Georgia Tech School of Civil and Environmental Engineering

Bias Correction of Outputs from Global Circulation Models: Focus on the Use of Artificial Neural Networks

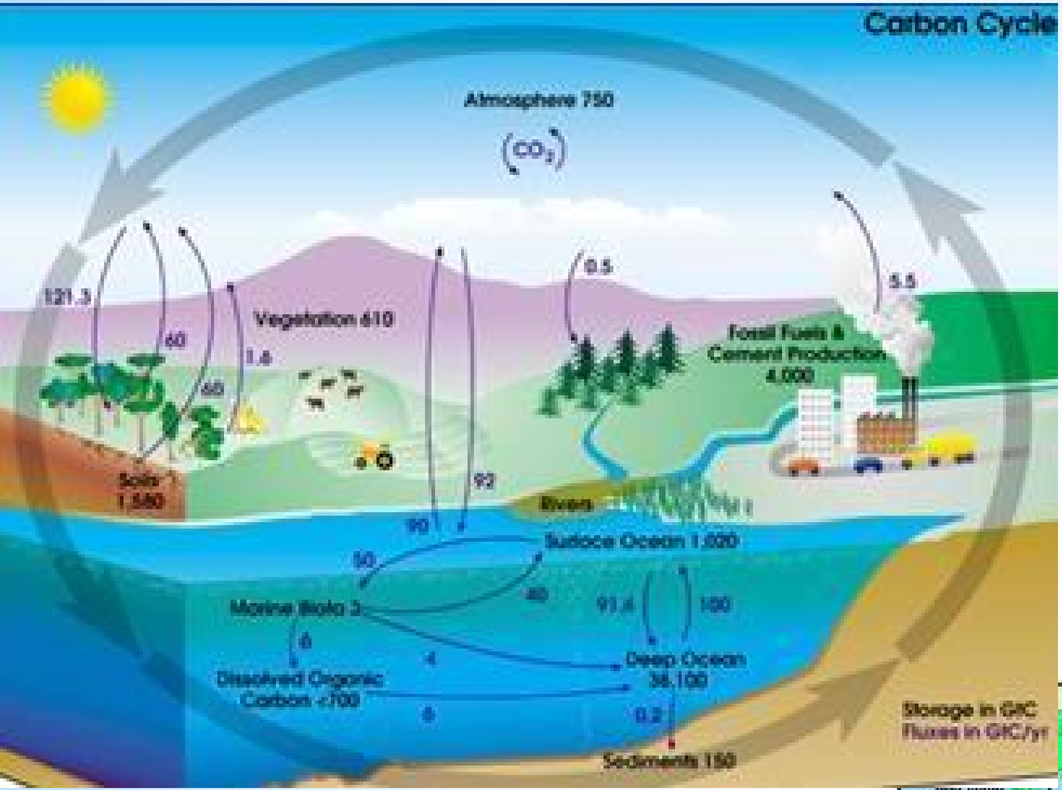
Dissertation of

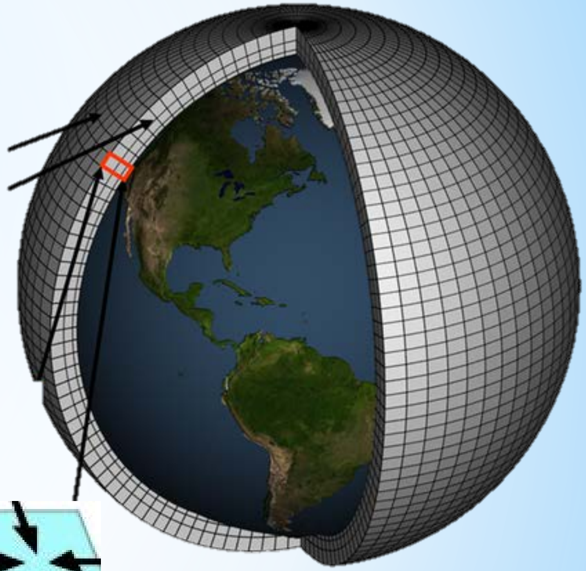
Sanaz Moghim

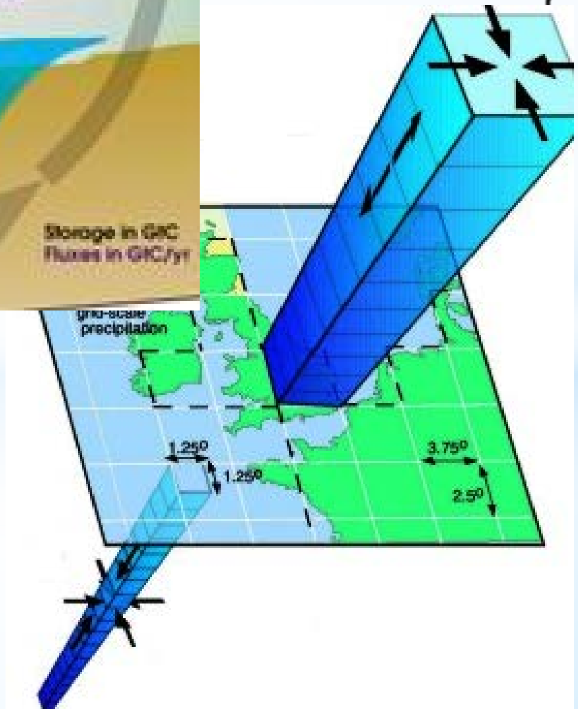
January 28, 2019

Colombella, Italy

Why we need General Circulation Models!







Why we need General Circulation Models!

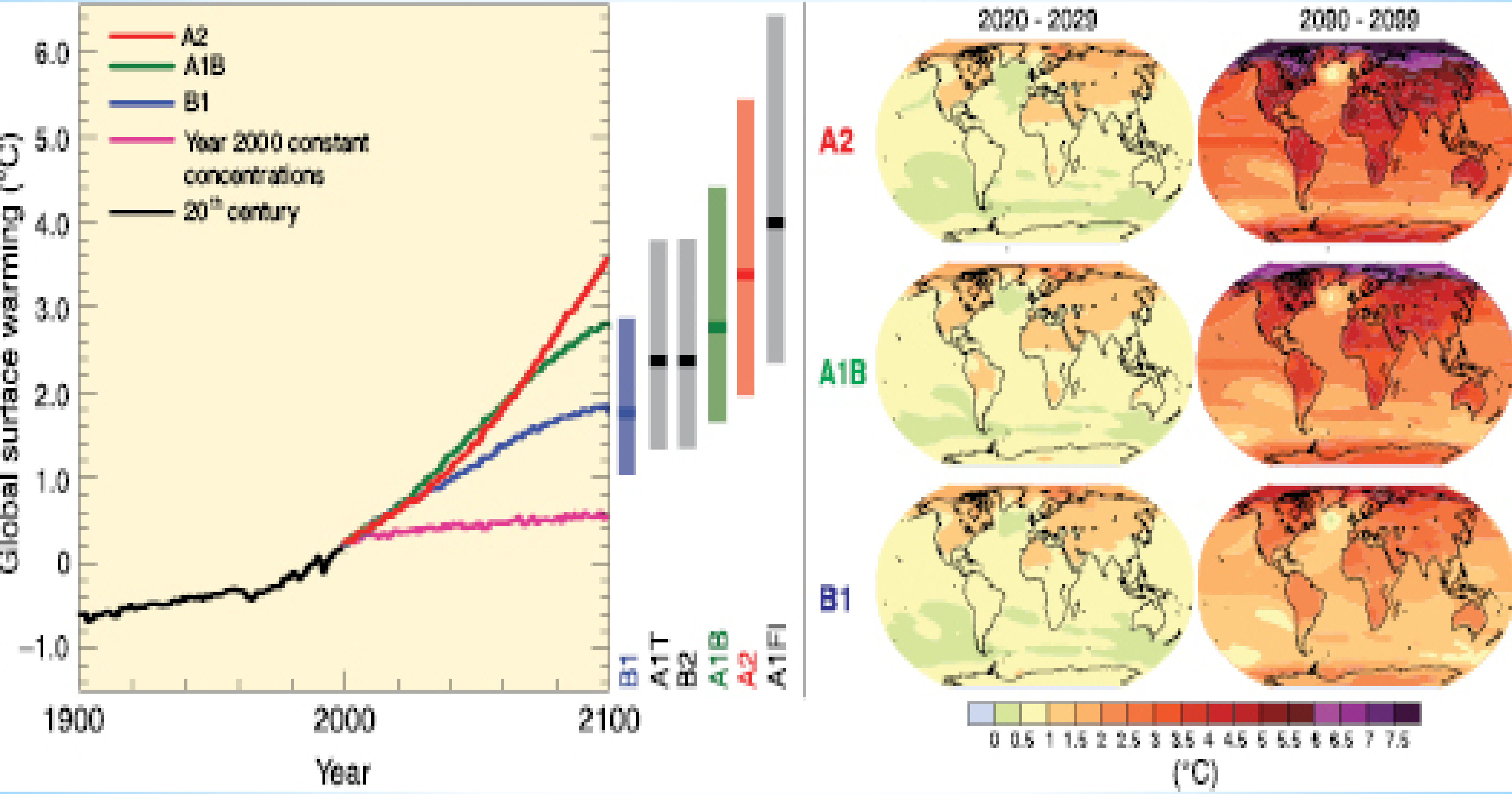


Figure from IPCC Fourth Assessment Report: Climate Change 2007

Why GCMs have bias!

- Inadequate knowledge of the physical processes
- Simplification of the atmospheric circulation

