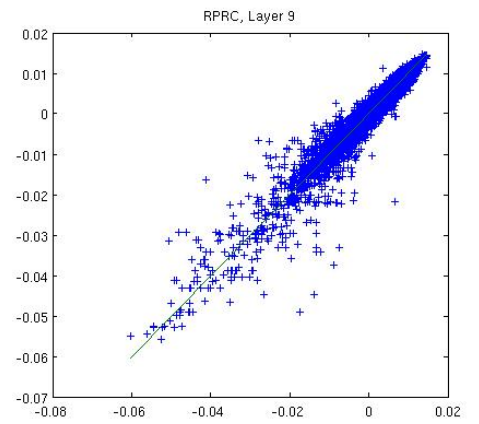
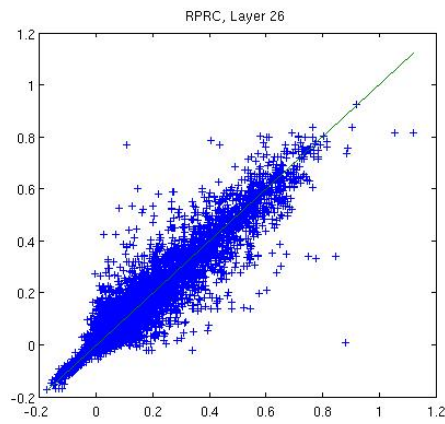
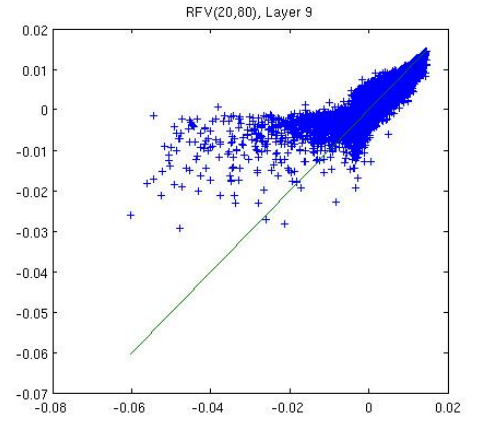
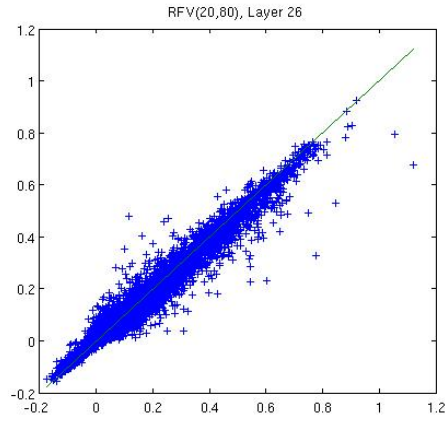


Figure 33: Scatterplots for the approximation of the heating rates in the 26th and 9th vertical layer with component-wise CART (RPRC) and the vector-valued Random Forest (RFV) approximation.

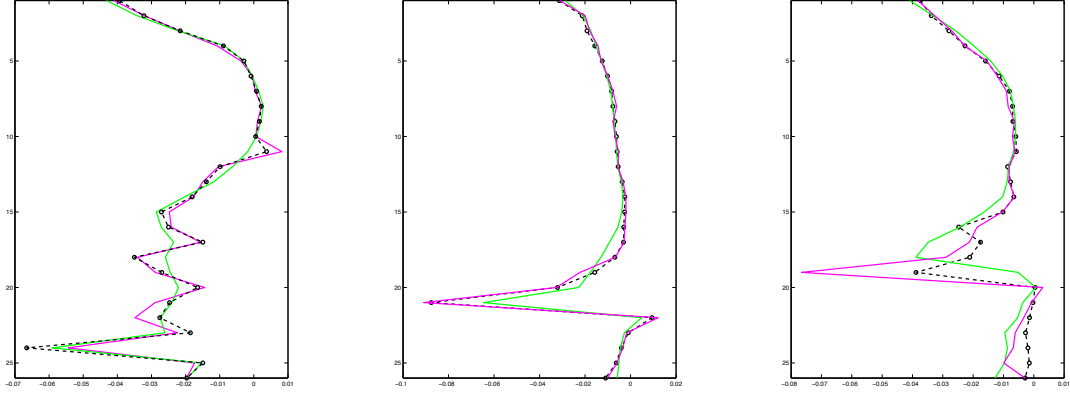


Figure 34: Approximation of three representative heating rate profiles. Black line with markers: original parameterization. Magenta: Component-wise emulation. Green: vector-valued Random-Forest emulation. Heating rates units are $J/kg/s$.

5.4 Results of a 10-Year Climate Simulation

Finally, we try to assess the impact of using a tree approximation of the LWR parameterization in a climate simulation. Therefore, we run the NCAR CAM v 2.0 for 10 years using the original parameterization, the neural network emulation and the component-wise CART emulation. As discussed above, this choice of tree design is debatable, since we could have achieved a more stable and more accurate (in terms of the RMSE) approximation using a vector-valued random forest design. However, the CART-design was the best available method at the time the experiment was set up. The relatively small training data set with its 196,608 samples was used because the parallel simulation was performed on a distributed memory systems, where each processor could only address 4GB of memory, the emulation had to be stored on each processor, and on each processor most of the memory had to be reserved for other parts of the simulation. Therefore, we do not give any numbers about the achieved

speed-up of the GCM, although even under these imperfect conditions, the speed up was still considerable.

One of the most desirable properties of emulation is the preservation of the time means of the prognostic and diagnostic fields. As it has been shown in the previous chapters, that neural network emulations reliably achieve this aim. The CART emulation in general produces good agreements, too, but it also seems to be more prone to produce local instabilities. As an example we consider the annual zonal means of the LW radiation heating rates (QRL) in Figure 35. Whereas the plots in the left column seem to be in very good agreement, the difference plots in the right column reveal that the CART approximation causes significant differences in the forecast in the lower atmospheric layers near the polar regions.

The same observation can be made, if we look at the annual means of the two-meter air temperature in Figure 36. The agreement of the control run with the tree-emulation run is satisfactory, but a comparison of the difference plots in the right column reveals that the neural network run is closer to the original parameterization. Again, we see that the largest differences occur in the polar regions.

For this reason we checked the approximation accuracy for all test data samples stemming from these regions separately. It turned out that the RMSE's were much worse for these points than for other regions of the earth, and the CART emulation was biased towards predicting higher heating rates than the original parameterization. The reason for this seems to be that the extreme weather conditions at the poles are represented only by a small fraction of all training samples. As remedy for this problem one can train separate approximation modules for different regions on the

earth, or balance the distribution of the training such that the statistical approximation error is equally distributed over the whole globe. The neural network approximation seems to be more reliable with regard to the generalization to rare states. The vector-valued random forest approximation also seems to be stable in this sense.

5.5 Discussion

In this chapter we investigated the possibility of substituting physical parameterizations in global climate models with non-parametric emulations (Belochitski et al., 2011). The results are positive in the sense that they show that both nearest neighbor type methods and regression trees are in principle able to achieve statistical approximation quality on par with neural networks, even if trained with a relatively moderate amount of data. It has been demonstrated that the NCAR CAM with a tree-based LWR emulation gave results in good agreement with the calculation using the original parameterization, except in the polar regions, which could have been expected from the statistical properties of the approximation. The main obstacle for the practical use of non-parametric methods is less a mathematical one, but rather one of implementation. Non-parametric approximation methods are memory-based, i.e., they need to store all the training data permanently. This makes its use in a parallel environment more difficult than is the case for the relatively compact neural network representation.. Therefore, the ideas and results presented in the current section can only be considered as a preliminary step towards a new emulation paradigm.

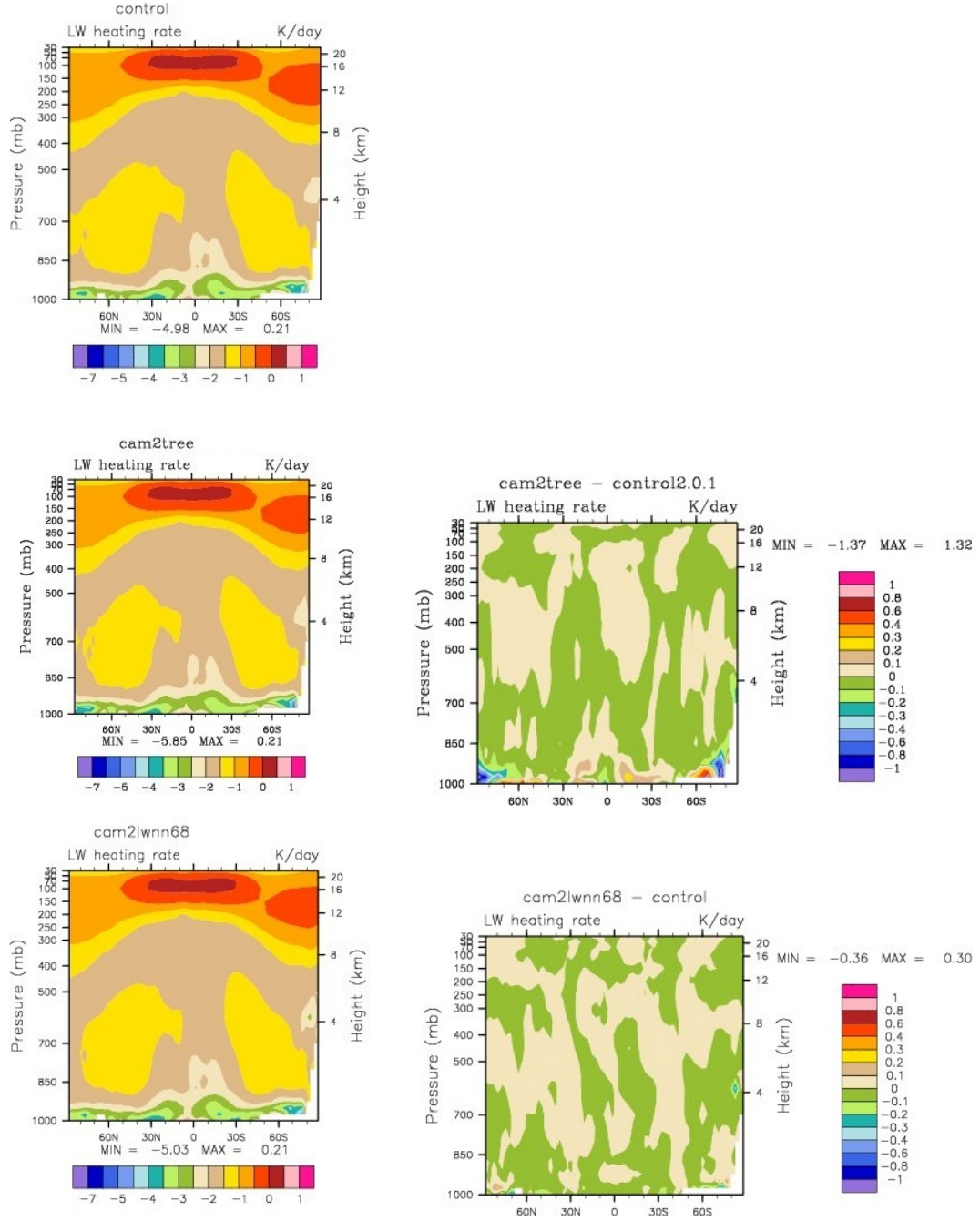


Figure 35: Comparison of the predicted annual zonal means of the LWR heating rates computed with the original parameterization (top row), a tree based emulation (center row) and a neural network emulation (bottom row). The right column plots the difference between the simulation and the control.

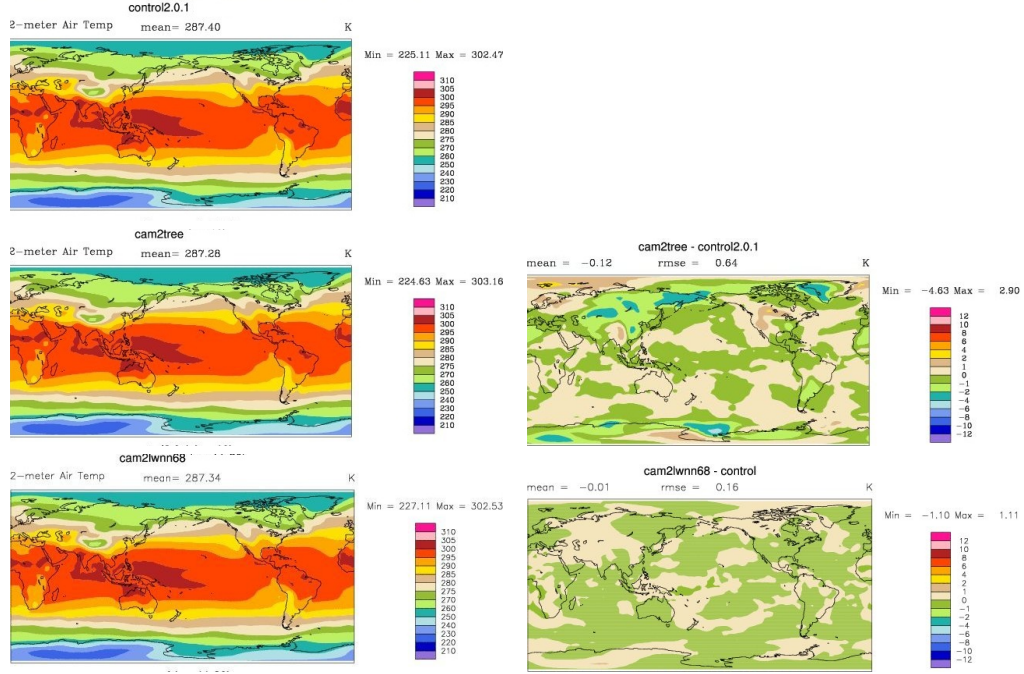


Figure 36: Comparison of the predicted annual means of the two meter air temperatures computed with the original parameterization (top row), a tree-based emulation (center row) and a neural network emulation (bottom row). The right column plots the difference between the simulation and the control.