    Key assumptions

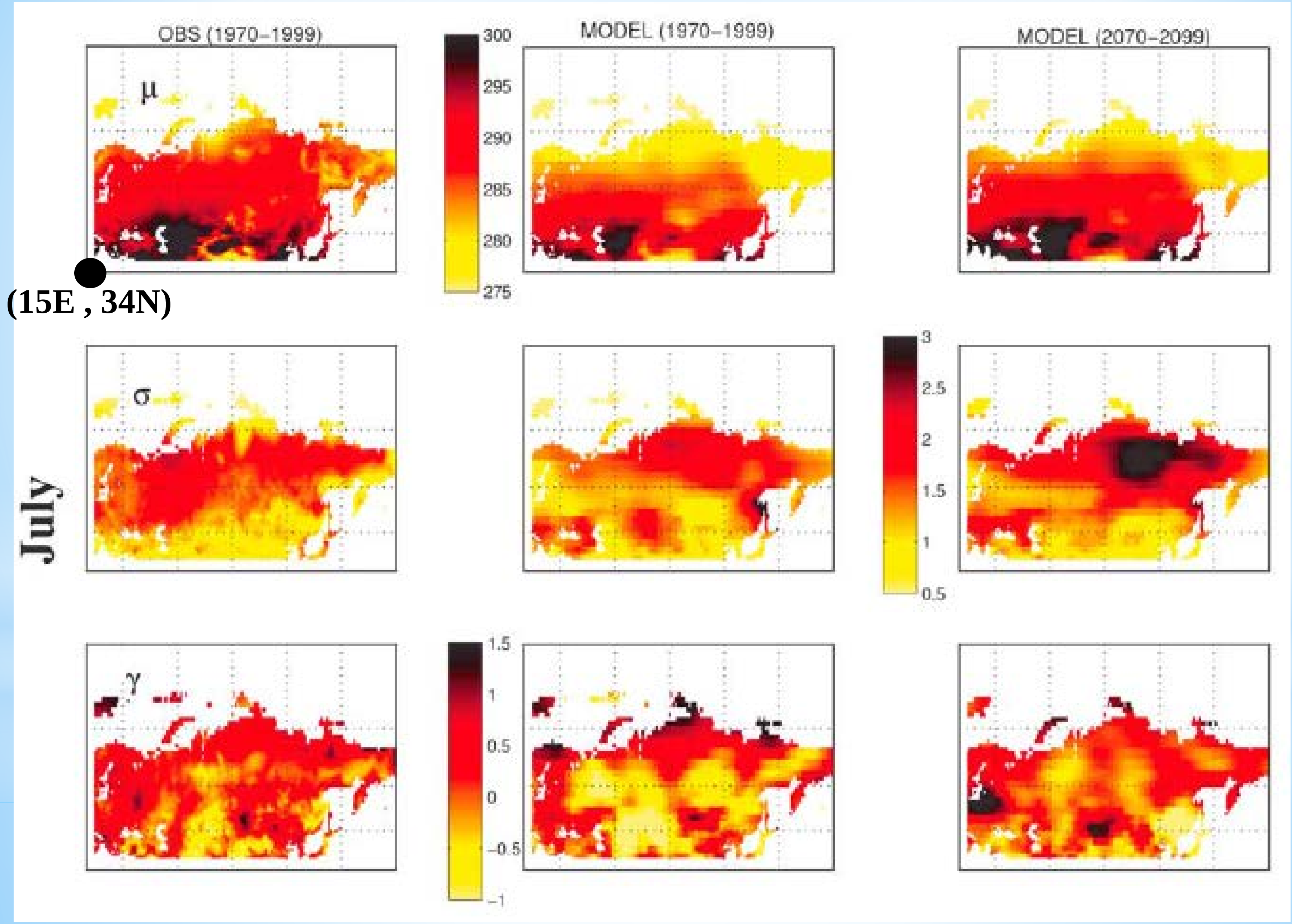
    Physical parameterizations

- Calibration process

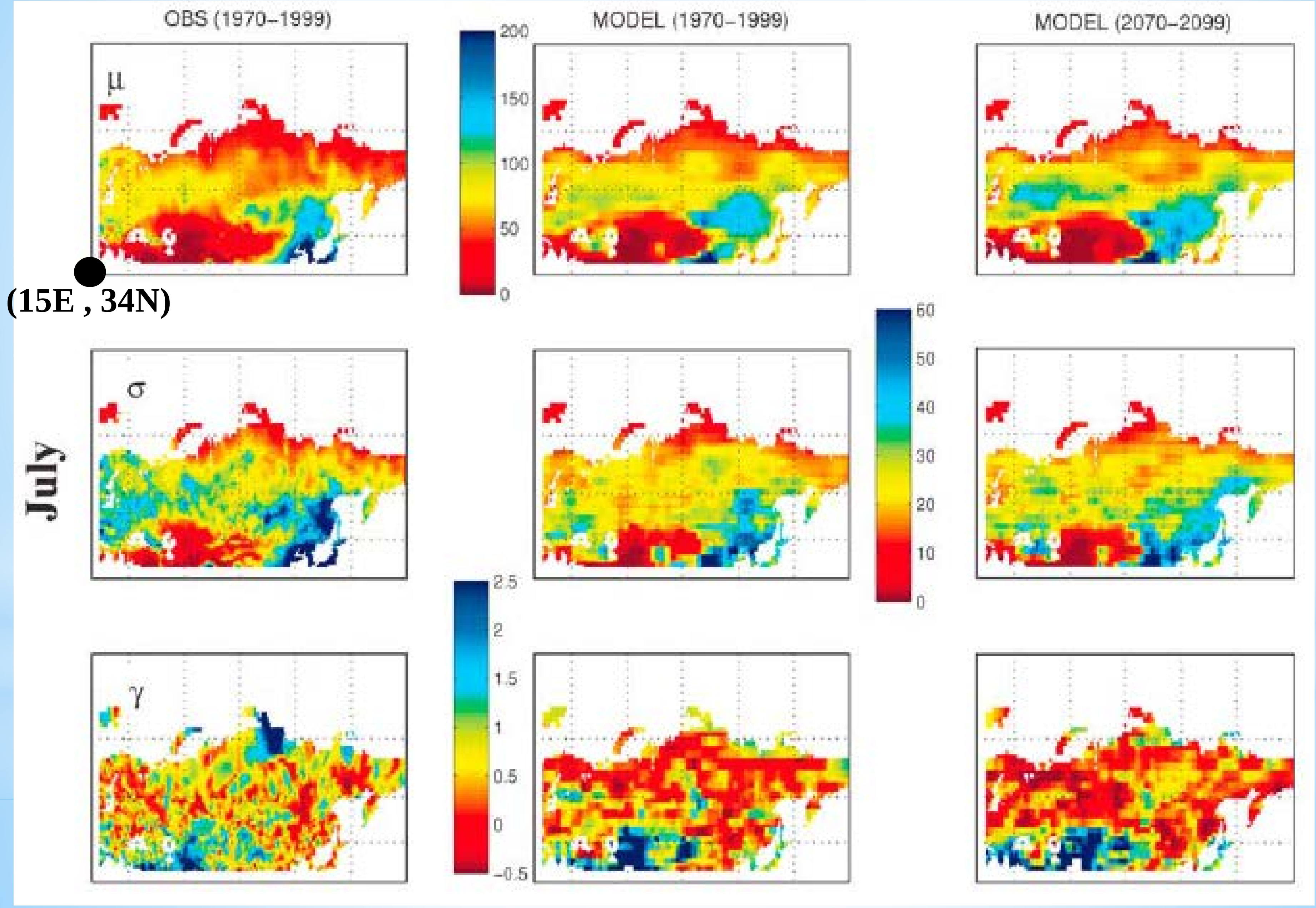
Why we need bias correction!

- Overestimation of the frequency of light precipitation and underestimation of the heavy one by most of the GCMs (Sun et al. 2005).
- Unsatisfying simulation of the tropical precipitation by CCSM (Dai et al. 2001).
- Overestimation of convective precipitation and underestim-ation of stratiform precipitation by PCM (Dai 2006).
- Underestimation of cloud cover and overestimation of SST by HadCM (Dai 2006).

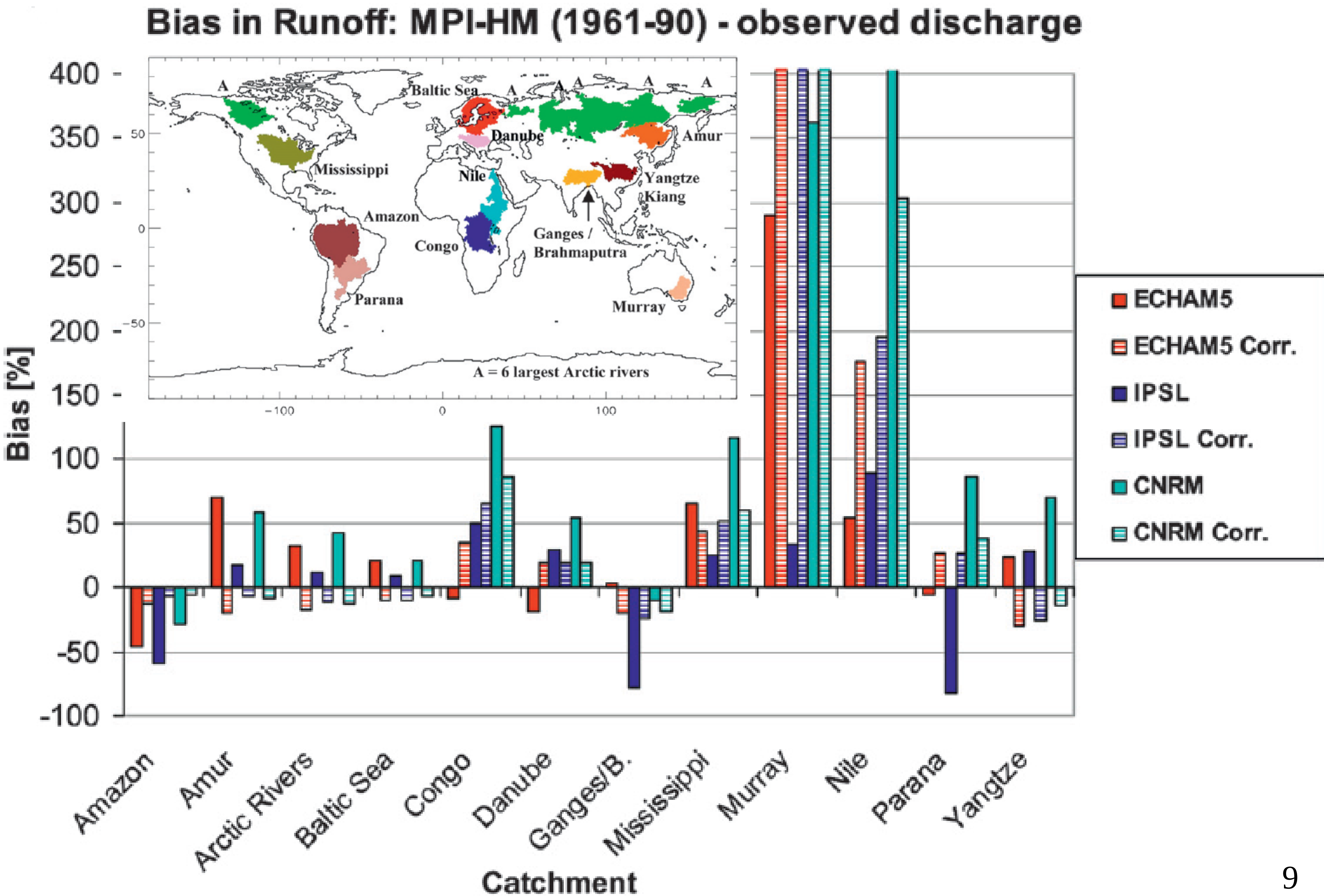
Li et. al (2010) T_PCM (K)



Li et. al (2010) P_PCM (mm month^{-1})



Hagemann et. al (2011)



What has been done!

- **Bias Correction methods:**

I. Dynamical Methods

II. Statistical Methods

What has been done!

- **Bias Correction methods:**

I. Dynamical Methods

II. Statistical Methods

- Delta Change Method:
(Hay et al. 2000; Ines and Hansen 2006; Horton et al. 2011)

- Precipitation

$$P^i_{adj} = P^i \times \frac{\bar{P}^i{}_{obs}}{\bar{P}^i{}_{GCM}}$$

P^i_{adj}: bias-corrected GCM precipitation
$\bar{P}^i$: a long-term monthly mean precipitation from the GCM ($\bar{P}^i{}_{GCM}$) and observations ($\bar{P}^i{}_{obs}$) for a given month i

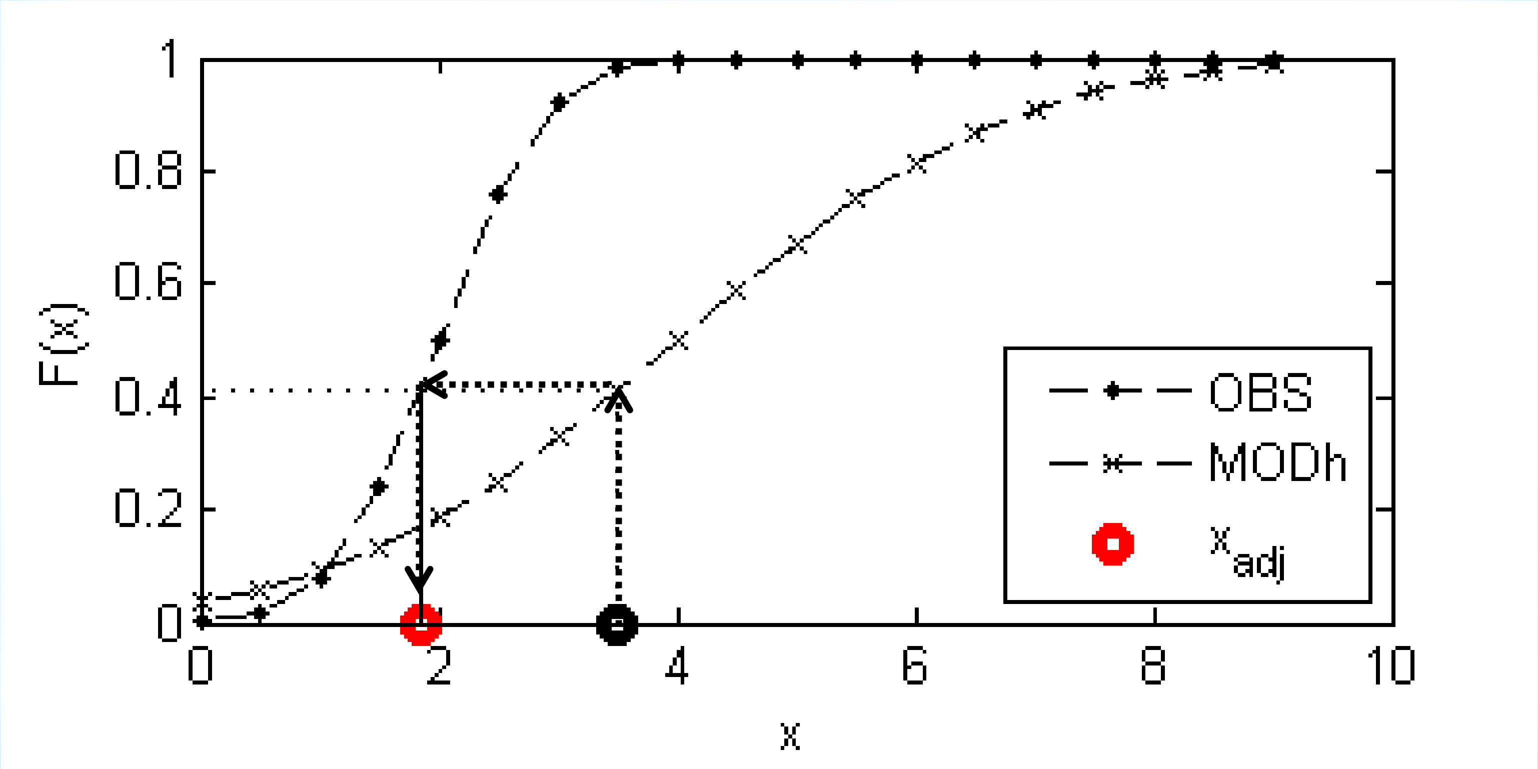
- Temperature

$$T^i_{adj} = T^i + \left(\bar{T}^i_{obs} - \bar{T}^i_{GCM}\right)$$

T^i_{adj}: bias-corrected GCM temperature
$\bar{P}^i$: a long-term monthly mean temperature from the GCM ($\bar{P}^i{}_{GCM}$) and observations ($\bar{T}^i{}_{obs}$) for a given month i

- Quantile Mapping Method (CDF matching):

(Panofsky and Brier 1968; Ines and Hansen 2006; Piani et al. 2010)



$$x_{adj} = F_{obs}^{-1}\left(F_{MODh}(x)\right)$$

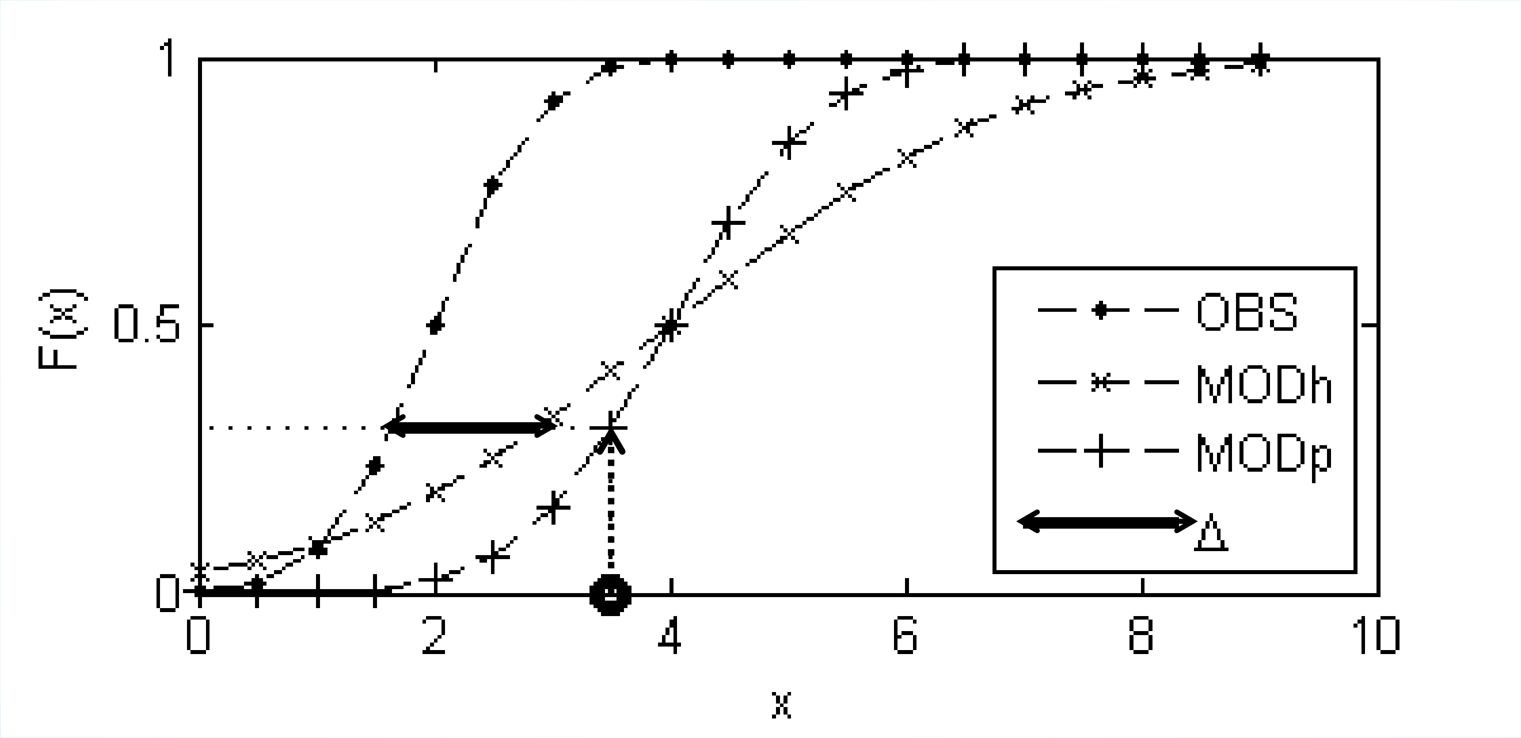
F : cumulative distribution function (CDF)

MODh : historical modeled outputs

OBS : observation

- Equidistant CDF method (EDCDF):

(Li et al. 2010)



$$\Delta = F_{obs}^{-1}\left(F_{MODP}(x)\right) - F_{MODh}^{-1}\left(F_{MODP}(x)\right)$$

$$x_{adj} = x + \Delta$$

MODp : projection modeled outputs

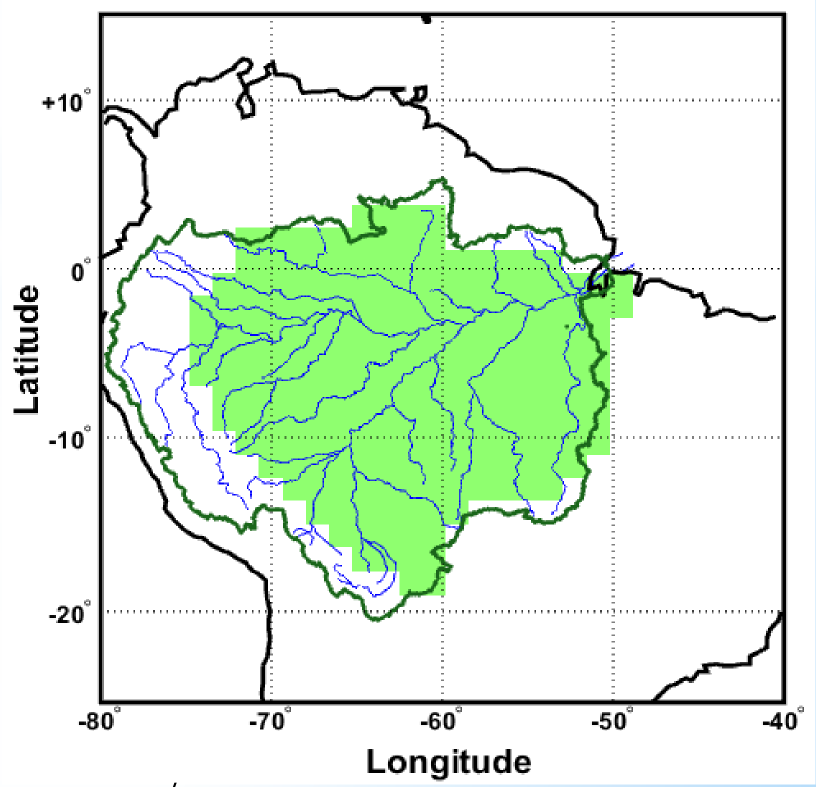
Moore Project 2011-2013

Datasets (under A2 scenario):