Chapter 6: Compound Parameterization of Full Radiation with a Quality Control of Larger Errors in NCAR CAM.

6.1 Accuracy and quality control of NN emulations

Tremendous complexity, multidimensionality, and nonlinearity of the climate/weather system and numerical models describing this system lead to complexity and multidimensionality of our NN emulations and data sets that are used for their development and validation. Also, the validation procedure for developed NN emulations becomes more complicated because, after their development, they are supposed to work in a complex and essentially nonlinear numerical model. The development of NN emulations of model physics and their accuracy depends significantly on our ability to generate a representative training set to avoid using NNs for extrapolation beyond the domain covered by the training set. Owing to the high dimensionality of the input domain (i.e., dimensionality of the NN input vector) which is of the order of several hundreds or more, it is difficult if not impossible to cover the entire domain, especially its “far corners” associated with rare or extreme events, even when we use model simulated data for the NN training. Also, the domain may change with time as in the case of climate change. In such situations the emulating NN may be forced to extrapolate beyond its generalization ability which may lead to larger errors in NN outputs and, as a result, to errors in the numerical models in which they are used.

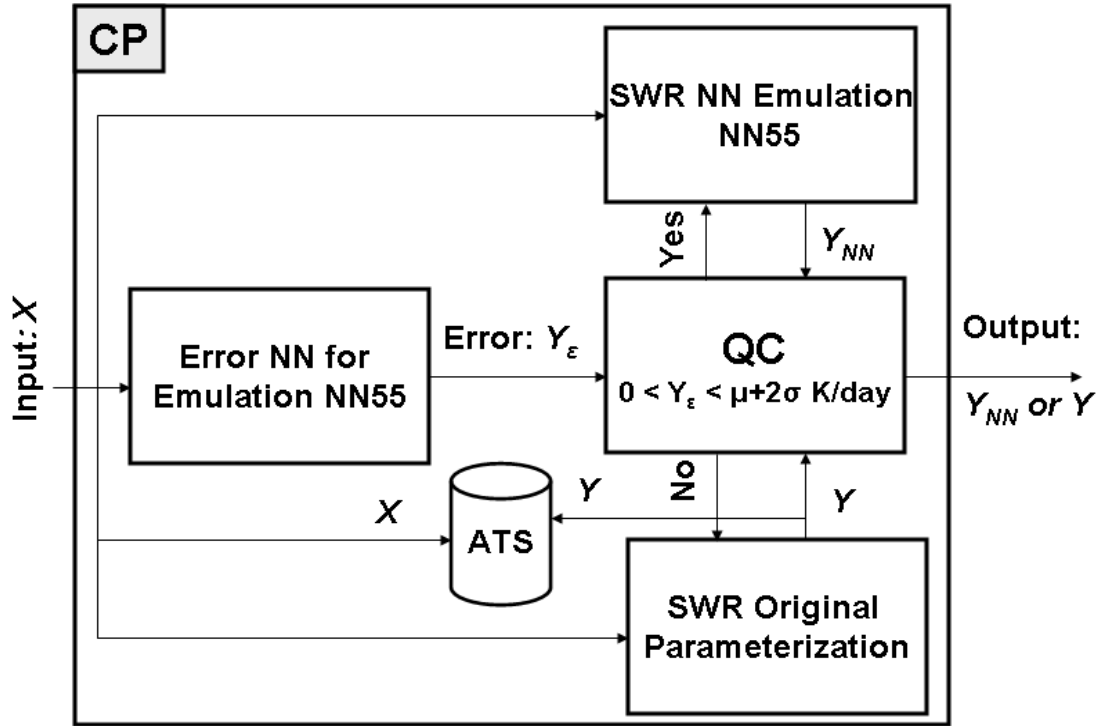


Fig. 37. Compound parameterization design for the NCAR CAM SWR. For each SWR NN emulation (NN55, in this case), additional NNs (Error NN) is trained specifically for predicting, for a particular input, X , the errors, Y_ϵ , in the NN emulation output Y_{NN} . If these errors do not exceed a predefined threshold (in this case, the mean value plus two standard deviations), the SWR NN emulation (NN55) is used; otherwise, the original SWR parameterization is used instead of the NN emulation. ATS stands for the auxiliary training set that is updated each time when QC requires using the original parameterization instead of NN emulation. ATS is used for the follow-up dynamical adjustment of the NN emulation.

The developed NN emulations are very accurate. Larger errors and outliers (a few extreme errors) in NN emulation outputs occur only when NN emulations are exposed to inputs not represented sufficiently in the training set. These errors have a very low probability (see Fig. 39) and are distributed randomly in space and time. However, when long multi-decadal climate simulations are performed and NN emulations are used in a very complex and essentially nonlinear climate model for such a long integration time, the probability for occurrence of larger errors and the probability of their undesirable impact on the model simulations increase. As we learned from our experiments with NCAR CAM, the model was in many but not in all cases (shown, for example, in Fig. 41) robust enough to overcome such randomly distributed errors without their accumulation in time. However, for these few cases, it is still essential to develop and use for NN emulations an internal quality control (QC) procedure capable of controlling their larger errors.

Therefore, it is essential to introduce a QC procedure, which can predict and eliminate larger errors of NN emulations during the integration of highly nonlinear numerical models, not just relying upon the robustness of the model that can vary significantly for different models. Such a mechanism would make our NN emulation approach more reliable, robust, and generic. In this chapter we introduce a *compound parameterization* (CP), which combines NN emulation with a QC technique.

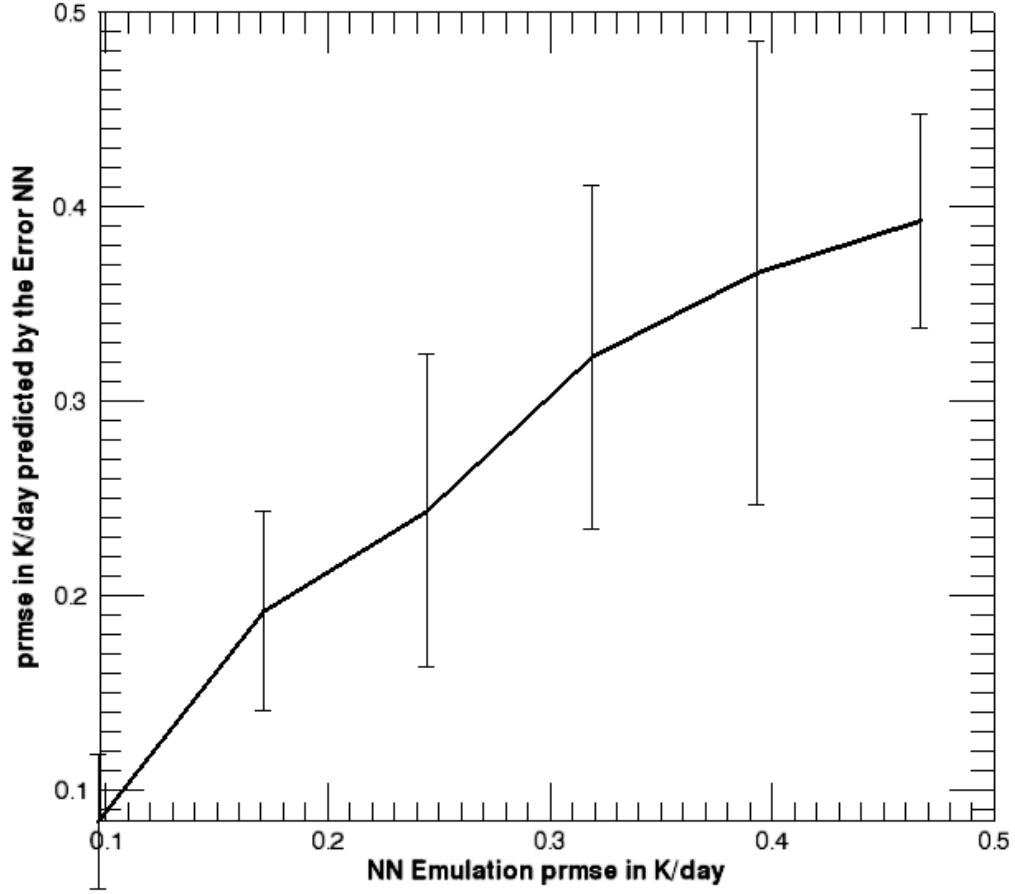


Fig. 38. The correlation (binned scatter plot, the error bar shows the standard deviation inside the bin) between the actual error (prmse of the NN emulation NN55) and the error predicted by the error NN calculated vs. the original parameterization on an independent test data set. The correlation coefficient between the two errors is 0.87.

6.2 Compound Parameterization Designs and Their Validation on Independent Data Sets.

The final goal of our developments is a stable functioning of the NN emulation in the complex nonlinear numerical model for a sufficiently long time and the similarity of the model results produced with the original component (the control run) and with the NN emulation of this component. For such a situation, the high accuracy of a NN emulation obtained on an independent test set does not guarantee its stable performance in a numerical model. Thus, in our case, a reasonably good accuracy of NN emulation on a test set is a necessary but not sufficient condition for the satisfactory validation of NN emulation. This is only the first step of the two-step validation procedure used for the validation of the developed NN emulations in the previous chapters, and also used for validation of CPs developed in this study. The second and the most important step of the validation procedure is the validation of the model run with NN emulation vs. the control run with the original parameterization. During this second validation step, the run with the NN emulation (or with CP) should demonstrate, in addition to its stable performance, a close similarity of all simulated results to those of the control run.

CP consists of the following three components: the original parameterization, its NN emulation, and a quality control (QC) block (see Figs. 37 and 40). During a routine numerical model simulation with CP, the QC block determines (based on some criteria presented below, at each time step of model integration and at each grid point) whether either the NN emulation or the original parameterization has to be used to

generate physical parameters (i.e. parameterization outputs). Namely, when the NN emulation errors are large (i.e., they exceed an error threshold) for a particular grid point and time step, the original parameterization is used instead of NN emulation. When the original parameterization is used instead of the NN emulation, its inputs and outputs are saved to further adjust the NN emulation. Although it goes beyond the scope of this study, it is worth mentioning that after accumulating a sufficient number of these records, an adjustment of the NN emulation can be produced by a short retraining using the accumulated input/output records. Thus, CP can be used for the development of NN emulations that become dynamically adjusted to the changes and/or new events/states produced by a complex environmental or climate system.

There are different possible QC designs considered for CP. The first and simplest QC design is based on a set of regular physical and statistical tests similar to those used for QC of meteorological observations (e.g., Dee et al (2001) and Gandin (1988)). Such approaches can be used to check the consistency of NN outputs. These are the simplest, most generic but not sufficiently flexible approaches. Statistical tests that are usually based on linear statistical correlations between inputs and/or outputs and errors in outputs, work not so well for larger and extreme errors, which are usually caused by nonlinear correlations (e.g., see Fig. 38). Statistical criteria are usually global and based on past data. They may not be sensitive enough to local perturbations and also to new situations emerging in the course of integration of a complex environmental or climate system due to the change of its simulated environment, such as an evolving climate change. When applied to NN emulation outputs, such criteria give a significant amount of false alarms. In the context of our

complex system application, which includes also a trade-off between the accuracy of an NN emulation and its computational performance, such a significant amount of false alarms leads to a significant reduction in the computational performance of CP. Namely, each false alarm leads to a rejection of an accurate (but falsely suspected) and fast NN emulation and to its unnecessary replacement by the time consuming original parameterization. Owing to these significant problems important for our CP application, the above simple statistical QC design was not used in this study.

The second and more sophisticated, nonlinear, and effective QC design is based on training an additional NN to specifically predict the errors of the NN emulation outputs for a particular input (Krasnopolsky and Fox-Rabinovitz, 2006). The error NN has the same inputs as the NN emulation and one or several outputs — errors of outputs generated by the emulation NN for these inputs. In this work, we used an error metric that produces one error for all outputs (given by eq. (9)); thus our error NN has one output. During the model integration, if this error does not exceed a predefined threshold, the NN emulation is used; otherwise, the original parameterization is used instead. An example of application of this CP design (see Fig. 37) is presented below for the NCAR CAM SWR.

Fig. 38 shows the results of the calculations performed with the data set containing more than 100,000 records, each of which consists of the error predicted by the error NN and the actual error of the NN emulation. The actual errors of the NN emulation were binned and for each bin a corresponding mean errors predicted by the error NN and its standard deviations were calculated and plotted as a curve with error bars. The plot with only six bins is presented in Fig. 38 for simplicity and convenience of

presentation. Increasing the number of bins does not change the dependence significantly. Fig. 38 shows a very strong correlation between the error predicted by the error NN and the actual error of the NN emulation (SWR NN55) calculated vs. the SWR original parameterization on an independent test data set. The dependence, linear for small errors, becomes nonlinear for larger errors. The high, 0.87, correlation coefficient is obtained between these two errors calculated on the entire 100,000 records long test set.

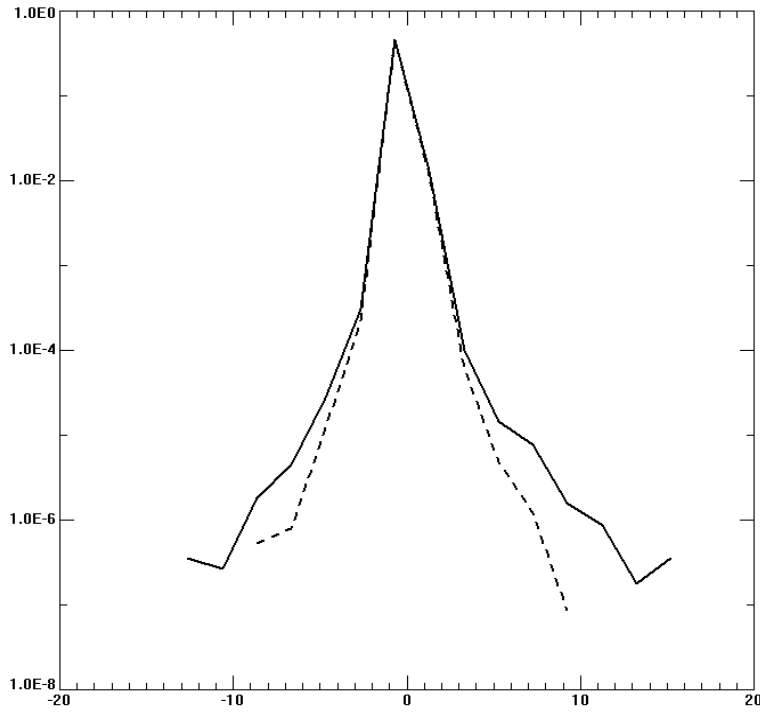


Fig. 39. Probability density distributions of emulation errors for the SWR NN emulation NN55 (solid line) and for the compound SWR parameterization shown in Fig. 37. The vertical axis is logarithmic and shows the error probability; the horizontal axis shows the NN emulation errors in K/Day. In both cases errors are calculated vs. the original SWR parameterization. The CP reduces the probability of medium and large errors by an order of magnitude.

Fig. 39 shows the comparison of two error probability density functions. One curve (solid line) corresponds to the NN55 emulation errors, another (dashed line) corresponds to the CP emulation errors shown in Fig. 37 (both errors are calculated vs. the original parameterization on the independent test set; the vertical axis is logarithmic). Fig. 39 demonstrates the effectiveness of CP; the application of CP reduces medium and large errors by about an order of magnitude. This is presented by the differences between the solid and dashed lines for NN emulation errors exceeding 5–10 or more K/day.

Fig. 40 demonstrates the effectiveness of CP in removing outliers, and Table 7 shows improvements in other statistical measures. The use of CP: (a) does not increase the systematic error (bias) which is almost zero; and (b) significantly reduces the random error. Especially significant is the reduction of extreme errors or outliers. It is noteworthy that for this CP and for this validation data set, less than 1% of the SWR NN55 emulation outputs are rejected by QC and calculated using the original SWR parameterization. Further refinement of the criteria used in the QC may result in a further significant reduction in the already small percentage of outliers as it will be shown below.

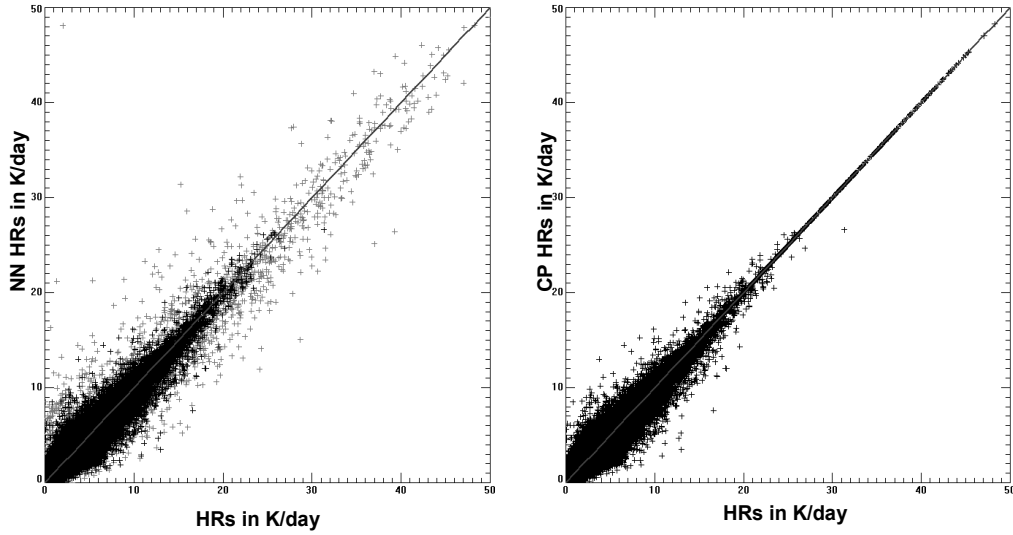


Fig. 40. Scatter plot for HRs (heating rates) calculated using the SWR NN emulation NN55 (the left panel) vs. the original SWR parameterization (left and right horizontal axes) and for HRs calculated using the SWR compound parameterization (the right panel) vs. the original SWR parameterization. Gray crosses (the left panel) show outliers that are eliminated by the compound parameterization (the right panel).

Table 7. Error Statistics for SWR NN Emulation NN55 and SWR Compound Parameterization: Bias and total RMSE, $RMSE_{26}$ at the lower model level, and Extreme Outliers (Min Error & Max Error). These statistics have been calculated on independent one year long test set.

	Bias	RMSE	RMSE₂₆	Min Error	Max Error
SWR NN55	$4 \cdot 10^{-3}$	0.19	0.43	-46.1	13.6
SWR CP	$4 \cdot 10^{-3}$	0.17	0.30	-9.2	9.5

6.3 Validation of Compound Parameterization in NCAR CAM

The second CP design outlined above has been implemented into NCAR CAM using the SWR NN55 emulation. A number of 50-year model simulations have been performed with the QC procedure using different thresholds. An appropriate threshold of 0.5 K/day has been determined experimentally. In this context, choosing an appropriate threshold means that the selected threshold (which is approximately equal to $\mu + 2\sigma$) does not allow for even limited accumulation of errors (see the light gray line in Fig. 41) during the CAM simulation and, at the same time, does not practically reduce the computational speed-up gained by using the fast NN emulation. Thus, at each integration time step and at each grid point of the model with CP, the error NN, that predicts the error of the NN emulation, was estimated, and if the predicted error did not exceed 0.5 K/day, the NN emulation outputs were calculated and used in the model; otherwise the original parameterization was calculated and its outputs were used in the model.

The example shown in Fig. 41 illustrates the effectiveness of CP in eliminating any accumulation of errors in the course of the model integration. When the model is integrated without QC, the SWR NN emulation NN55 produces moderately increased errors (errors increase from 0.07 K/day to 0.14 K/day) during the period between 24th and 25th years of the integration (the gray curve in Fig. 41). The error NN predicts this increase of the errors very well (the black curve in Fig. 41). After the QC was turned on, that is the model was integrated with the CP, the level of errors dropped significantly in general and, what is even more important, the bump between 24th and 25th years disappeared completely (the light gray curve in Fig. 41).

Using CP provides a stable and reduced error environment for model simulations compared to the model simulations performed without QC. It is noteworthy that, at each time step, the NN emulation outputs were rejected by the QC and the original parameterization was used instead mostly only for 0.05%–0.1% but below 0.4%–0.6% of model grid points, throughout the entire 50-year model simulation. Therefore, the computational performance of the model with NN emulation was practically not reduced and CP is still about 20 times faster than the original SWR parameterization.

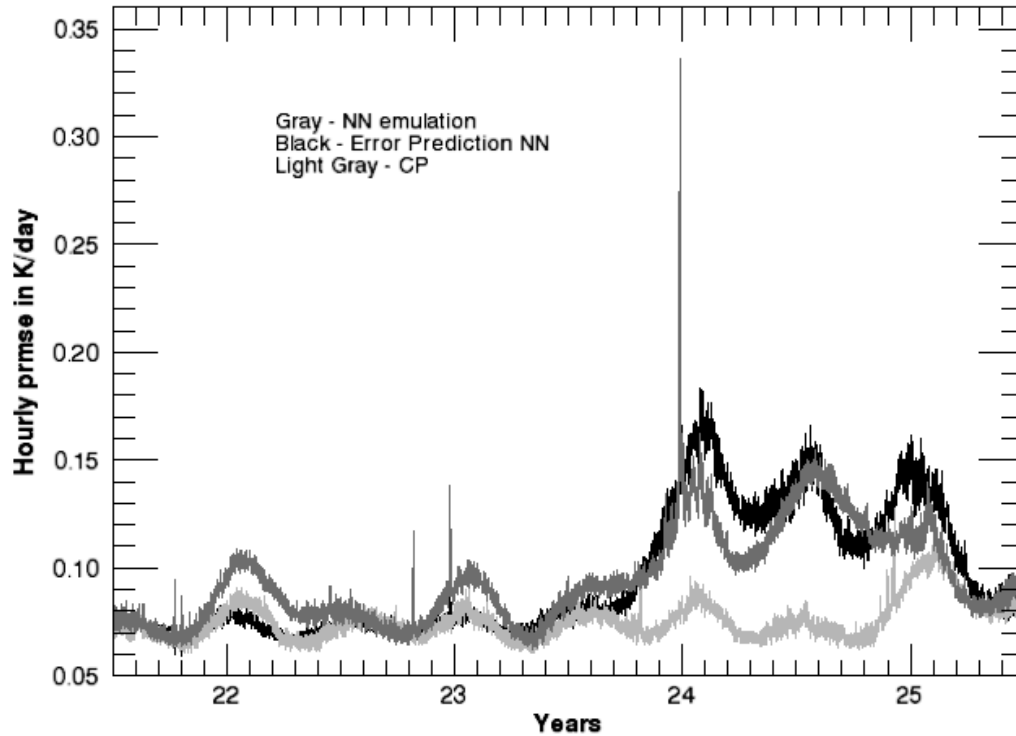


Fig. 41. Errors (vs. the original SWR parameterization) produced by the SWR NN emulation during the model run (gray line), errors predicted by the error NN (black line), and errors produced after introducing CP instead of the SWR NN emulation (light gray line).

6.4 Discussion

A new improved NN emulation approach called a compound parameterization, which incorporates NN-based quality control techniques for controlling larger errors of NN emulations, has been developed. One design of a compound parameterization presented in this

chapter uses a special NN trained to predict errors in outputs of NN emulation of a climate model physics component. It is shown that the accurate representation of a model physics component using a compound parameterization with a quality control of larger errors is essential for successful climate simulations.

The CP approach can be considered as an engineering solution that does not investigate the problem (why on some rare occasions a NN emulation does not perform well) but bypasses it allowing using this NN emulation safely in the essentially nonlinear and complex environment of a numerical model. If a second error NN can be trained to reliably predict errors of the NN emulation, then it looks like these errors can be investigated, explained, and eliminated by correction of the NN emulation itself. Theoretically speaking, this is correct. However, practically speaking, it is hardly possible. As we have mentioned before, the main reason for such larger errors to occur is our inability to generate a completely representative training set, that is, to get each far corner of the domain of the mapping (1) represented. For modern climate and weather models this domain has dimensionality of the order of 10^3 and higher. A systematic investigation of such an object is a formidable task that requires significant special efforts. Using CP allows us to flag

and to bypass these questionable far corners of the domain leaving their investigation for the future research.

There is also another important aspect of this problem: some larger NN emulation errors are ignored by the numerical model where this NN emulation is introduced; whereas some other larger NN emulation errors cause a significant reaction of the model like the one presented in Fig. 41 (the bump between 24th and 25th years). Currently, we can only speculate why such larger differences between the NN emulation and the original parameterization happen and why the reaction of the model is so different. It is worth noting that the original parameterization is an approximate physical model that itself may have discontinuities and inconsistencies. Actually, some of the larger NN emulation errors can be caused by such inconsistencies and discontinuities; in these cases the NN emulation “errors” may lead to smoother physics and a better performance of the model. Further investigation of these problems is very important and illuminating; it can also provide a valuable feedback to developers of model physics parameterizations.

Chapter 7: Ensemble of Neural Network Emulations for Climate Model Physics: The Impact on Climate Simulations.

During the last decade, the ensemble approach demonstrated a significant success in numerical weather prediction (NWP) modeling (Palmer, 2007; Buizza, 2005) and in climate modeling (Broccoli, 2003; Murphy, 2004; Staniforth, 2005). The traditional ensemble approach widely used in NWP is based on introducing perturbations into initial conditions because NWP forecasts (specifically, for short- to medium-term or 1 to 10 day weather predictions) are the initial condition problems. Hereafter, we will call this kind of ensembles the perturbed initial condition ensemble (PICE).

It was also found that, for both the NWP and especially for climate applications, the spread of PICE forecasts is insufficient to systematically capture the natural climate and weather variability (both spatial and temporal). Another approach to ensemble modeling based on perturbing model physics developed and implemented for ensemble forecast systems (Buizza, 2005; Buizza, 1999). Climate simulations which are from months to decades (and sometimes centuries) long are not initial condition but rather boundary condition problems. In other words, climate simulations “forget” the initial conditions after two-three weeks of model integrations, and are driven by the right hand side (r.h.s.) or model physics forcing.. For this kind of problems, an ensemble approach based on perturbation of model physics (or perturbation of model forcing) seems to be appropriate. The perturbed physics ensembles are expected to be more suitable for climate model simulations and projections (Staniforth, 2005).