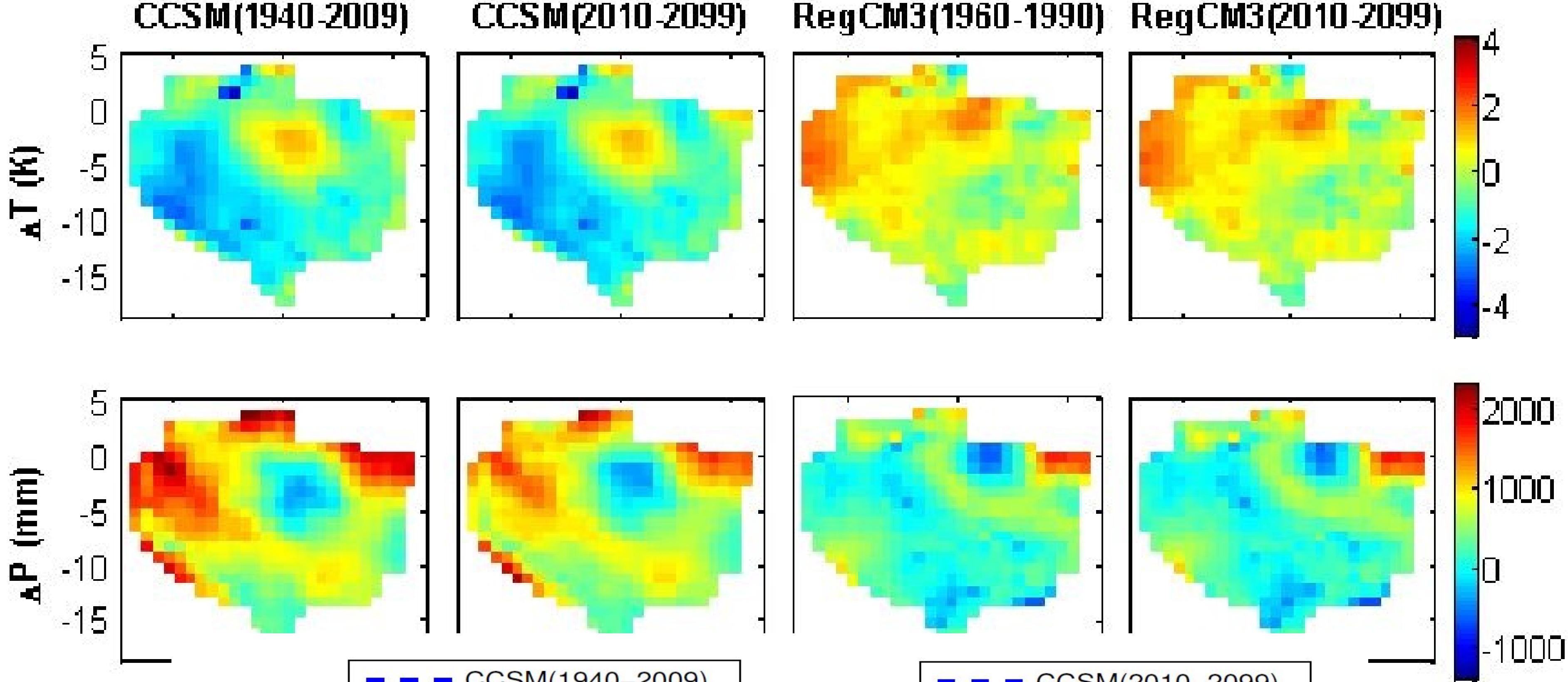
- Community Climate System Model CCSM3 (1940-2009, 2010-2099)
- Regional Climate Model driven by the Hadley Centre Coupled Model RegCM3 (1960-1990, 2010-2099)

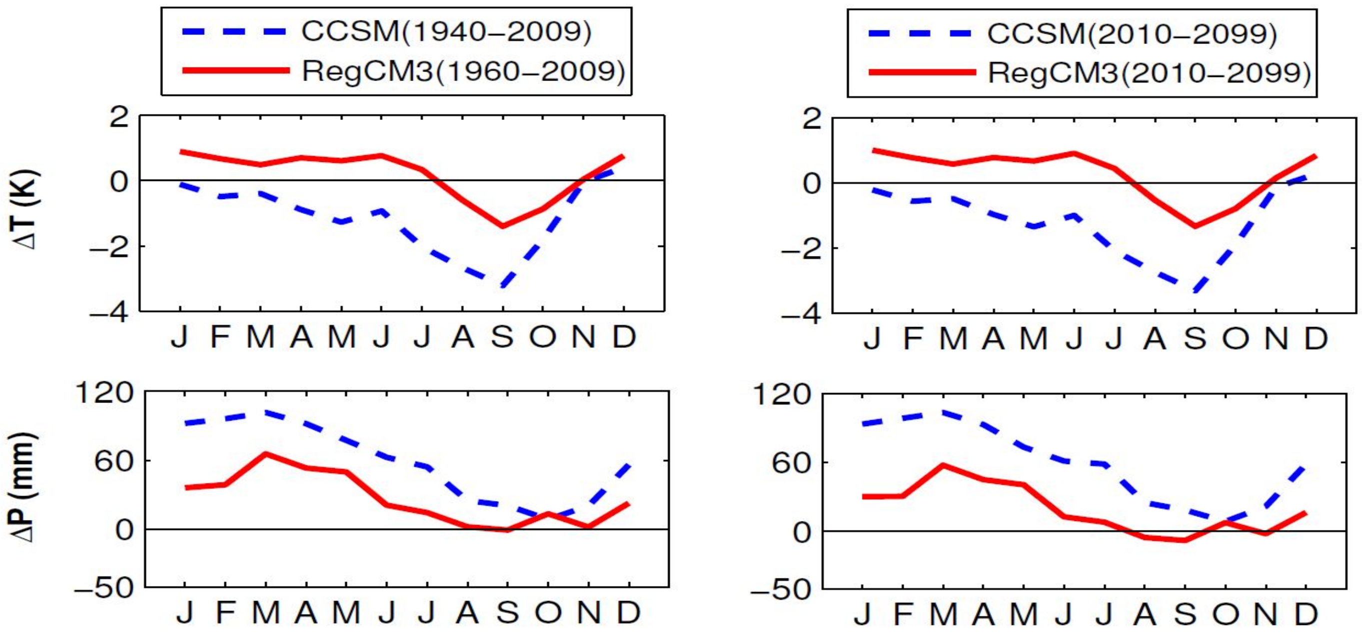
Observation:

- Climate Research Unit (CRU)



http://www.oeb.harvard.edu/faculty/moorcroft/andes-amazon/


CCSM(1940-2009)
CCSM(2010-2099)
RegCM3(1960-1990)
RegCM3(2010-2099)
ΔT (K)
ΔP (mm)


CCSM(1940–2009)
RegCM3(1960–2009)
CCSM(2010–2099)
RegCM3(2010–2099)
ΔT (K)
ΔP (mm)
J F M A M J J A S O N D

Problem Statement

Finding an approach for bias correction

A method should be able to:

I. Learn from available information (observation).
II. Adapt itself.
III. Be generalized (for the case of no information).

Problem Statement

Finding an approach for bias correction

A method should be able to:

I. Learn from available information (observation).
II. Adapt itself.
III. Be generalizable (for the case of no information).

ANN

Mathematical Form

$$y = f(x)$$

y : output x : input

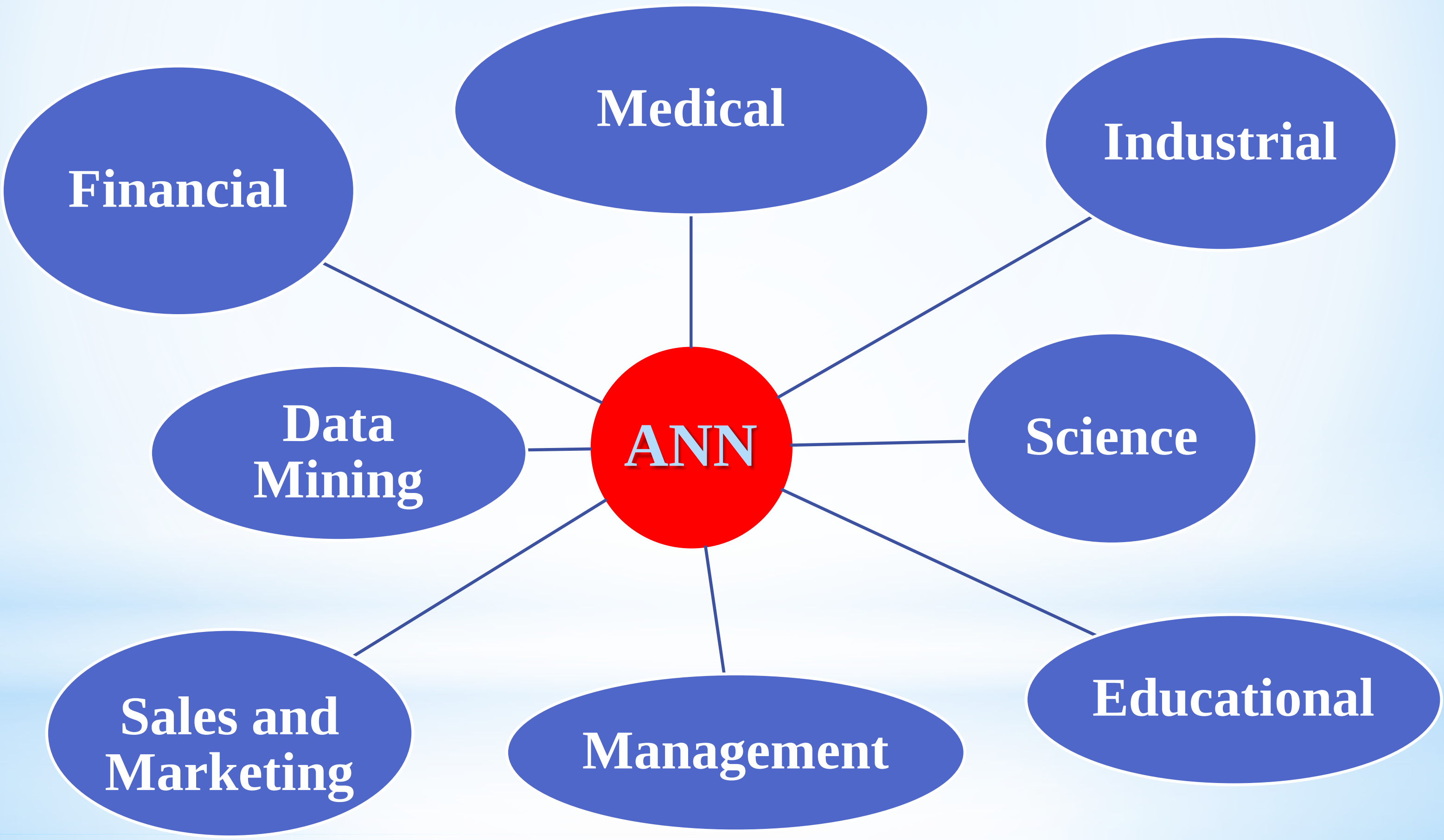
Why Artificial Neural Networks!

- Ability to identify complex and nonlinear functions.

- Having fast computation capability due to parallel structure.

- Having generalization capability.

Applications of ANNs



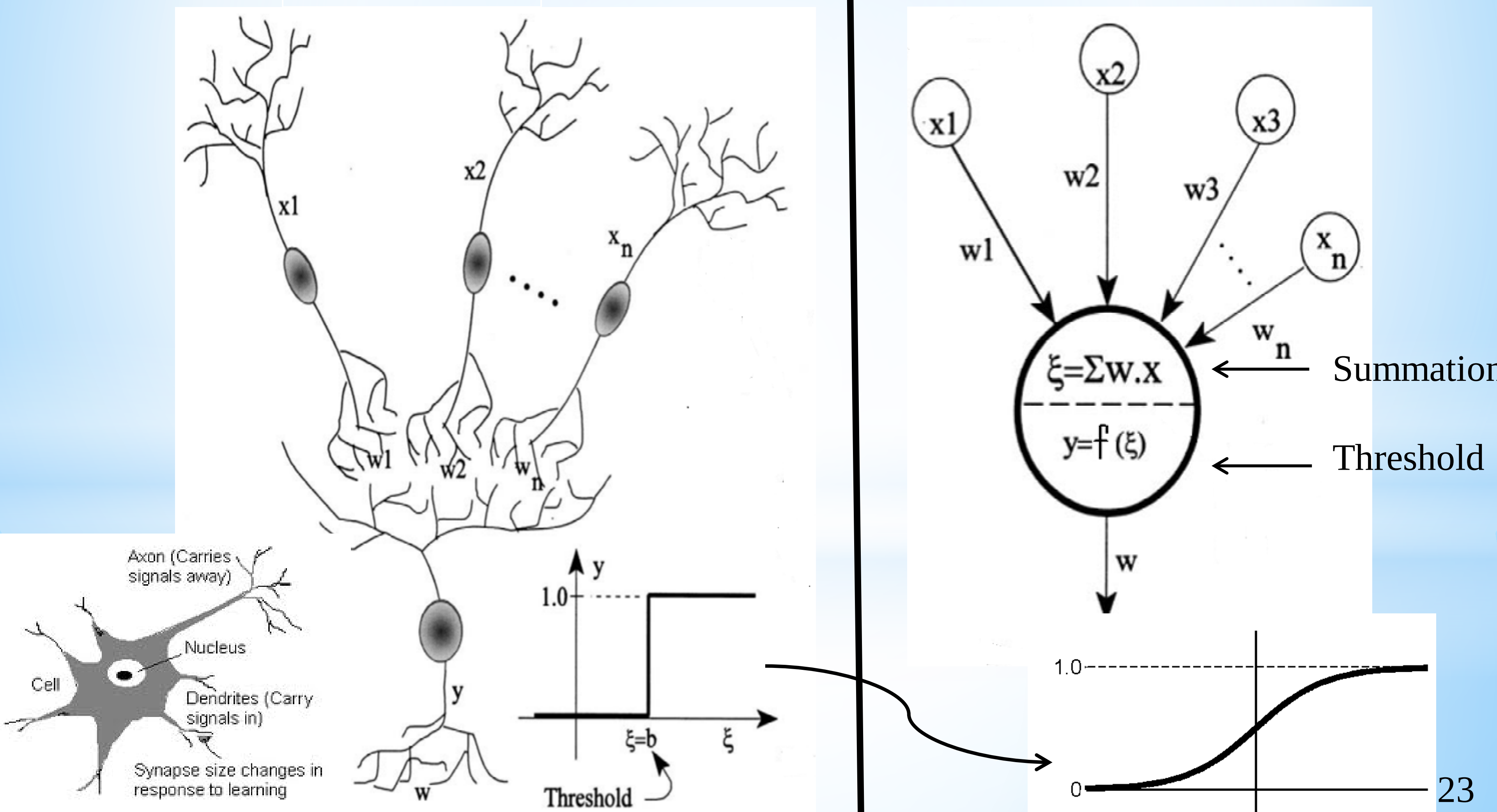
- **Water Recourses Applications**

- Modeling of the rainfall-runoff process (Hsu et al. 1996).
- Rainfall estimation by remote sensing data (Hsu et al. 1999).
- River flow prediction (Cheng et al. 2005; Joorabchi et al. 2007).
- Water quality modeling (Muttil and Chau 2006; May and Sivakumar 2009).
- Simulation of hourly groundwater levels (Taormina et al. 2012).

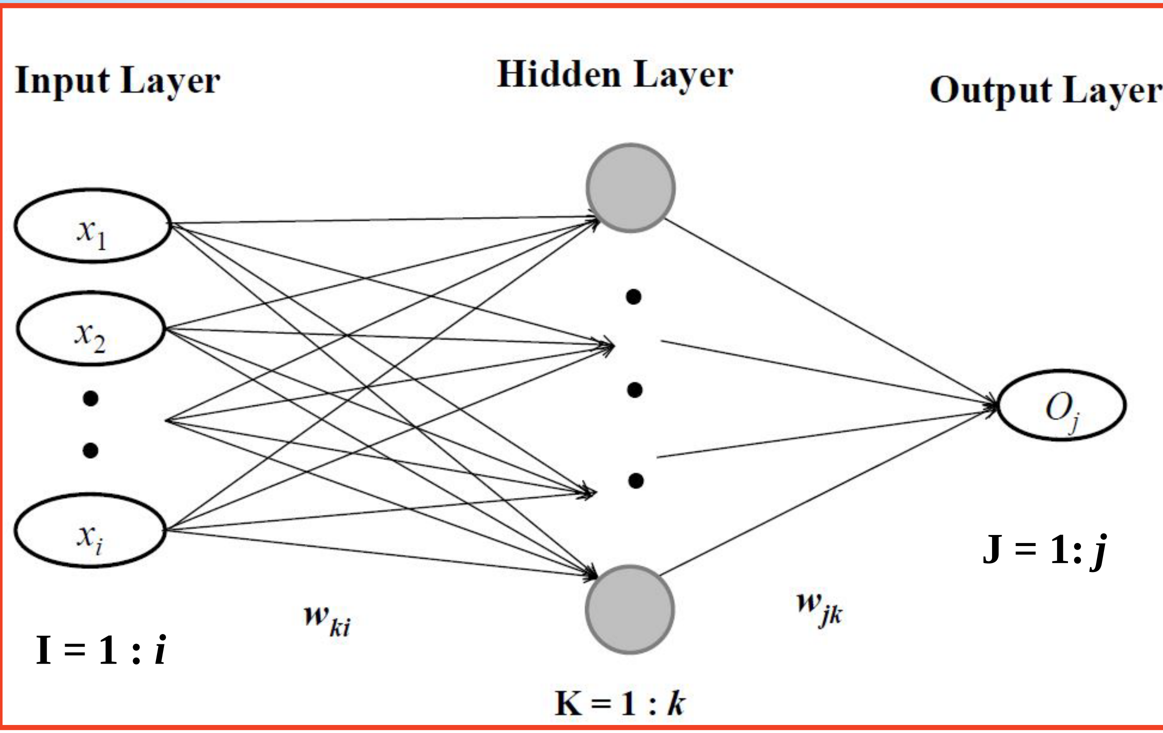
ANN

Biological Neuron

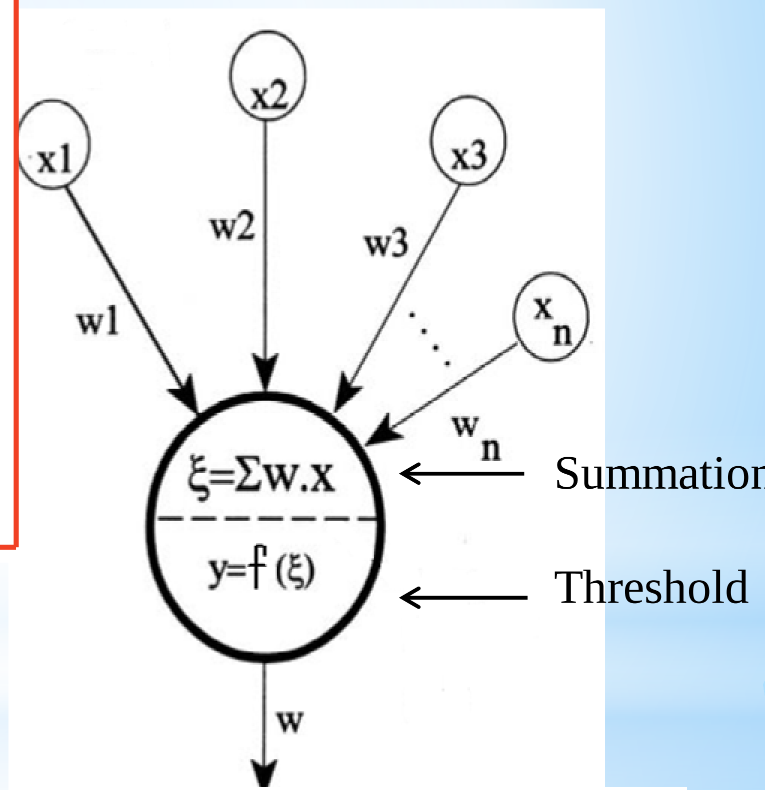
Artificial Neuron

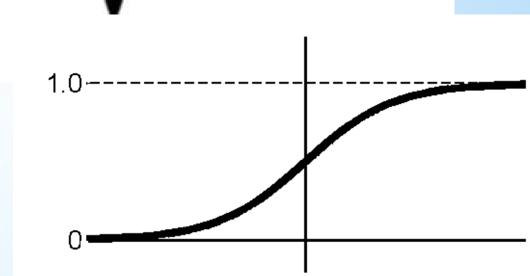


ANN



Artificial Neuron





Experimental Setup of the ANN

- ANN can be described as a set of stimuli-response (S-R) as (Schalkoff, 1997)

$$r_i = f(\underline{s_i}, \underline{w}, a_c)$$

r_i : response (output)
$\underline{s_i}$: stimuli (input)
$\underline{w}$: interconnection weights between the nodes
a_c: combination of unknown characteristics of the network including:

- Training Set
- Network Architecture (geometry, number of hidden nodes and layers)
- Training Algorithm (transfer function, training function, learning rate)

Determination of the Training Set

Physics & Trial and Error

$$\boldsymbol{T, TS, Q, LW_n, SW_n, R_n, P, PS}$$

T : air temperature, TS : skin temperature, Q : specific humidity,
LW_n : net longwave radiation, SW_n : net shortwave radiation,
R_n : net radiation, P : precipitation, PS : surface pressure,

downward longwave radiation (LW_d), downward shortwave radiation SW_d,
horizontal winds (u, v)

Stepwise Approach

a) Temperature

$$\boldsymbol{T, TS, Q, LW_n, SW_n}$$

T : air temperature, TS : skin temperature, Q : specific humidity,
LW_n : net longwave radiation, SW_n : net shortwave radiation,
R_n : net radiation, P : precipitation, PS : surface pressure,