In this chapter we investigate different possibilities of using the neural network (NN) emulation technique, introduced in the earlier chapters, in combination with ensemble approaches. We discuss two types of perturbed physics ensembles: a long term perturbed physics ensemble (PPE) and a short-term perturbed physics ensemble (STPPE).

We also show that the NN emulation technique can be efficiently used to create PPE and STPPE. We demonstrate that all three aforementioned types of ensembles (PICE, PPE, and STPPE) can significantly benefit, in terms of their numerical performance, from using accurate NN emulations of model physics; however, STPPE becomes especially efficient (orders of magnitude faster than PICE and PPE) when the NN technique is used to produce the ensemble of perturbed realizations of model physics.

7.1 Ensemble Approaches for NWP and Climate Simulations

General circulation models (GCMs) used for numerical climate simulations and NWP are complex nonlinear systems composed of many elements: initial conditions, $\psi(0)$,

model dynamics, $D(\psi, t)$, model physics, $P(\psi, t) = \sum_k p_k(\psi, t)$ (p_k are

parameterizations of physical processes), etc. Here ψ is the atmospheric state vector.

Each of these elements as well as boundary conditions can be considered as a specific component that has its own internal (natural) uncertainty. Each of these components may be perturbed within its natural uncertainty to produce an ensemble of model realizations. Each of these ensemble realizations produces a prediction, which constitutes an ensemble member.

Formally, an ensemble forecast system may be represented as a set of numerical integrations,

$$\psi_j(T) = \psi_j(0) + \int_0^T [P_j(\psi_j, t) + D(\psi_j, t)] dt \quad (11)$$

where $j = 1, \dots, N$ is the number of an ensemble member. All ensemble members represent rather close but different simulated states. The ensemble approach allows for integrating the specific information contained in the individual ensemble members into an ensemble mean that has more information about or represents the predicted climate or weather presumably better than each of the individual ensemble members.

where $j = 1, \dots, N$ is the number of an ensemble member. All ensemble members represent rather close but different simulated states. The ensemble approach allows for integrating the specific information contained in the individual ensemble members into an ensemble mean that has more information about or represents the predicted climate or weather presumably better than each of the individual ensemble members.

7.1.1 Ensembles with Perturbed Initial Conditions

Because NWP model integrations (specifically, for short- to medium-term or 1 to 10 day weather forecasts) are based on solving the initial condition problems, the traditional ensemble approach, PICE, widely used in NWP, consists of introducing perturbations into initial conditions; model physics is not perturbed and P_j are the same in eq. (11) for all ensemble members. Within this approach, each ensemble member run starts from uniquely perturbed initial condition $\psi_j(0)$. After running independently for some prescribed time T , the results of the individual ensemble runs

(i.e., the individual ensemble members) are compared with each other and with observations and eventually averaged to produce the ensemble mean (see Fig. 42).

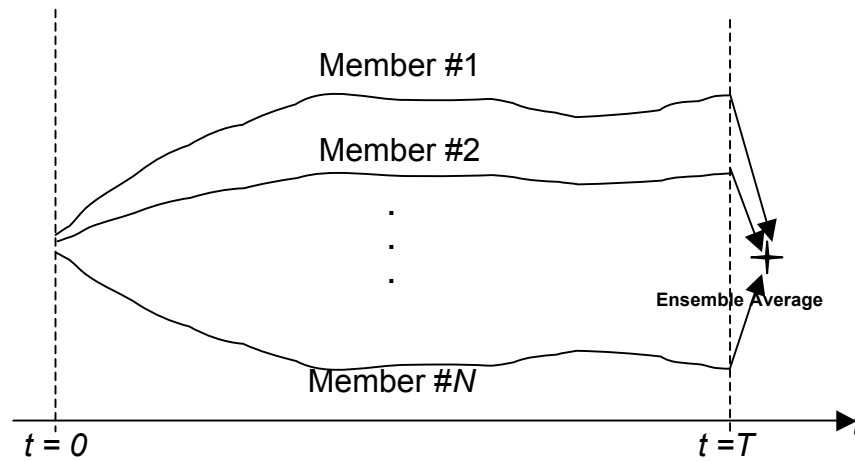


Fig. 42 The PICE and PPE scenario

Usually, the ensemble mean describes better an actual state of weather or climate at $t = T$ than an individual ensemble member. PICEs allow us to observe how small uncertainties in initial conditions develop over model integration time into significant/measurable differences in predicted atmospheric states. For PICEs, only initial perturbations are introduced into a deterministic NWP model integration. PICE proved to be an effective tool for NWP; however, it was also concluded that the spread of PICE forecast is often insufficient for providing systematic improvements of NWP (Palmer, 2007; Buizza, 2005).

7.1.2 Ensembles with Perturbed Physics

For the NWP and climate problems it was also shown that, a perturbed physics ensemble may provide a larger spread and better results (Buizza, 2005; Murphy, 2004); Stensrud, 2000). For example, ECMWF (European Center for Medium-term Weather Prediction) operational ensemble forecast system has been already augmented by including perturbed physics ensembles (Buizza, 2005; Buizza, 1999).

For climate models, which are not initial condition problems but rather boundary condition and r.h.s forcing problems, an ensemble generation approach based on perturbation of model physics (or perturbation of model forcing) is appears to be appropriate. Uncertainties in model physics that arise from the fact that the sub-grid effects are taken into account only approximately in model physics parameterizations, which include many uncertain parameters and approximations, have a different nature and spatial and temporal scales than uncertainties in initial conditions. In a sense, model physics parameterizations produce perturbations at each GCM grid point at each time step of its integration. The perturbed physics ensembles (PPE) are shown to be very effective for climate simulations and projections (Broccoli, 2003; Murphy, 2004; Staniforth, 2005; Stensrud, 2000; Kharin, 2000). Within this approach, each ensemble member uses a uniquely perturbed version of model physics P_j . PPE can also be used in combination with PICE as it is shown in eq. (11).

Several different approaches have been used for perturbing model physics:

- Model random errors associated with physical parameterizations are simulated by multiplying the total parameterized tendencies of P by a random number r_j sampled from a uniform distribution between 0.5 and 1.5 ($P_j = r_j \times P$).
- One or several model physics parameters controlling key physical characteristics of sub-grid scale atmospheric and surface processes can be perturbed at a time, within the scope of their natural uncertainty (Murphy, 2004; Staniforth, 2005).
- Different model physics parameterization schemes can be used to create various versions of perturbed model physics; the different versions are used in different ensemble members (Stensrud, 2000).

In section 7.2 of this chapter, a new method of generating ensemble of perturbed model physics is introduced that uses NN emulations of model physics as a tool to create different realizations of model physics.

The traditional perturbed physics ensemble (PPE) follows the same scenario, as that of depicted in Fig. 42 for the PICE with perturbed initial conditions. A particular GCM ensemble member uses a particular version of the perturbed physics, P_j , throughout the entire GCM run, for a long integration time T . Thus, in PPE different versions of perturbed physics (different realizations of the sub-grid physics) are used for different ensemble members, and each ensemble member exists and evolves over the entire GCM integration period T that is much longer than a characteristic time scale of sub-grid physical processes.

7.1.3 Short Term Ensembles with Perturbed Physics

Using the perturbed physics approach for generating ensembles offers an opportunity to introduce an alternative ensemble approach, namely a new type of ensemble – a short term perturbed physics ensemble (STPPE) that is not possible in the framework of the traditional PICE approach. In the STPPE mode, the ensemble of different realizations or perturbed model physics versions is introduced for a short time interval comparable with the time scales of the sub-grid processes, namely during one time step (or for some parameterizations for a few to several time steps) of the model integration. Symbolically, STPPE can be written as,

$$\psi(T) = \psi(0) + \int_0^T [\sum_{j=1}^N P_j(\psi, t) + D(\psi, t)] dt \quad (12)$$

At each time step, an ensemble of different realizations of model physics is generated and averaged. The ensemble average is used to integrate the model for producing the next time step. The STPPE scenario is shown in Fig. 43.

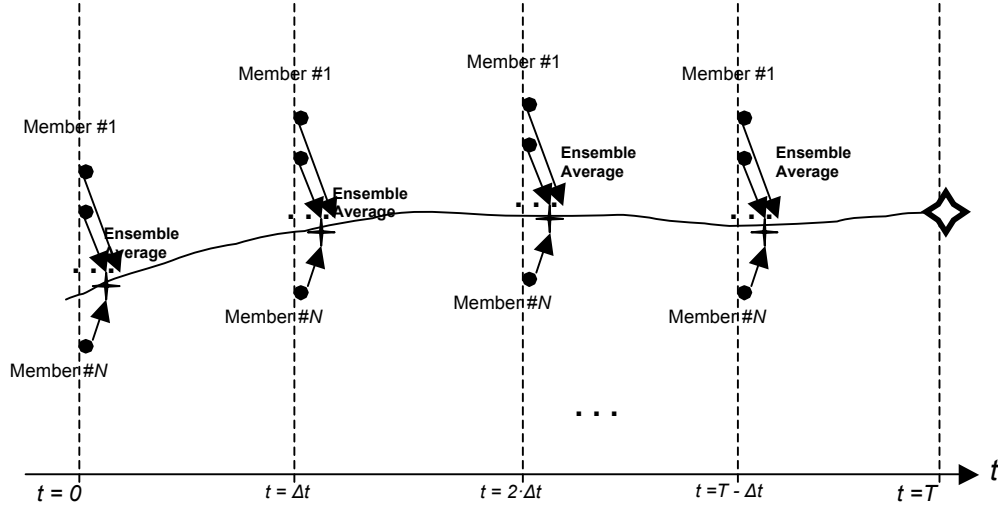


Fig. 43 The STPPE scenario.

The major differences between a PICE or PPE approaches (Fig. 42) and STPPE (Fig. 43) are as follows:

- PICE and PPE consist of N independent model runs; STPPE consists of a single model run.
- In the PICE and PPE approaches, the ensemble averages for climate or weather fields are calculated at the end of all N model integrations, by averaging climate or weather fields for all individual ensemble member runs; within STPPE, the ensemble average is calculated at each integration time step, Δt , for the outputs of the ensemble members composed of perturbed versions of model physics components. The weather or climate fields obtained at the end of model integration are the results of this single STPPE run. There is no additional averaging of weather or climate fields in this approach.

- SLPPE may be significantly faster than PICE or PPE; if calculations of a perturbed version (or a component) of model physics take about $\tau = \frac{1}{m}T$, where $1/m < 1$ is a fraction of T required for calculation of the model physics (or a particular component/parameterization of model physics that is perturbed), and T is the total time required for integration of one PICE member, so that the STPPE run takes time

$$T_{STPPE} = \left[\left(1 - \frac{1}{m}\right) + \frac{N}{m} \right] \cdot T \quad (13)$$

whereas PICE or PPE runs take a longer time

$$T_{PICE} = N \cdot T = N \cdot \left[\left(1 - \frac{1}{m}\right) + \frac{1}{m} \right] \cdot T \quad (14)$$

The major technical difficulty in realization of all three ensemble approaches discussed above (PICE, PPE, and STPPE) is their time consumption. Both PICE and PPE cost N (N – is the number of ensemble members) times more than a single model run; that is $N \cdot T$, where T is the time required for one GCM run. STPPE needs significantly less time because only model physics is calculated N times. For example, if the calculation of model physics takes 50% of the total model calculation time, STPPE will be about 2 times faster than PICE or PPE runs, assuming that the number of ensemble members is the same, N . If model physics calculation time is reduced the STPPE becomes even more computationally efficient. In the next section, we show that STPPE becomes very efficient (orders of magnitude faster than PICE and PPE) when the neural network (NN) technique is used to produce the ensemble of perturbed realizations of model physics.

7.2 Neural Network Ensembles with Perturbed Physics

If we produce N perturbed versions of model physics adding some perturbations to the entire model physics or to one of its components (parameterizations), we can use these N perturbed versions to create PPE members following a traditional scenario of PICE (Fig. 42). These N versions can also be used as members of STPPE following an alternative scenario presented in Fig. 43.

The j^{th} perturbed version of the unperturbed model physics, P , can be written as,

$$P_j = P_j^{NN} = P + \varepsilon_j \quad (15)$$

where P_j^{NN} is a NN emulation number j of the original model physics, P , and ε_j is an emulation error for the NN emulation number j . As we have shown in the previous chapters, ε_j can be controlled and changed significantly by varying k (the number of hidden neurons). Not only the value but also the statistical properties of ε_j can be controlled. For example, the systematic components of the emulation errors (biases) can be made negligible (therefore, ε_j are purely random in this case). Thus, ε_j can be made of the order of magnitude of a natural uncertainty of the model physics (or of a particular parameterization) due to an unaccounted variability of sub-grid processes.

Using NN emulations will speed up calculations of all three kinds (PICE, PPE, and STPPE) of ensembles. One PICE or PPE run with N ensemble members using N different NN emulations, each of which is n times faster than the original model physics, as perturbed versions of model physics will take time,

$$T_{PPE}^{NN} = N \cdot \left[\left(1 - \frac{1}{m}\right) + \frac{1}{m \cdot n} \right] \cdot T \quad (16)$$

Thus, in the case of NCAR CAM, where $m \approx 3/2$ to 2 and $n \approx 10$ to 100, using NNs for PICE or LTPPE will speed up its calculations about two to three times.

The speed-up of calculations of PICE and PPE due to the use of NN emulations of model physics is significant. However, the speed-up is even much more significant for SLPPE. When we use N NN emulations each of which are n times faster than the original model physics, the STPPE run takes time

$$T_{STPPE}^{NN} = \left[\left(1 - \frac{1}{m}\right) + \frac{N}{m \cdot n} \right] \cdot T \quad (17)$$

It means that STPPE with $N = n$ ensemble members (N different NN emulations of model physics taken as ensemble members) can be run as fast as a single ensemble member of PICE or PPE (see eq. (14)).

Here, the legitimate question to ask is how efficient is the STPPE approach. In other words, does it improve the accuracy of climate simulations to a degree at least comparable with improvements provided by the PICE and LTPPE approaches? This point is discussed in the next section.

7.3 Comparisons of Different Ensembles using Perturbed NCAR CAM LWR

For validation of our experiments, we use the NCAR CAM run using the original model physics and the original NCAR CAM initial conditions as a control against which all ensemble members and ensembles means for all three considered types of the ensembles are validated. In other terms, the climate obtained from the 15-year run of NCAR CAM with the original model physics (including the original LWR parameterization) and original initial conditions is used below as a “true” or control climate. All ensemble members and ensemble means for different ensembles (PICE,

PPE and STPPE) are compared with these synthetic “observations”. Then to create an ensemble of perturbed physics, we emulated the original LWR parameterization (Collins, 2002) with six different NNs which approximate the original LWR parameterization with different limited approximation errors.

The perturbed LWR parameterizations can be written as,

$$LWR_j^{NN} = LWR + \varepsilon_j \quad (18)$$

where LWR is the original NCAR CAM LWR, LWR_j^{NN} is a NN emulation number j of the original NCAR CAM LWR, and ε_j is an emulation error for the NN emulation number j . Thus, the model physics that includes LWR NN emulation, LWR_j^{NN} , can be considered as perturbed versions of model physics, P_j .

There are many different approaches to creating different NN emulations of the same original parameterization (or different perturbed versions of model physics). We have selected a sufficiently diverse group of six NN emulations mixing two different approaches to create a set of NN emulations (perturbed versions of model physics). Five of these six NN emulations or realizations of LWR have the same architecture that is the same number of neurons ($k = 150$); however, these NNs are different because different initializations for the NN weights have been used to start the NN training; the NNs have different weights (coefficients) and give close but different approximations of LWR (i.e. realizations of LWR). The sixth NN emulation has a different architecture ($k = 90$ neurons). In terms of the accuracy of the approximation, there is a significant spread between the members of this NN emulation set. The approximation rms errors vary from 0.28 to 0.40 K/day for the six NN emulations. It means that by using NN emulations instead of the original LWR

parameterization, we introduced on average such a level of perturbation into the LWR model physics.

The distribution of approximation errors (perturbations) is shown in Fig. 44. It is obviously not normal. For the normal distribution with the same mean value and standard deviation, the perturbation values would be very limited; however, because the distribution of ε_j is not normal, there is a small but finite probability of larger perturbations. If we compare these perturbations with mean value, μ , and standard deviation, σ , of LWR itself ($\mu = -1.4$ K/day and $\sigma = 1.9$ K/day), we will see that the majority of perturbations belong to the interval $\mu \pm \sigma$; however, a very small amount of perturbations reaches the magnitude of about $\mu \pm 3\sigma$. Such a distribution of perturbations is in a good agreement with the fact that the parameterizations of model physics on average describe the parameterized processes well enough and the level of errors introduced due to parameterization of sub-grid effects is rather moderate; however, in some cases (e.g. for rare or extreme events) the errors may be significant. In the case of NCAR CAM LWR the NN emulations are about $n = 100$ times faster than the original LWR parameterization. Since the calculation of the original NCAR CAM LWR takes about 30% of the total model integration time T ($m = 3$ in eqs. (13), (14), (16), and (17)), using LWR NN emulations in PICE and PPE results in speeding up the model calculations by about 30%, by reducing the time required for calculating NCAR CAM LWR by $n \cdot m$ times. For SHPPE the use of NN emulations provides a much more significant speed up of calculations. Just as an example, a STPPE with $N = 100$ ensemble members (eq. (16)) runs as fast as a single ensemble member of PICE or PPE (eq. (14)).

Also, to run a PICE that is used mostly for comprehensive comparison purposes, we created six randomly perturbed initial conditions members by perturbing original initial conditions for temperature fields used for the control run. Then we performed a PICE run (see Fig. 42); six climate simulations have been run with NCAR CAM for 15 years, each with one of these six perturbed initial conditions.

Next we performed a PPE run (see Fig. 42); six climate simulations have been run with NCAR CAM for 15 years, each with one of the aforementioned six NN emulations (also used as the perturbed versions of model physics for STPPE).

The results (climate fields and diagnostics) of each simulation (an ensemble member) were compared with the control climate run of NCAR CAM performed with the original LWR and original initial conditions. The climate simulation errors - systematic (bias), rmse, maximum (an extreme positive outlier), and minimum (an extreme negative outlier) - have been calculated for simulated prognostic and diagnostic fields for each ensemble member vs. the control climate. These errors are shown by diamonds (for PICE) and crosses (for PPE) in Figs. 45-50. Then the PICE and PPE averages were calculated (shown by thick large diamonds and thick large crosses respectively in Figs. 45-50).

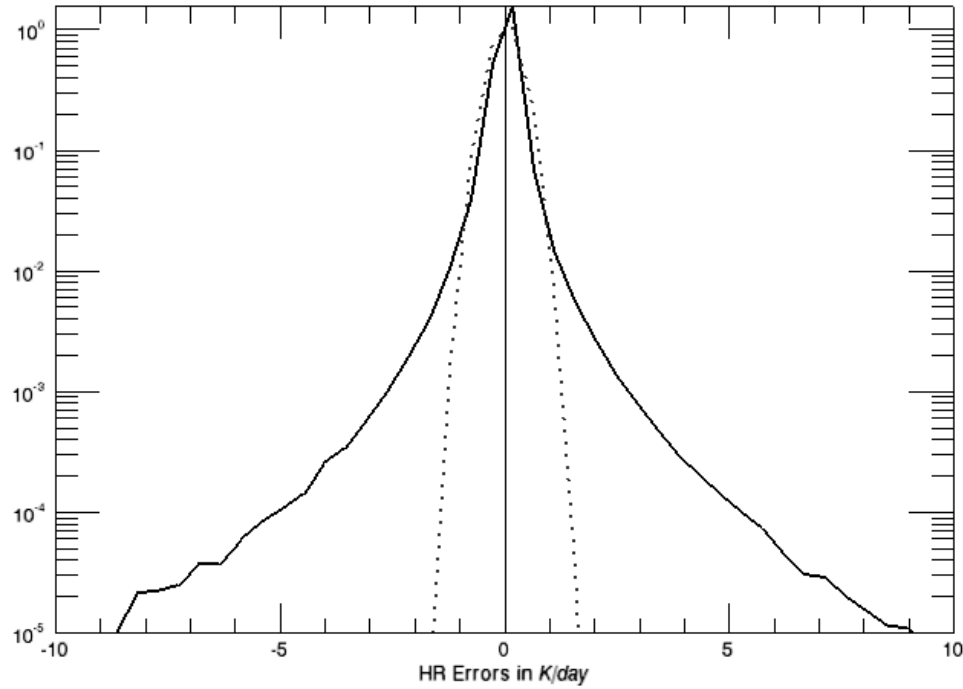
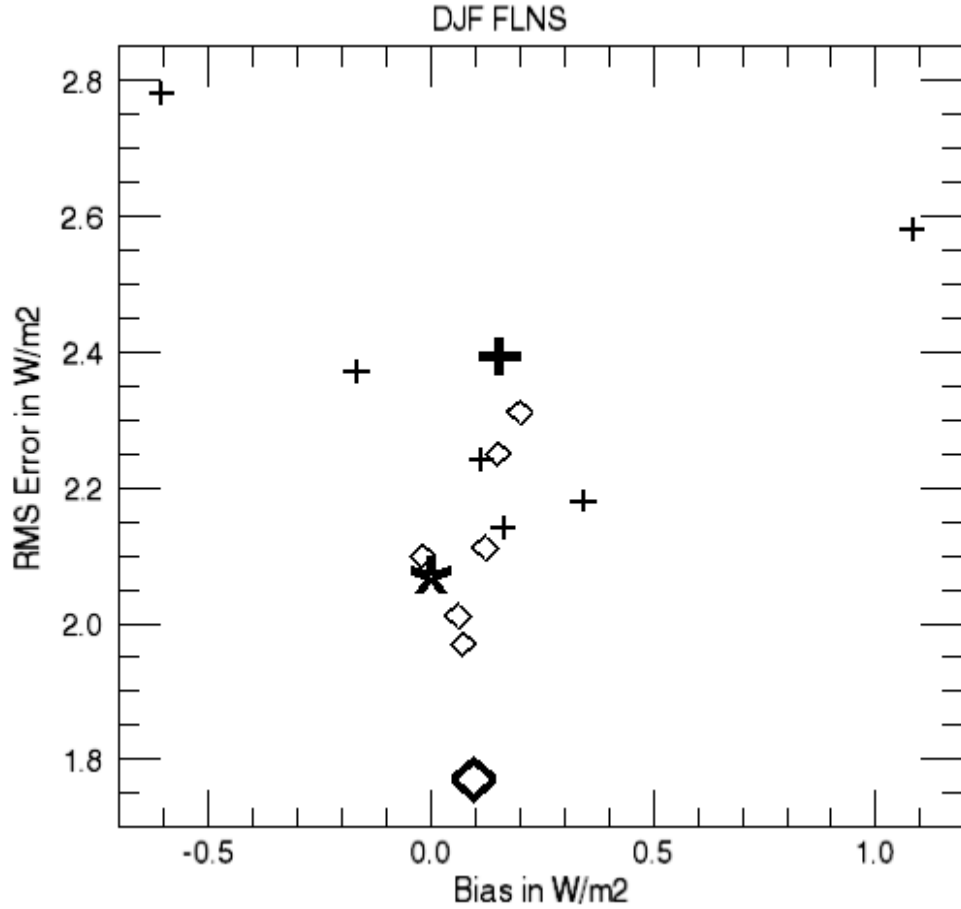


Fig. 44. Probability density function for ϵ_j . Mean $\epsilon_j = 3 \cdot 10^{-4}$ K/day and the standard deviation of ϵ_j is 0.35 K/day. The dashed line shows a normal distribution with the same mean and standard deviation for comparison.

Next, one 15-year STPPE climate run has been performed. For this run, six aforementioned NN emulations were applied and the LWR outputs are calculated as the mean of these six NN emulation outputs, at each time step and at each grid point throughout the entire model integration. The errors for mean climate fields and diagnostics of this model simulation (the SLPPE mean) are calculated against the control run and are shown by thick large stars in Figs. 45-50.



minimum and maximum errors for the relevant 15-year NCAR CAM climate simulations for all ensemble members and ensemble means.

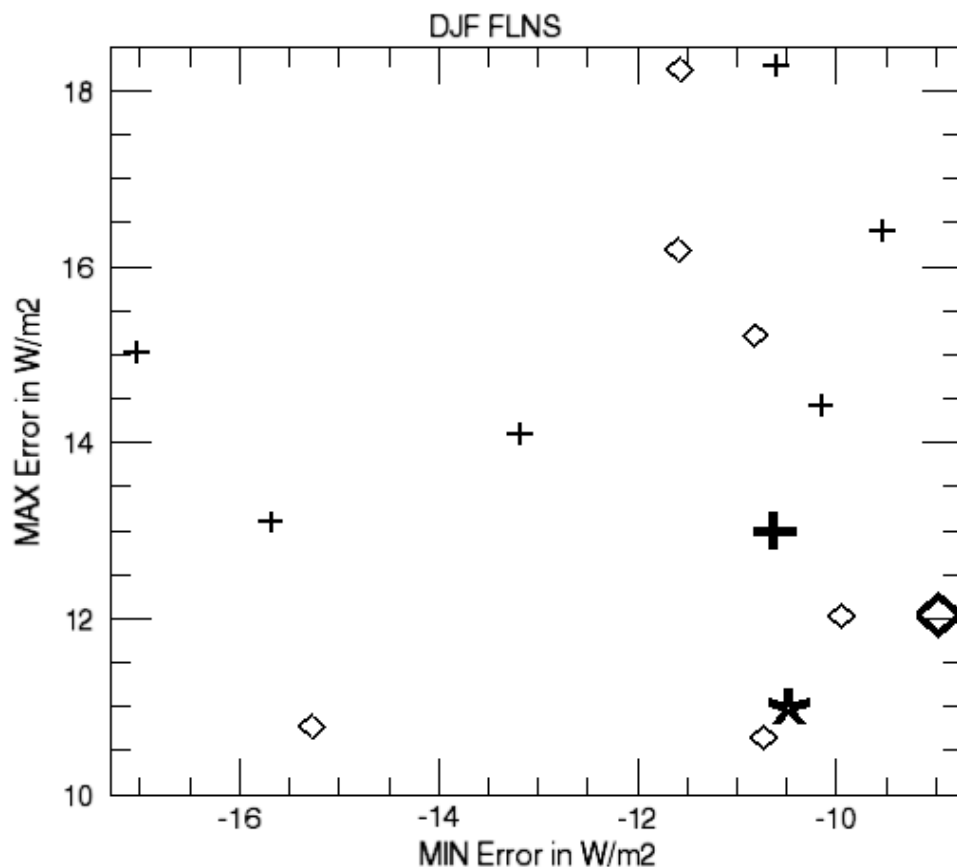


Fig. 46. Min and max errors, in W/m^2 , for the mean winter DJF FLNS. Symbols as in Fig. 45.