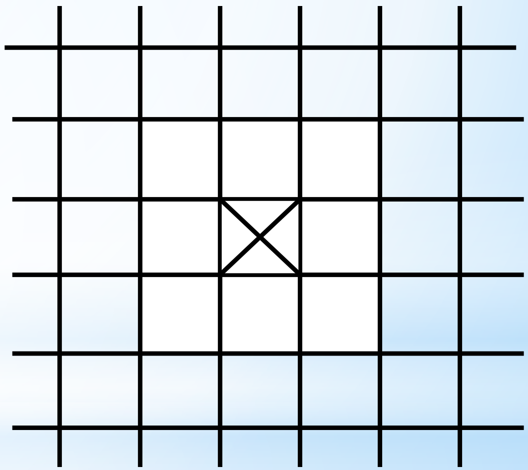
b) Precipitation

$$P_t, P_{t-1}, P_{t-2}, P_{t-3}, \sigma_t^{3by3}$$

P : precipitation

$$\mu^{nbyn} = \frac{1}{N}\sum_{i=1}^{N} P_i$$

$$\sigma^{nbyn} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_i - \mu^{nbyn})^2}$$



surrounding pixels n = 3, 5, or 7

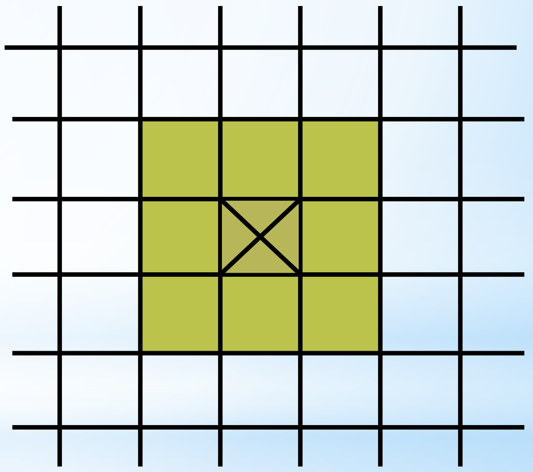
b) Precipitation

$$P_t, P_{t-1}, P_{t-2}, P_{t-3}, \sigma_t^{3by3}$$

P : precipitation

$$\mu^{nbyn} = \frac{1}{N}\sum_{i=1}^{N} P_i$$

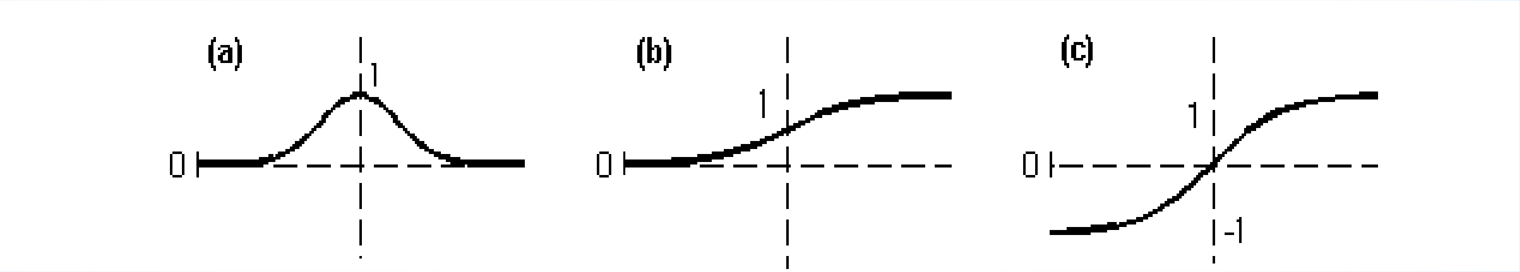
$$\sigma^{nbyn} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_i - \mu^{nbyn})^2}$$



surrounding pixels n = 3, 5, or 7

Determination of the Training Algorithm

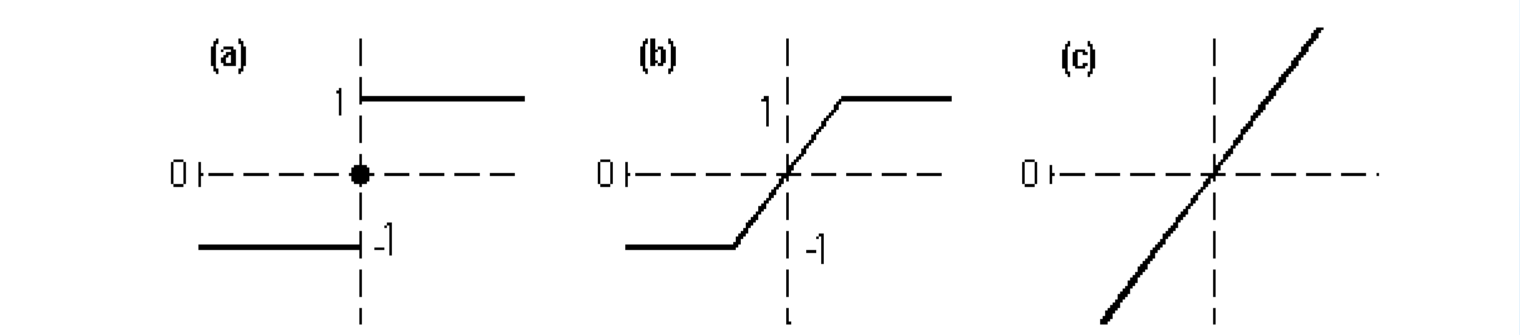
- Transfer Function



$$f(x) = exp\left[-\frac{\|x - O\|^2}{2\sigma^2}\right]$$

$$f(x) = \frac{1}{1+e^{-x}}$$

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$



$$\begin{cases} f(x) = 1 & x > 0 \\ f(x) = 0 & x = 0 \\ f(x) = -1 & x < 0 \end{cases}$$

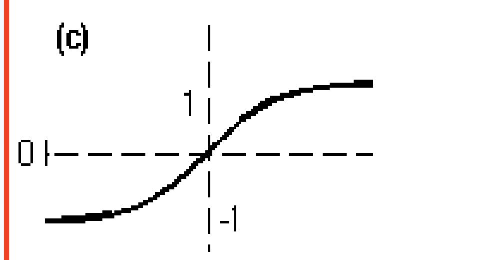
$$\begin{cases} f(x) = 1 & x > 1 \\ f(x) = x & -1 \le x \le 1 \\ f(x) = -1 & x < -1 \end{cases}$$

$$f(x) = x$$

- **Determination of the Training Algorithm**

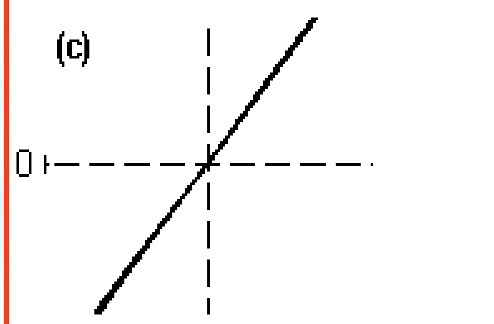
- Transfer Function

Hidden Layer



$$f(x) = \frac{2}{1+e^{-2x}} - 1$$

Output Layer



$$f(x) = x$$

Determination of the Training Algorithm

Training Function

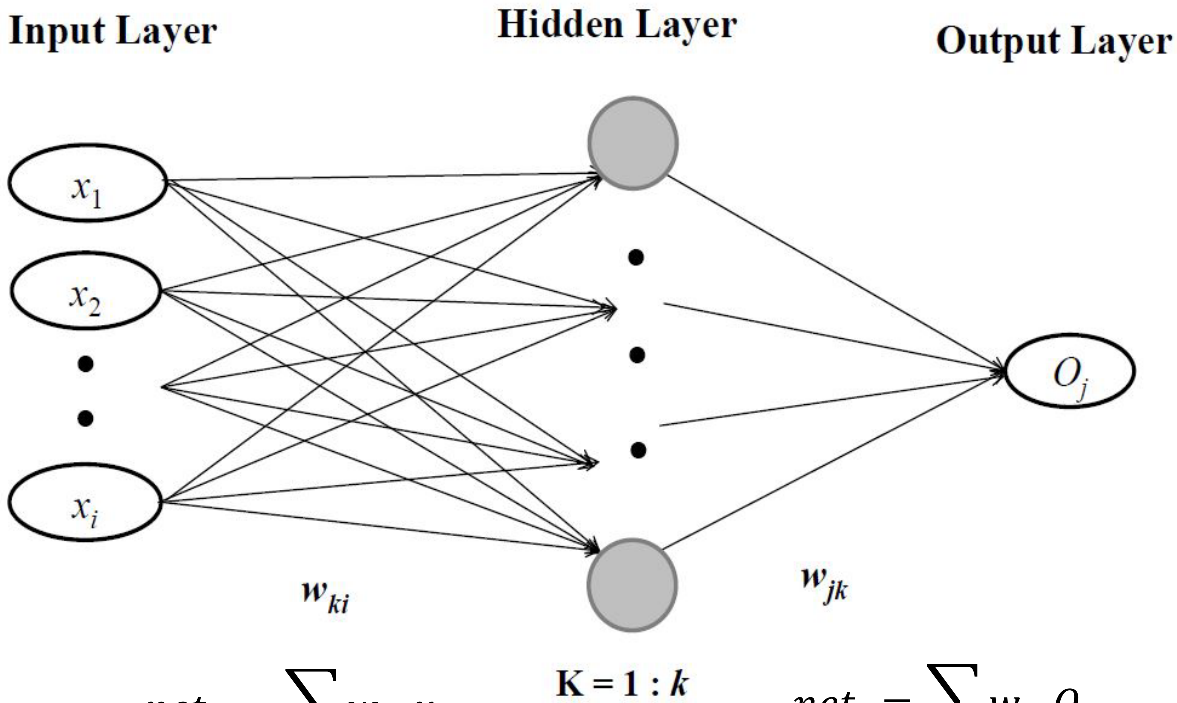
$$\boldsymbol{E} = \sum_{j} \frac{1}{2} \left[Tar_j - O_j \right]^2$$

j : time index
O : output
E : error function
Tar : target

Back Propagation Generalized Delta Rule (BPGDR)

$$\nabla E\,(\underline{w}) = \left(\frac{\partial E}{\partial w_{11}}, \dots, \frac{\partial E}{\partial w_{k1}}, \dots, \frac{\partial E}{\partial w_{ki}}, \dots, \frac{\partial E}{\partial w_{jk}} \right) = 0$$

∇ : gradient or differential operator



$$net_k = \sum_i w_{ki} x_i$$

$$\downarrow$$

$$O_k = f_k(net_k)$$

$$net_j = \sum_k w_{jk} O_k$$

$$\downarrow$$

$$O_j = f_j(net_j)$$

- For the output weights:

$$\Delta w_{jk} = \eta \frac{\partial E}{\partial w_{jk}} = \eta \frac{\partial E}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{jk}} = \eta \frac{\partial E}{\partial O_j} \cdot \frac{\partial O_j}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{jk}}$$

η : learning rate (0-1) **?**

$$\Delta w_{jk} = \eta \left(Tar_j - O_j\right) f_j'(net_j)\, O_k$$

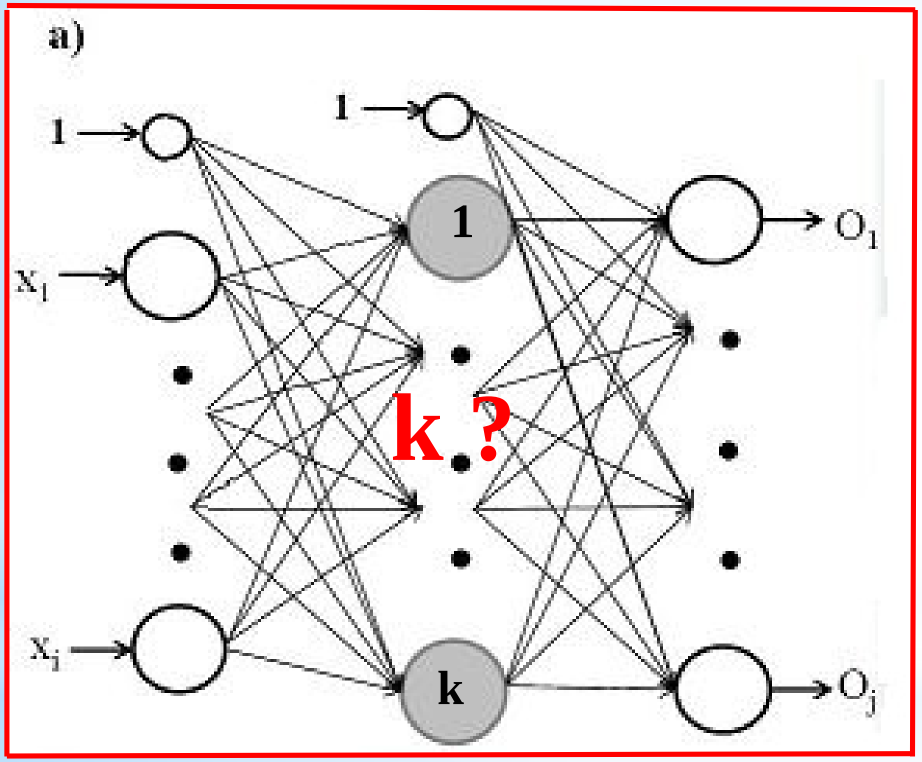
- For the hidden weights:

$$\Delta w_{ki} = \eta \frac{\partial E}{\partial w_{ki}} = \eta \frac{\partial E}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{ki}} = \eta \frac{\partial E}{\partial O_k} \cdot \frac{\partial O_k}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{ki}}$$

$$\Delta w_{ki} = \eta \frac{\partial E}{\partial net_j} w_{jk}\, f_k'(net_k)\, x_i$$

• Determination of the ANN's Architecture

Feedforward Neural Network
FNN



Input layer
1,…., i

Hidden layer

Output layer
1,…., j

One Hidden Layer FNN

❖ Determination of Number of Hidden Nodes (hn)

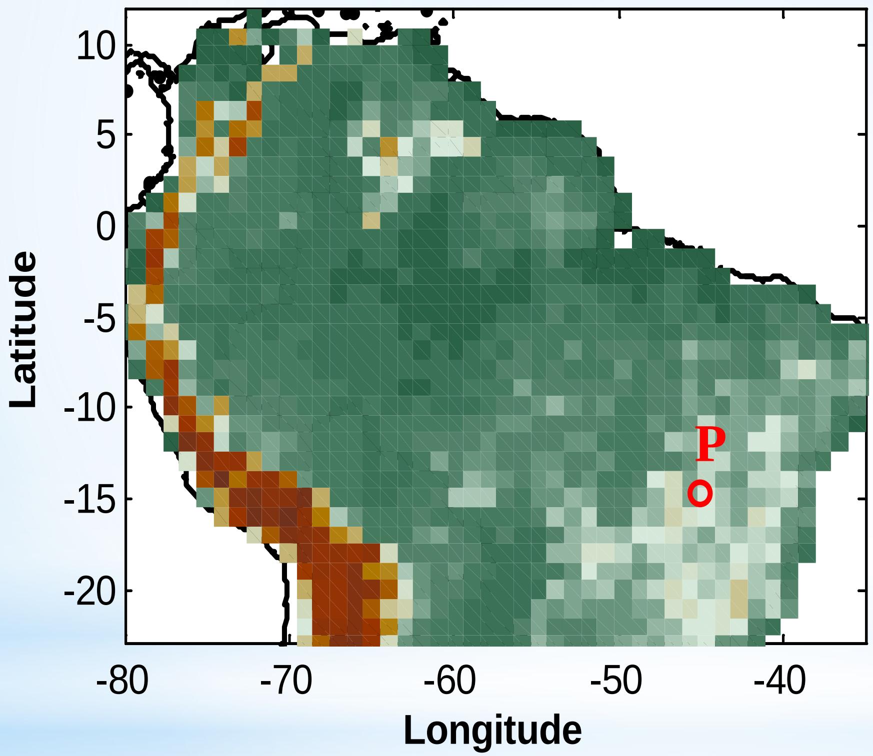
and

Learning Rate (η)

Study Domain

Lon : 80°W to 35°W

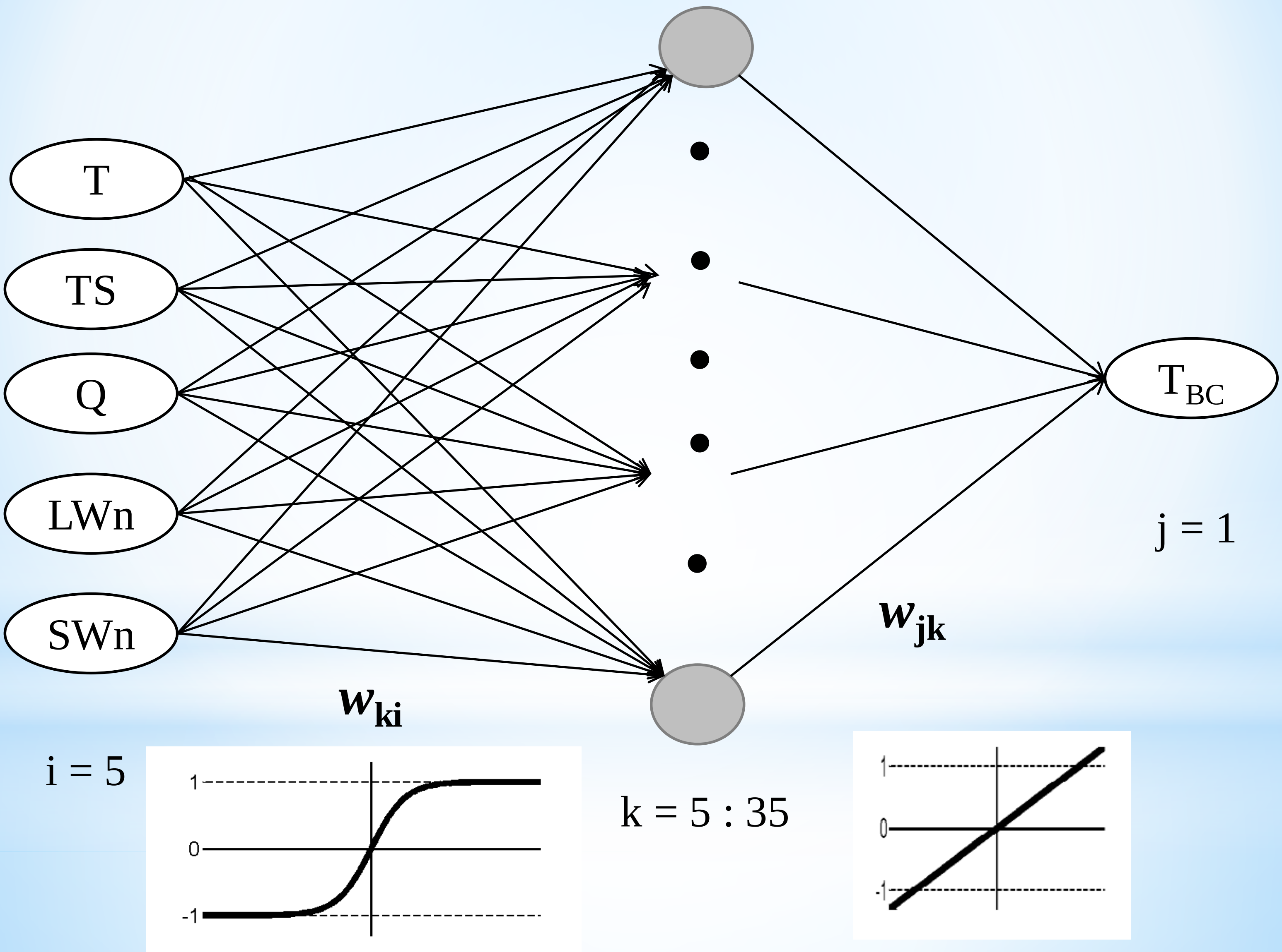
Lat : 23°S to 12°N



P : Lat = 14.71°S , Lon = 45°W

*Temperature

Diagram of the ANN



Datasets

- Community Climate System Model (CCSM3)

Developed by the University Corporation for Atmospheric Research (UCAR).

Spatial Resolution : 1.4° (T85)

Scenarios : 20C3M, A2

Temporal Coverage : 1900-2099

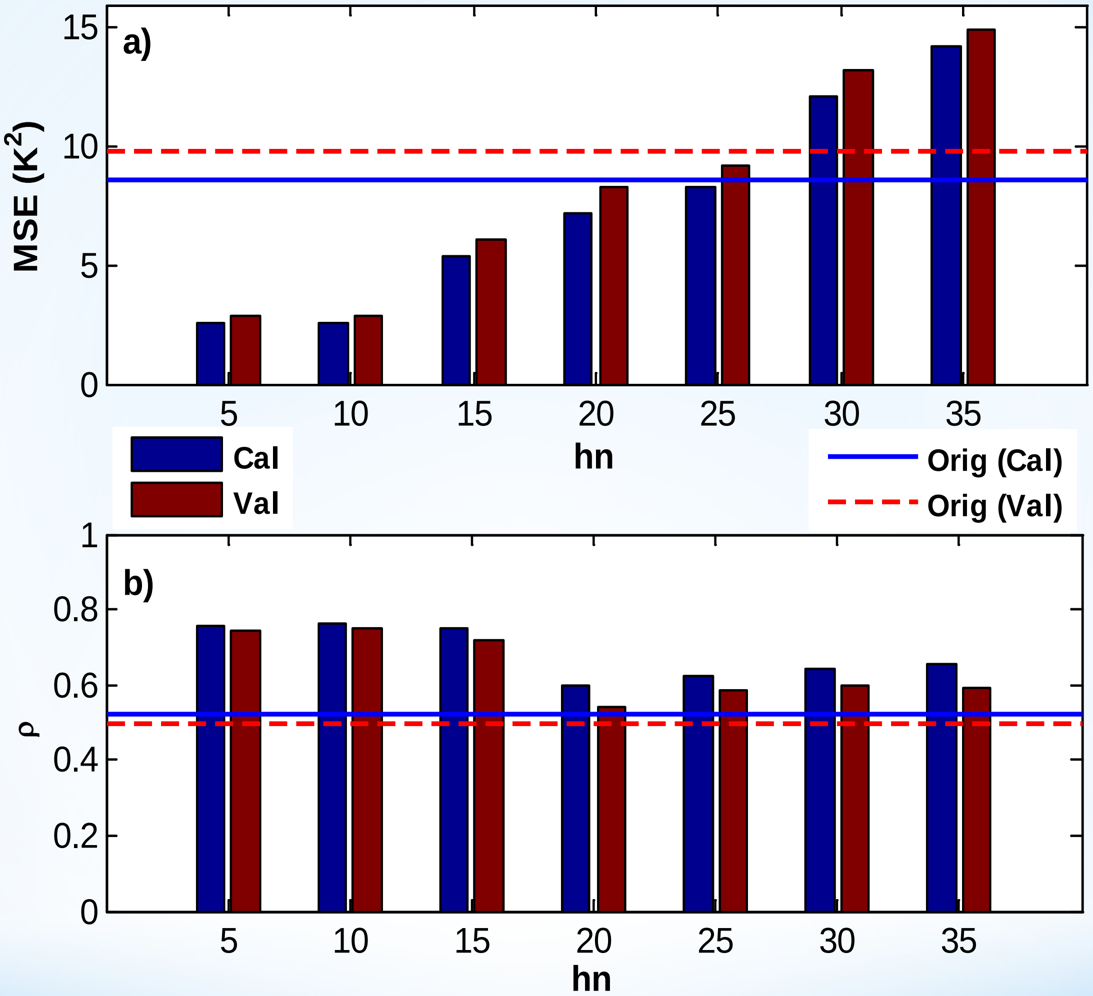
- Meteorological Forcing Data (Sheffield et al. 2006)

Bias-corrected NCEP Reanalysis data.