Figs. 45 and 46 show the winter DJF (December through February) surface net LWR flux (FLNS) errors, in W/m^2 , as deviations from the control climate. It is noteworthy that global min and max errors shown in Figs. 46, 48 and 50 are extreme outliers obtained for the entire 15-year NCAR CAM integrations. Similarly, Figs. 47 and 48 show another major radiation diagnostic, the DJF TOM (top of the model) net LWR flux (FLNT), a.k.a. the outgoing long-wave radiation (OLR), and errors as deviations

from the control climate (in W/m^2). Figs. 47 and 50 show DJF PSL errors as deviations from the control (in hPa).

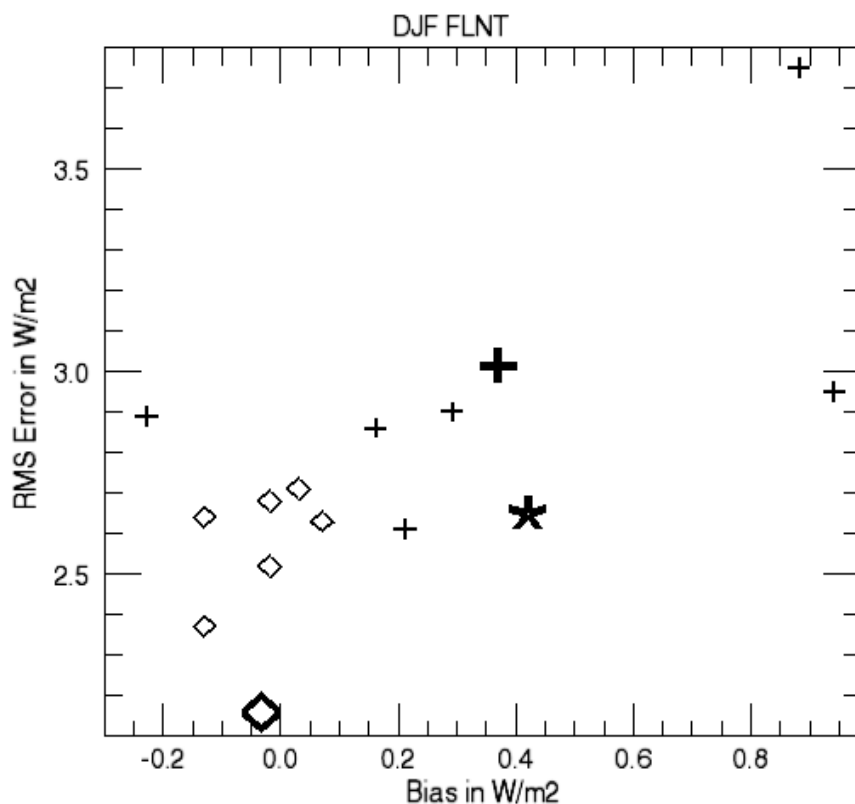


Fig. 47. Mean error (bias) and RMSE, in W/m^2 , for the winter DJF (December through February) top of the model net LWR flux (FLNT). Symbols as in Fig. 45.

The results presented in Figs. 45-50 clearly demonstrate that all three considered ensemble approaches show the similar overall improvement of the accuracy (or reduction of uncertainty) of climate simulations for their corresponding ensemble means. They also show that PPE generates an overall larger spread of the ensemble members than PICE with random perturbation of initial conditions. Insufficient spreads are typical for PICE; however in our case the spreads are even smaller due to a simplified technique used for introducing random perturbation, which under-

represents the natural uncertainty of initial conditions. It leads to limitations of the comparisons of the PPE and PICE spreads. More representative perturbations of initial conditions for PICE will be used for our future ensemble experiments.

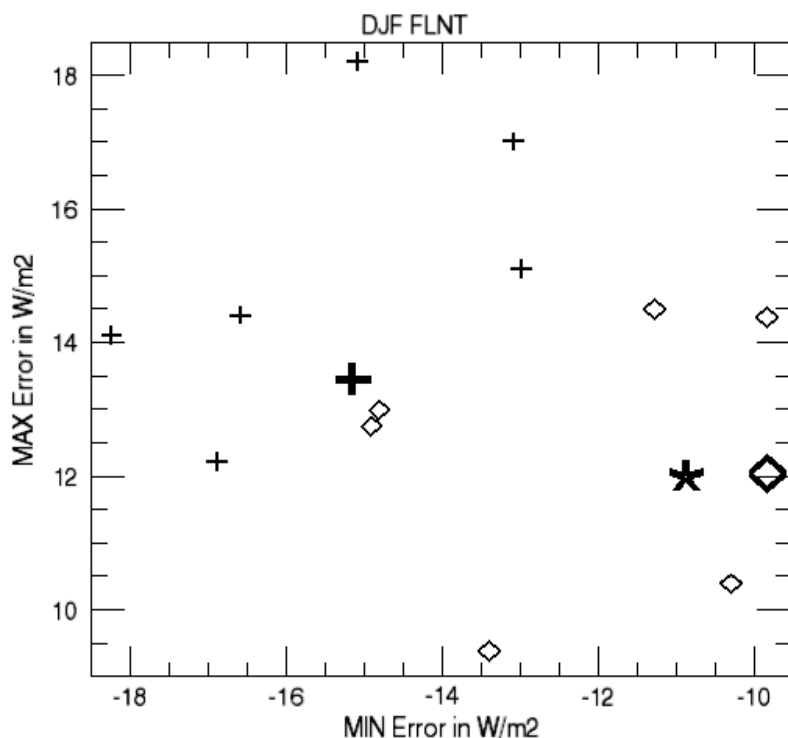


Fig. 48. Min and max errors, in W/m^2 , for FLNT climate. Symbols as in Fig. 45.

More specifically, the PPE and STPPE biases for the radiation diagnostics, FLNS and FLNT, (Figs. 45 and 47), are small and close to each other whereas the STPPE RMSE is smaller than that of PPE. For PSL (Fig. 49), the STPPE and PICE show close near-zero biases and close RMSEs; the PPE bias is only slightly larger but its RMSE is smaller.

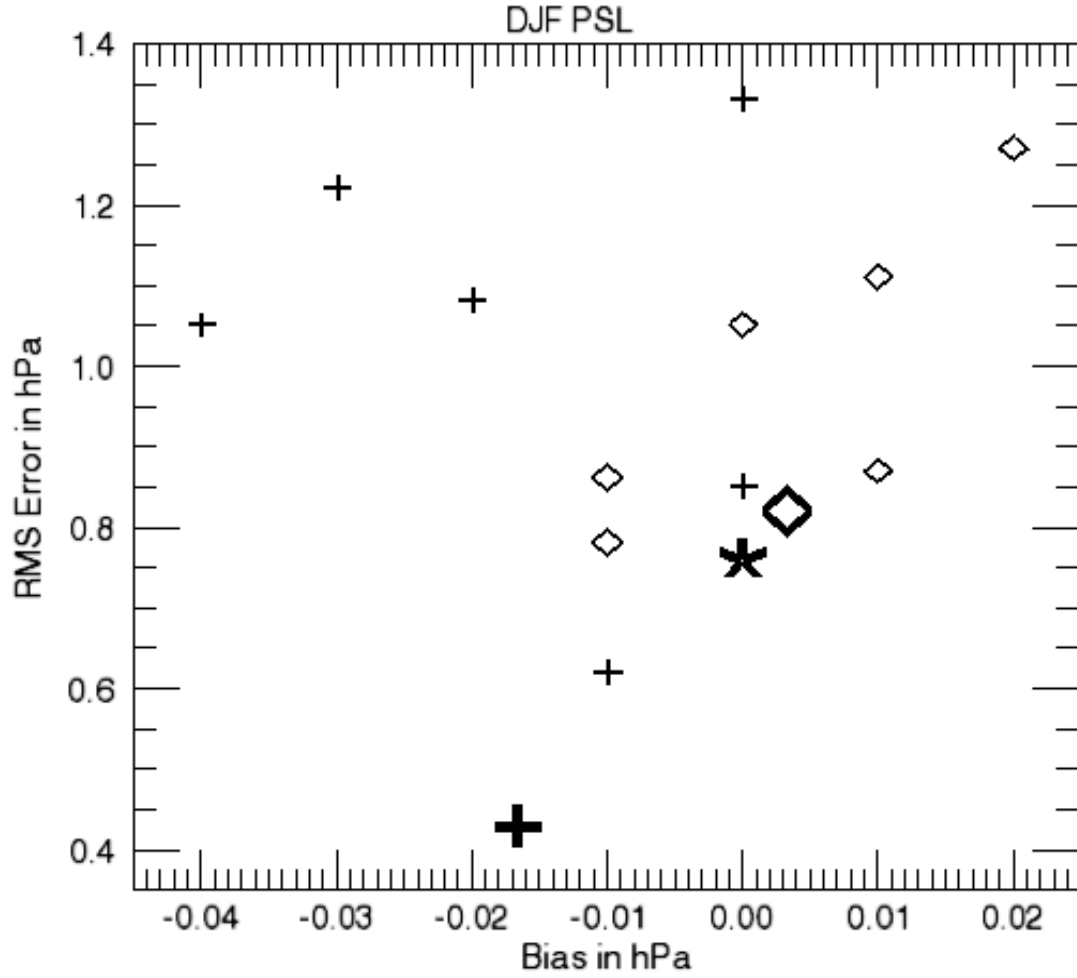


Fig. 49. Mean error (bias) and RMSE, in hPa, for the pressure at the surface level (PSL). Symbols as in Fig. 45.

The PPE and STPPE minimum errors for FLNS are close to each other but the maximum error for STPPE is smaller than that of LTPPE (Fig. 47). The PICE minimum error is larger than that of STPPE but smaller than that of PPE. For FLNT, STPPE and PICE have close maximum and minimum errors, which are smaller than those of PPE (Fig. 49). The STPPE and PICE minimum errors for PSL are close and

smaller than that of PPE (Fig. 50) whereas the STPPE and PPE maximum errors are close to each other and larger than that of PICE.

Overall, biases for all three types of ensembles are small and RMSEs are limited. STPPE RMSEs are slightly smaller than those of PPE.

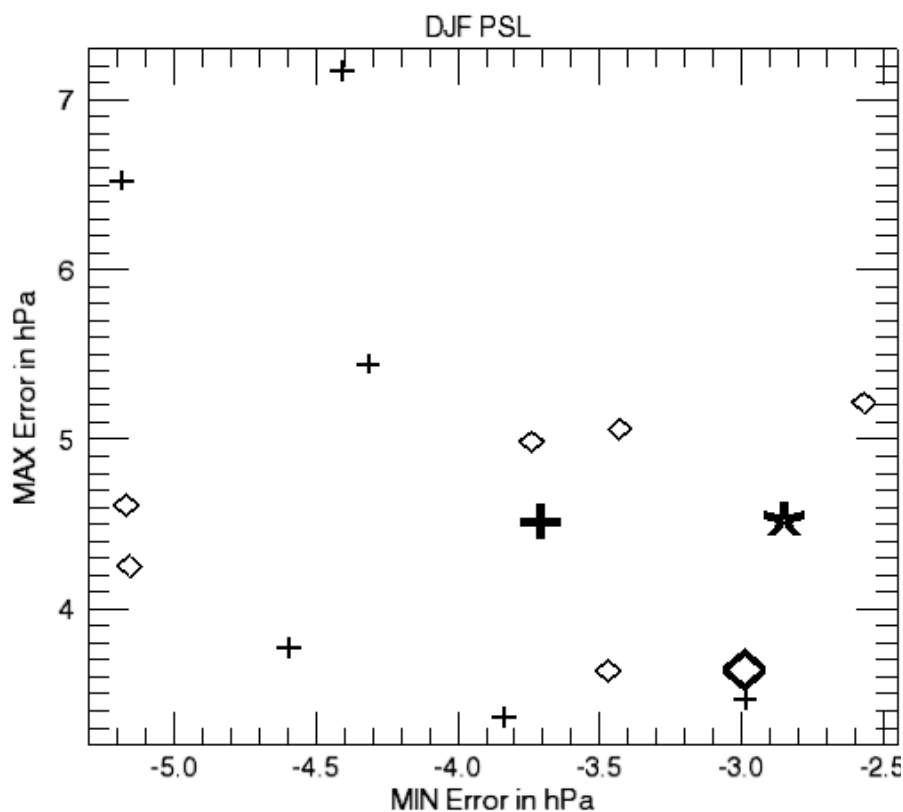


Fig. 50. Min and max errors, in hPa, for PSL climate. Symbols as in Fig. 45.

7.4 Discussion

The results presented here show that all three ensemble approaches, the perturbed initial conditions ensemble (PICE), the traditional perturbed physics ensemble (PPE), and STPPE, produce similar results: the use of any of these ensembles for climate simulation reduces significantly the systematic error (bias); it also reduces the random

error making it close to that of the best individual ensemble member. The same is true for the extreme (min and max) errors.

All three considered ensembles demonstrate similar improvements of the climate simulation accuracy. Using NN emulations of model physics significantly improve the computational performance of any of investigated ensemble techniques. However, it is important to emphasize that STPPE is significantly faster than PICE and PPE. It is $2N$ times (12 times for the case of $N = 6$ ensemble members considered in our study) faster than PICE and N times (6 times in our study) faster than PPE. Also, our results indicate that PPE and STPPE using NN perturbed physics provides a significantly larger spread of ensemble members than PICE with randomly perturbed initial conditions.

This study is actually a pilot or proof of concept study that introduces and preliminary evaluates NNs as a tool for perturbing model physics and for using it in perturbed model physics ensembles. This study also introduces STPPE as a new kind of the ensemble approach. Some additional issues should be (and will be) investigated to obtain a better understanding of advantages and limitations of this approach:

- In this work we evaluated aforementioned ensemble techniques using the basic statistical metrics like bias, rmse, min and max errors. Various statistical metrics specifically designed for evaluation of ensemble prediction systems (EPS) should be applied to perform enhanced quantitative comparison between PICE, LTPPE, and STPPE.

- It was shown that the perturbation ε_j introduced by the NN emulation technique can be controlled and changed not only in terms of its value but also in terms of its statistical properties. A broader sample of NN emulations with a broader spread of error statistics should be considered and evaluated.
- In this study we used an unperturbed NCAR CAM run with the original parameterizations of physics as a control run or “synthetic observations”. Similar evaluation should be performed with real observations.
- A climate model, NCAR CAM, was used to evaluate aforementioned ensemble techniques in the climate simulation environment. Similar evaluation should be performed in the framework of a numerical weather prediction EPS to evaluate these techniques for NWP models.
- More realistic perturbation technique like those of used in Palmer (2007) or Buizza (2005) should be applied to create a better PICE with a more realistic and larger spread for a better comparison with LTPPE and STPPE.
- Some parts of the climate/weather numerical models like convection physics, or full model physics (containing boundary layer, land, and ice models), or model chemistry are not as well defined as the model radiation that we perturbed in this study; they introduce larger uncertainties in model calculations. These components may be even better candidates for using the NN emulation technique for producing their perturbed versions for creating model ensembles.

Chapter 8: Summary

Our work covered five application areas of Statistical Learning Techniques to atmospheric numerical modeling.

8.1 NN Emulations of Full Model Radiation: Climate Runs

We presented an approach based on a synergetic combination of deterministic modeling based on physical (first principle) equations and statistical learning (NN emulation) components within an atmospheric model. The statistical learning approach was used to develop highly accurate and fast NN emulations for model physics components. Here we presented a NN emulation of the full atmospheric radiation, i.e. for long- and short-wave radiation parameterizations used in numerical climate and weather prediction models.

This study has shown the practical possibility of using highly efficient NN emulations for the full model radiation block for decadal climate simulations in an uncoupled medium resolution atmospheric model driven by climatological SSTs (NCAR CAM) and a coupled high resolution climate model with prescribed time dependent CO₂ and aerosols (NCEP CFS). A very high accuracy and increased speed of NN emulations for the both NCAR CAM and NCEP CFS full radiation (LWR and SWR) has been achieved. The systematic errors introduced by NN emulations of full model radiation are negligible and do not accumulate during the decadal model simulation. The random errors of NN emulations are also small. Almost identical results have been obtained for the parallel multi-year climate runs. These results show the potential of

developing efficient NN emulations for model physics components and the entire model physics.

Because model vertical resolution determines the NN emulation architecture, i.e., the number of inputs and outputs, every time the vertical resolution of the model is changed (which is usually done quite rarely), the NN emulation needs to be retrained.

It is noteworthy that NN retraining can be done routinely and takes a limited time and effort once the practical framework for a specific model is developed.

8.2 NN Emulations of Full Model Radiation: Short- To Medium- Range Forecasts

We also studied the applicability of the NN emulation technique of the full model radiation to the short- to medium- range forecasting in a very high resolution atmospheric model (NCEP GFS). We found that the developed NN radiation is very accurate; the NN run closely follows the control run. The differences between NN and control runs increase slowly with the forecast time; however, in many cases NN run demonstrates slightly better results (higher anomaly correlation, lower bias and RMSE) at larger forecast times.

In addition we investigated level of robustness of the developed NN radiation with respect to changes in the model environment. It shows that the developed NN radiation (CFS NN radiation) survived the transplantation from an old version of coupled model (CFS) to the newest version of uncoupled GFS. It also survived about 5 years of constant model evolution resulted in many changes in other than radiation physics parts of the model. After all these changes it still produces reasonable results. This is a very important practical result, which shows that the NN radiation does not

require frequent updates and may work in the model for many years without retraining. Of course, when the original radiation parameterization or the vertical resolution of the model is changed, the NN radiation has to be retrained.

8.3 Application of Non-Parametric Learning Methods to Emulation of Model Components.

We conducted numerical experiments investigating the possibility of substituting physical parameterizations in global climate models with non-parametric emulations. The results are positive in the sense that they show that both nearest neighbor type methods and regression trees are in principle able to achieve statistical approximation quality on par with neural networks, even if trained with a relatively moderate amount of data. It has been demonstrated that the NCAR CAM with a tree-based LWR emulation gave results in good agreement with the calculation using the original parameterization, except in the polar regions, which could have been expected from the statistical properties of the approximation. The main obstacle for the practical use of non-parametric methods is less a mathematical one, but rather one of implementation. Non-parametric approximation methods are memory-based, i.e., they need to store all the training data permanently. This makes its use in a parallel environment more difficult than is the case for the relatively compact neural network representation. Of course, for huge, complex projects like climate simulation software, implementation issues are a major concern. Therefore, the ideas and results presented in the current paper can only be considered as preliminary step towards a new emulation paradigm.

8.4 Compound Parameterization of Full Radiation with a Quality Control of Larger Errors in NCAR CAM

Applying the NN emulation approach, which allows us to achieve such a significant speed-up with preservation of the accuracy and functional integrity of model physics, may create some challenges that can be resolved using the tremendous flexibility of statistical learning techniques and of the NN technique in particular. Because NN emulations are statistical approximations, there exists a small probability of larger approximation errors or outliers. The major reason for obtaining larger errors is high dimensionality n of the input space of the approximated mapping, which reaches several hundreds for NCEP CFS and may reach thousands for future models with significantly higher vertical resolution. It is difficult to sample uniformly a domain in such a high dimensional space. Far corners of the domain may remain underrepresented in the training set. During the NN run, if input vectors belonging to these underrepresented far corners of the domain are encountered, they may cause larger errors in the NN outputs.

To alleviate this problem a new improved NN emulation approach called a compound parameterization, which incorporates NN-based quality control techniques for controlling larger errors of NN emulations, has been developed. One design of a compound parameterization presented in the paper uses a special NN trained to predict errors in outputs of NN emulation of a climate model physics component. It is shown that the accurate representation of a model physics component using a compound parameterization with a quality control of larger errors is essential for successful climate simulations.

The CP approach can be considered as an engineering solution that does not investigate the problem (why on some rare occasions a NN emulation does not perform well) but bypasses it allowing to use this NN emulation safely in the essentially nonlinear and complex environment of a numerical model. If a second error NN can be trained to reliably predict errors of the NN emulation, then it looks like these errors can be investigated, explained, and eliminated by correction of the NN emulation itself. Theoretically speaking, this is correct. However, practically speaking, it is hardly possible. As we have mentioned before, the main reason for such larger errors to occur is our inability to generate a completely representative training set, that is to get represented each far corner of the domain of the approximated mapping. For modern climate and weather models this domain has dimensionality of order of 10^3 and higher. A systematic investigation of such an object is a formidable task that requires significant special efforts. Using CP allows us to flag and to bypass these questionable far corners of the domain leaving their investigation for the future research.

8.5 Ensemble of Neural Network Emulations for Climate Model Physics

We also introduced a new type of perturbed physics ensemble (STPPE) approach. STPPE uses the ensemble mean of different versions of perturbed physics. These versions are represented by different NN emulations used for calculating model physics (LWR in our case) at each time step and at every grid point throughout the entire model integration. The neural network emulation technique allows us: (1) to speed up significantly (up to 2 -3 times) ensemble model calculations by introducing

fast neural network emulations of model physics components; (2) to introduce consistent perturbations into model physics and develop its fast versions, and (3) to introduce a new short term perturbed physics ensemble for which computation time is comparable with the computation time that of needed just for a single model run.

Preliminary results presented here show that all three ensemble approaches, the perturbed initial conditions ensemble (PICE), the traditional perturbed physics ensemble (PPE), and STPPE, produce similar results: the use of any of these ensembles for climate simulation reduces significantly the systematic error (bias); it also reduces the random error making it close to that of the best individual ensemble member. The same is true for the extreme (min and max) errors.

The five application areas of Statistical Learning Techniques to atmospheric numerical modeling with hybrid models, i.e. comprised of deterministic and statistical learning components, presented in this dissertation have shown the advantages of the approach for application to model radiation and its potential for applications to other model physics. We believe that the concept of hybrid modeling deserves a closer attention of the modeling community.

Glossary of Acronyms

ACF: Autocorrelation Function

AGCM: Atmospheric GCM

CAM: Community Atmospheric Model

CART: Classification and Regression Tree

CARTC: Component Classification and Regression Tree

CDAS: Coordinated Data Analysis System

CFS: Climate Forecasting System

CP: Compound Parameterization

CTL: Control

DJF: December-January-February

ECMWF: European Center for Medium Range Weather Forecasting

ENM: Environmental Numerical Model

ENSO: El Niño-Southern Oscillation

GCM: General Circulation Model

GFS: Global Forecast System

HEM: Hybrid Environmental Model

HGCM: Hybrid General Circulation Model

JJA: June-July-August

LTPPE: Long-term Perturbed Physics Ensemble

LWR: Long Wave Radiation

MLP: Multilayer Perceptron

MLT: Machine Learning Technique

NCAR: National Center for Atmospheric Research

NCEP: National Centers for Environmental Prediction

NN: Neural Network

NWP: Numerical Weather Prediction

PICE: Perturbed Initial Conditions Ensemble

PRATE: Total Precipitation Rate

PRMSE: Profile RMSE

PPE: Perturbed Physics Ensemble

QC: Quality Control

RMSE: Root Mean Square Error

RFV: Vector Valued Random Forest

SLT: Statistical Learning Technique

SST: Sea Surface Temperature

STPPE: Short-term Perturbed Physics Ensemble

SWR: Short Wave Radiation

UKMO: United Kingdom's Meteorological Office

Bibliography

Aires, F., M. Schmitt, A. Chedin, and N. Scott (1999), The “weight smoothing” regularization of MLP for Jacobian stabilization, *IEEE Trans. Neural Networks*, 10, 1502–1510.

Atkinson, P. M., and A. R. L. Tatnall (1997), Neural networks in remote sensing—Introduction, *Int. J. Remote Sens.*, 18, 699 – 709.

Attali, J.-G., and G. Page’s (1997), Approximations of functions by a multilayer perception: A new approach, *Neural Networks*, 6, 1069 – 1081.

Beale, R., and T. Jackson (1990), *Neural Computing: An Introduction*, 240 pp., Adam Hilger, Bristol, U. K.

Belochitski A., P. Binev, R. DeVore, M. Fox-Rabinovitz, V. Krasnopolsky, and P. Lamby, 2011: Tree Approximation of the Long-Wave Radiation Parameterization in the NCAR CAM Global Climate Model, *Journal of Computational and Applied Mathematics*, 236, pp. 447-460

Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, 482 pp., Oxford Univ. Press, Oxford, U. K.

Bishop Ch. M., 2006: *Pattern Recognition and Machine Learning*. Springer, 738 pp.

Breiman, L, J. H. Friedman, R. A. Olshen, and C. J. Stone (1984): *Classification and Regression Trees*. Wadsworth

Breiman., L (2001): Random forests. *Machine Learning*, 45:5–32.

Broccoli, A, K. W. Dixon, T. L. Delworth, T. R. Knutson, and R. J. Stouffer (2003): Twentieth-century temperature and precipitation trends in ensemble climate simulations including natural and anthropogenic forcing, *J. Geophys. Res.*, 108 (D24), 4798-4811

Buizza, R., P. L. Houtekamer, Z. Toth, G. Pellerin, M. Wei, and Y. Zhu (2005): A Comparison of the ECMWF, MSC, and NCEP Global Ensemble Prediction Systems, *Monthly Weather Review*, 133, pp. 1076–1097

Buizza, R., M. Miller, and T. N. Palmer (1999): Stochastic representation of model uncertainties in the ECMWF Ensemble Prediction System, *Quarterly Journal of the Royal Meteorological Society*, 125, pp. 2887-2908

Chen, T., and H. Chen (1995a), Approximation capability to functions of several variables, nonlinear functionals and operators by radial basis function neural networks, *Neural Networks*, 6, 904–910.

Chen, T., and H. Chen (1995b), Universal approximation to non-linear operators by neural networks with arbitrary activation function and its application to dynamical systems, *Neural Networks*, 6, 911–917.

Cherkassky, V., and F. Mulier (1998): *Learning From Data*, 441 pp., John Wiley, Hoboken, N. J.

Chevallier, F., F. Chérut, N. A. Scott, and A. Chédin, 1998: A neural network approach for a fast and accurate computation of longwave radiative budget, *Journal of Applied Meteorology*, 37, 1385-1397.

_____, J.-J. Morcrette, F. Chérut, and N. A. Scott, 2000: Use of a neural-network-based longwave radiative transfer scheme in the EMCWF

atmospheric model, Quarterly Journal of Royal Meteorological Society, 126, 761-776.

Chou, M.-D., and M. Suarez, 1999: A solar radiation parameterization for atmospheric studies. NASA/TM-1999-104606, Vol. 15, 40pp.

Clough, S.A., M.W. Shephard, E.J. Mlawer, J.S. Delamere, M.J. Iacono, K.Cady-Pereira, S. Boukabara, and P.D. Brown, 2005: Atmospheric radiative transfer modeling: A summary of the AER codes, J. Quant. Spectrosc. Radiat. Transfer, 91, 233-244, doi:10.1016/j.jqsrt.2004.05.058.

Collins, W.D., 2001: Parameterization of generalized cloud overlap for radiative calculations in general circulation models, J. Atmos. Sci., 58, 3224-3242

_____, J.K. Hackney, and D.P. Edwards, 2002: An updated parameterization for infrared emission and absorption by water vapor in the National Center for Atmospheric Research Community Atmosphere Model, J. Geophys. Res., 107 (D22), 1-20.

Cote, J., S. Gravel, A. Me'thot, A. Patoine, M. Roch, and A. Staniforth, 1998a: The operational CMC-MRB global environmental multiscale (GEM) model. Part I: Design considerations and formulation, Mon. Weather Rev., 126, 1373– 1395.

Cote, J., J.-G. Desmarais, S. Gravel, A. Me'thot, A. Patoine, M. Roch, and A. Staniforth, 1998b: The operational CMC-MRB global environmental multiscale (GEM) model. Part II: Mesoscale results, Mon. Weather Rev., 126, 1397–1418.

Cybenko, G. (1989), Approximation by superposition of sigmoidal functions, Math. Control Signals Syst., 2, 303–314.

Dee, D., Rukhovets, L., Todling, R., da Silva, A., & Larson, J. (2001). An adaptive buddy check for observational quality control. *Q.J.R.Meteorol.Soc.* 127, 2451-2471.

DeVore, R. A. (1998), Nonlinear approximation, *Acta Numer.*, 8, 51 – 150.

DeVore, R., K. Oskolkov, and P. Petrushev (1997): Approximation by feed-forward neural networks. *Annals of Numerical Mathematics*, 4:261–287.

Funahashi, K. (1989), On the approximate realization of continuous mappings by neural networks, *Neural Networks*, 2, 183– 192.

Gandin, L.S. (1988). Complex quality control of meteorological observations. *Mon. Wea. Rev.*, 116, 1137-1156,

Gardner, M. W., and S. R. Dorling (1998), Artificial neural networks (the multilayer perceptron): A review of applications in the atmospheric sciences, *Atmos. Environ.*, 32, 2627–2636.

Hansen, L. K., and P. Salamon (1990), Neural network ensembles, *IEEE Trans. Pattern Anal. Mach. Intel.*, 12, 993–1001.

Hashem, S. (1997), Optimal linear combination of neural networks, *Neural Networks*, 10, 599–614.

Haykin, S. (1994), *Neural Networks: A Comprehensive Foundation*, 696 pp., Macmillan, New York.

Hou, Y.-T., S. Moorthi, and K.A. Campana, 2002: Parameterization of solar radiation transfer in the NCEP Models. *NCEP/EMC Tech. Mem.* 441, 46 pp.

Hornik, K. (1991), Approximation capabilities of multilayer feed-forward network, *Neural Networks*, 4, 251–257.

Hornik, K., M. Stinchcombe, and H. White (1990), Universal approximation of an unknown mapping and its derivatives using multilayer feed-forward networks, *Neural Networks*, 3, 551–560.

Hornik, K. (1989): . Multilayer feed-forward networks are universal approximators. *Neural Networks*, 2:359–366,

Hopfield, J. J. (1982), Neural networks and physical systems with emergent collective computational ability, *Proc. Natl. Acad. Sci. U. S. A.*, 79, 2554–2558.

Hsieh, W. W. (2001), Nonlinear principle component analysis by neural networks, *Tellus, Ser. A*, 53, 599–615.

Hsieh, W. W. (2004), Nonlinear multivariate and time series analysis by neural network methods, *Rev. Geophys.*, 42, RG1003, doi:10.1029/2002RG000112.

Hsieh, W. W., and B. Tang (1998), Applying neural network models to prediction and data analysis in meteorology and oceanography, *Bull. Am. Meteorol. Soc.*, 79, 1855–1870.

Iacono, M. J., E. J. Mlawer, S. A. Clough, and J.-J. Morcrette, 2000: Impact of an improved longwave radiation model, RRTM, on the energy budget and thermodynamic properties of the NCAR community climate model, CCM3, *J. Geophys. Res.*, 105(D11), 14,873–14,890.

Janiskova', M., J.-F. Mahfouf, J.-J. Morcrette, & F. Chevallier, 2002: Linearized radiation and cloud schemes in the ECMWF model: development and evaluation. *Quarterly Journal of the Royal Meteorological Society*, 128, 1505–1528.

Kalnay E, et al., 1996: The NCEP/NCAR 40-Year Reanalysis Project. *Bull. Amer. Meteorol. Soc.*, 77: 437–472.

Kistler R , et al., 2001: The NCEP-NCAR 50-Year Reanalysis, Bull. Am. Meteorol. Soc., 82, 247–268.

Kharin, V, F. W. Zwiers, 2000: Changes in the Extremes in an Ensemble of Transient Climate Simulations with a Coupled Atmosphere-Ocean GCM, J. Climate, 13, pp.3760-3788

Kohonen, T. (1982), Self-organizing formation of topologically correct feature maps, Biol. Cybern., 43, 59–69.

Krasnopolsky, V.M., 1997: A neural network forward model for direct assimilation of SSM/I brightness temperatures into atmospheric models, in Research Activities in Atmospheric and Oceanic Modeling, CAS/JSC Working Group on Numerical Experimentation, Report No. 25, WMO/TD-No. 792, 1.29-1.30

_____, and F. Chevallier (2003), Some neural network applications in environmental sciences. Part II: Advancing computational efficiency of environmental numerical models, Neural Networks, 16, 335–348.

_____, 2007a: Neural network emulations for complex multidimensional geophysical mappings: applications of neural network techniques to atmospheric and oceanic satellite retrievals and numerical modeling, Reviews of Geophysics, 45, RG3009, doi:10.1029/2006RG000200.

_____, 2007b: Reducing uncertainties in neural network Jacobians and improving accuracy of neural network emulations with NN ensemble approaches”, Neural Networks, 20, pp. 454–461

_____, and M.S. Fox-Rabinovitz, 2006a: A new synergetic paradigm in environmental numerical modeling: hybrid models combining

deterministic and machine learning components, *Ecological Modelling*, 191, pp. 5–18.

_____, and M.S. Fox-Rabinovitz, 2006b: Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction, *Neural Networks* 19, 122–134.

_____, M.S. Fox-Rabinovitz, and D.V. Chalikov, 2005a: “Fast and accurate neural network approximation of long wave radiation in a climate model”, *Monthly Weather Review*, vol. 133, No. 5, pp. 1370-1383.

_____, M. S. Fox-Rabinovitz, and D. V. Chalikov, 2005b, Reply, *Mon. Weather Rev.*, 133, 3724–3729.

_____, M.S. Fox-Rabinovitz, A. A. Belochitski, 2008a: "Decadal climate simulations using accurate and fast neural network emulation of full, long- and short wave, radiation.", *Monthly Weather Review*, 136, 3683–3695, doi: 10.1175/2008MWR2385.1.

_____, M.S. Fox-Rabinovitz, H.L. Tolman, and A. A. Belochitski, 2008b: neural network approach for robust and fast calculation of physical processes in numerical environmental models: compound parameterization with a quality control of larger errors, *Neural Networks*, doi:10.1016/j.neunet.2007.12.019

_____, V. M., and H. Schiller (2003), Some neural network applications in environmental sciences. Part I: Forward and inverse problems in satellite remote sensing, *Neural Networks*, 16, 321 – 334.

Lacis, A. A., V. Oinas, 1991: A description of the correlated k-distribution method for modeling non-gray gaseous absorption, thermal emission and multiple scattering in vertically inhomogeneous atmospheres. *J. Geophys. Res.* 96: 9027-9063.

Lippmann, R. P. (1989), Pattern classification using neural networks, *IEEE Commun. Mag.*, 27, 47–64.

Maclin, R., and J. Shavlik (1995), Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks, paper presented at 11th International Conference on Artificial Intelligence, Am. Assoc. on Artif. Intel., Detroit, Mich.

Manners, J., J.-C. Thelen, J. Petch, P. Hill & J.M. Edwards, 2009: Two fast radiative transfer methods to improve the temporal sampling of clouds in NWP and climate models, *Q. J. R. Meteorol. Soc.* 135: 457 – 468; doi: 10.1002/qj.385

Mlawer, E. J., S. J. Taubman, P. D. Brown, M. J. Iacono, and S. A. Clough 1997: Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave, *J. Geophys. Res.*, 102(D14), 16,663–16,682.

Moorthi, S., 2009: Private communications.

Moorthi S., Hua-Lu Pan and Peter Caplan, 2001: “Changes to the 2001 NCEP Operational MRF/AVN Global Analysis/Forecast System”, Technical Procedures Bulletin Series No. 484, National Weather Service, Office of Meteorology.

Morcrette, J.-J., G. Mozdzynski and M. Leutbecher, 2008: A reduced radiation grid for the ECMWF Integrated Forecasting System, *Monthly Weather Review*, 136, 4760-4772, doi: 10.1175/2008MWR2590.1

_____, P. Bechtold, A. Beljaars, A. Benedetti, A. Bonet, F. Doblas-Reyes, J. Hague, M. Hamrud, J. Haseler, J.W. Kaiser, M. Leutbecher, G. Mozdzynski, M. Razinger, D. Salmond, S. Serrar, M. Suttie, A. Tompkins, A. Untch and A. Weisheimer, 2007: Recent advances in radiation transfer parameterizations, ECMWF Technical Memorandum No. 539, October 18, 2007

Murphy, J., D.M. Sexton, D.N. Barnett, G.S. Jones, M.J. Webb, M. Collins, D.A. Staniforth: 2004, Quantification of modelling uncertainties in a large ensemble of climate change simulations, *Nature*, 430, 768 - 772

Naftaly, U., N. Intrator, and D. Horn (1997), Optimal ensemble averaging of neural networks, *Network Comput. Neural Syst.*, 8, 283 – 294.

Nguyen, D., and B. Widrow (1990), Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights, in *Proceedings of International Joint Conference on Neural Networks*, 1990, vol. 3, pp. 21–26, IEEE Press, Piscataway, N. J.

Nilsson, N. J. (1965), *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*, 137 pp., McGraw-Hill, New York.

Opitz, D., and R. Maclin (1999), Popular ensemble methods: An empirical study, *J. Artif. Intel. Res.*, 11, 169–198.

Palmer, et al. (2007): The Ensemble Prediction System – Recent and Ongoing Developments, ECMWF Technical Memorandum No. 540

Ripley, B. D. (1996), *Pattern Recognition and Neural Networks*, 403 pp., Cambridge Univ. Press, Cambridge, U. K.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986), Learning internal representations by error propagation, in *Parallel Distributed Processing*, vol. 1, edited by D. E. Rumelhart, J. L. McClelland, and P. R. Group, pp. 318–362, MIT Press, Cambridge, Mass.

Saha, S., S. Nadiga, C. Thiaw, J. Wang, W. Wang, Q. Zhang, H. M. van den Dool, H.-L. Pan, S. Moorthi, D. Behringer, D. Stokes, M. Pena, S. Lord, G. White, W. Ebisuzaki, P. Peng, P. Xie, 2006: The NCEP Climate Forecast System. *Journal of Climate*, Vol. 19, No. 15, pages 3483-3517.

Sato M., J.E. Hansen, M.P. McCormick, and J.B. Pollack (1993): Stratospheric aerosol optical depths, 1850-1990, *J. Geophys. Res.*, 98, 22987-22994.

Selfridge, O. G. (1958), Pandemonium: A paradigm for learning, in *Mechanization of Thought Processes*, pp. 513 – 526, Her Majesty's Stn. Off., London.

Shakhnarovich, G., T. Darrell, and P. Indyk, Eds. (2006): *Nearest-neighbor Methods in Learning and Vision: Theory and Practice*, MIT Press.

Sharkey, A. J. C. (1996), On combining artificial neural nets, *Connection Sci.*, 8, 299–313.

Staniforth, D, et al., (2005): Uncertainty in Predictions of the Climate Responses to Rising Levels of Greenhouse Gases, *Nature*, 433, 403 – 406

Stensrud, D., J.-W. Bao, and T.T. Warner, 2000: Using Initial Condition and Model Physics Perturbations in Short-Range Ensemble Simulations of Mesoscale Convective Systems, *Monthly Weather Review*, v. 128, pp. 2077-2107.

Vapnik, V. N. (1995), The Nature of Statistical Learning Theory, 189 pp., Springer, New York.

Vapnik, V. N. (2006), Estimation of Dependences Based on Empirical Data, translated from Russian by S. Kotz, 495 pp., Springer, New York.

Venema V., A. Schomburg, F. Ament, C. Simmer: 2007, Two adaptive radiative transfer schemes for numerical weather prediction models. Atmospheric Chemistry and Physics 7(21): 5659 – 5674

Washington, W.M. and D.L. Williamson (1977), A Description of NCAR GCM's in General Circulation Models of the Atmospheres. Methods in Computational Physics, J. Chang Ed., 17, Academic Press, 111-172.

Wendland, H (2005): Scattered Data Approximation. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press,.

Wessels, L. F. A., and E. Bernard (1992), Avoiding false local minima by proper initialization of connections, IEEE Trans. Neural Networks, 3, 899–905.