Spatial Resolution : 1°

Temporal Coverage : 1970-2008

$\eta = 0.01$



Orig : temperature before bias correction and target

Cal: calibration

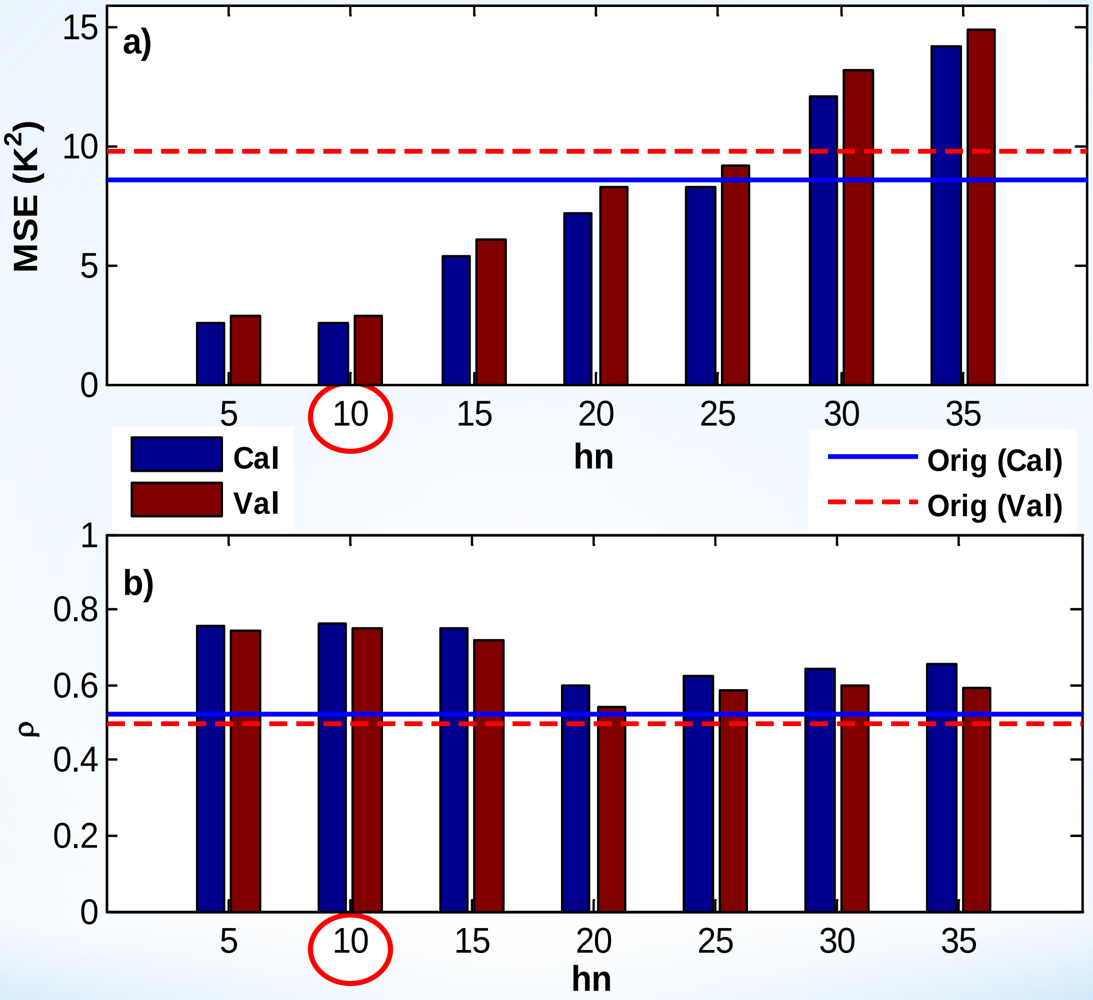
Val: Validation

Mean Square Error $$MSE = E[(Tar - O)^2]$$

Correlation $$\rho = \frac{E[(Tar - E[Tar])(O - E[O])]}{\sqrt{E[(Tar - E[Tar])^2]\, E[(O - E[O])^2]}}$$

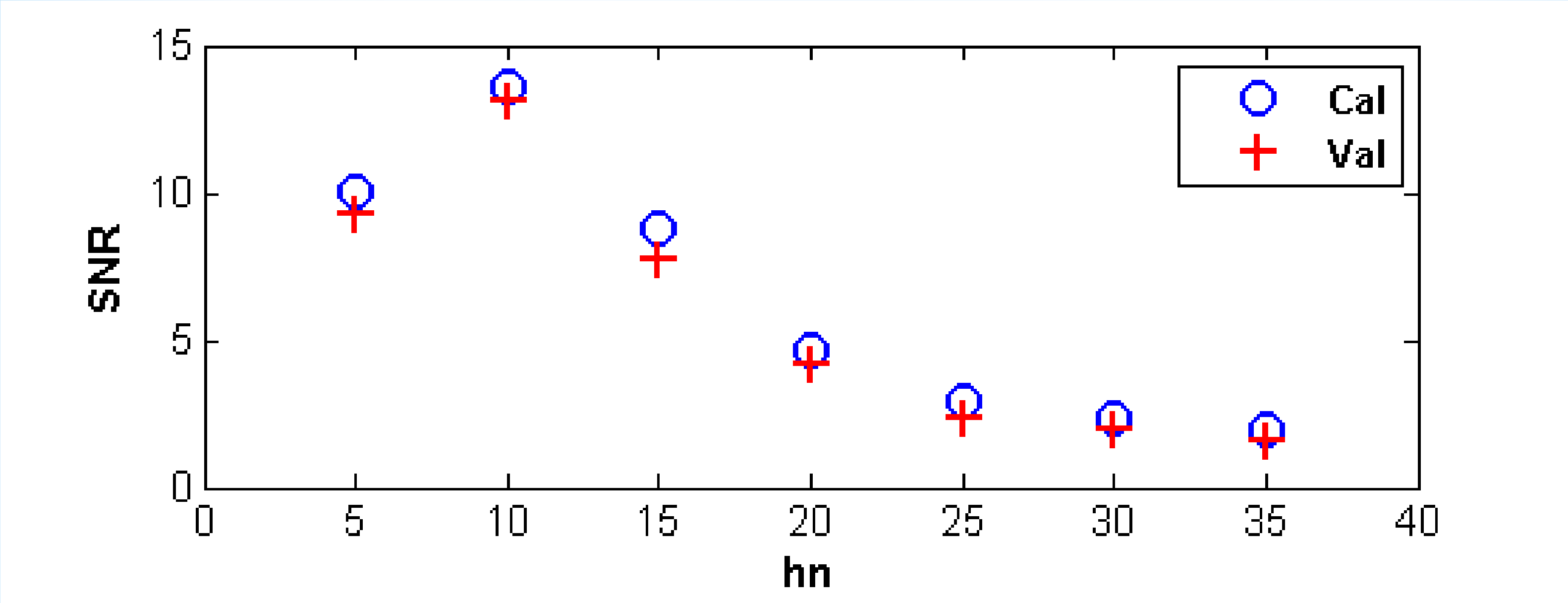
$\eta = 0.01$

Orig : temperature before bias correction and target



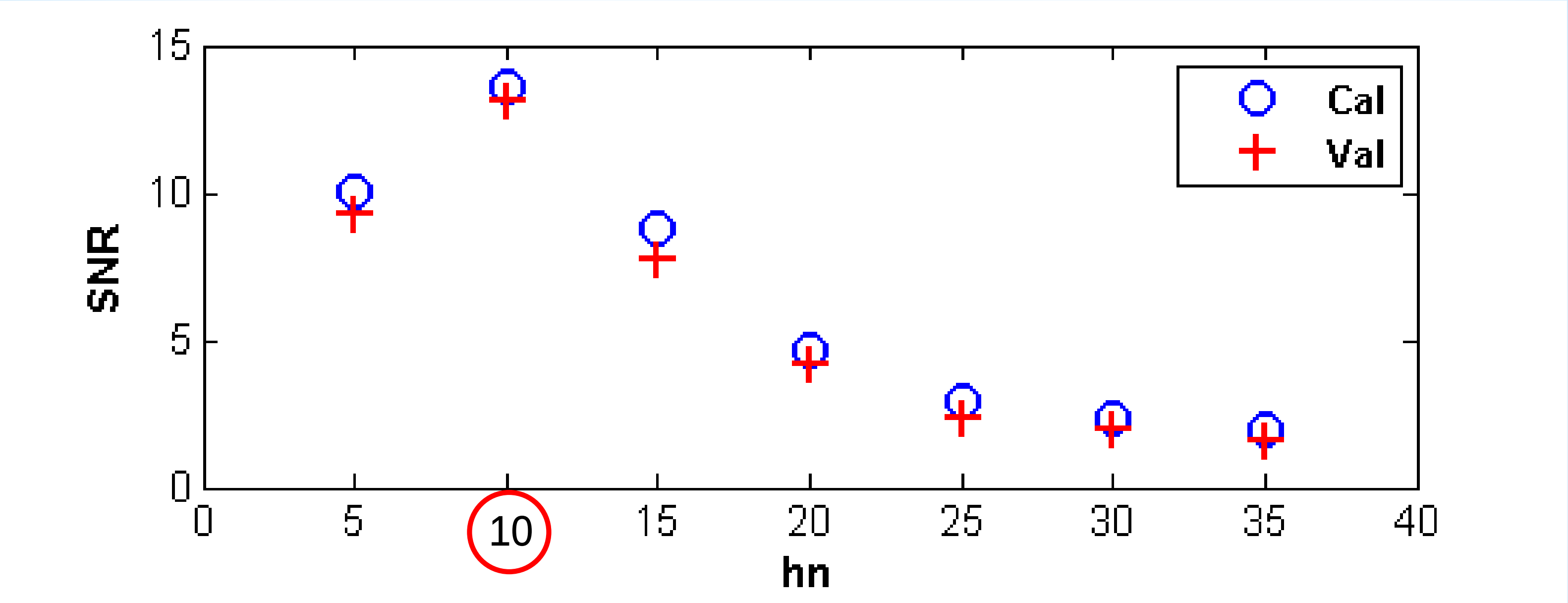
Mean Square Error $$MSE = E[(Tar - O)^2]$$

Correlation $$\rho = \frac{E[(Tar - E[Tar])(O - E[O])]}{\sqrt{E[(Tar - E[Tar])^2]\, E[(O - E[O])^2]}}$$



Signal to Noise Ratio $$SNR = \frac{\sigma_T}{|\sigma_T - \sigma_O|}$$

σ_O : standard deviations of the ANN outputs
σ_T : standard deviations of the targets



Signal to Noise Ratio

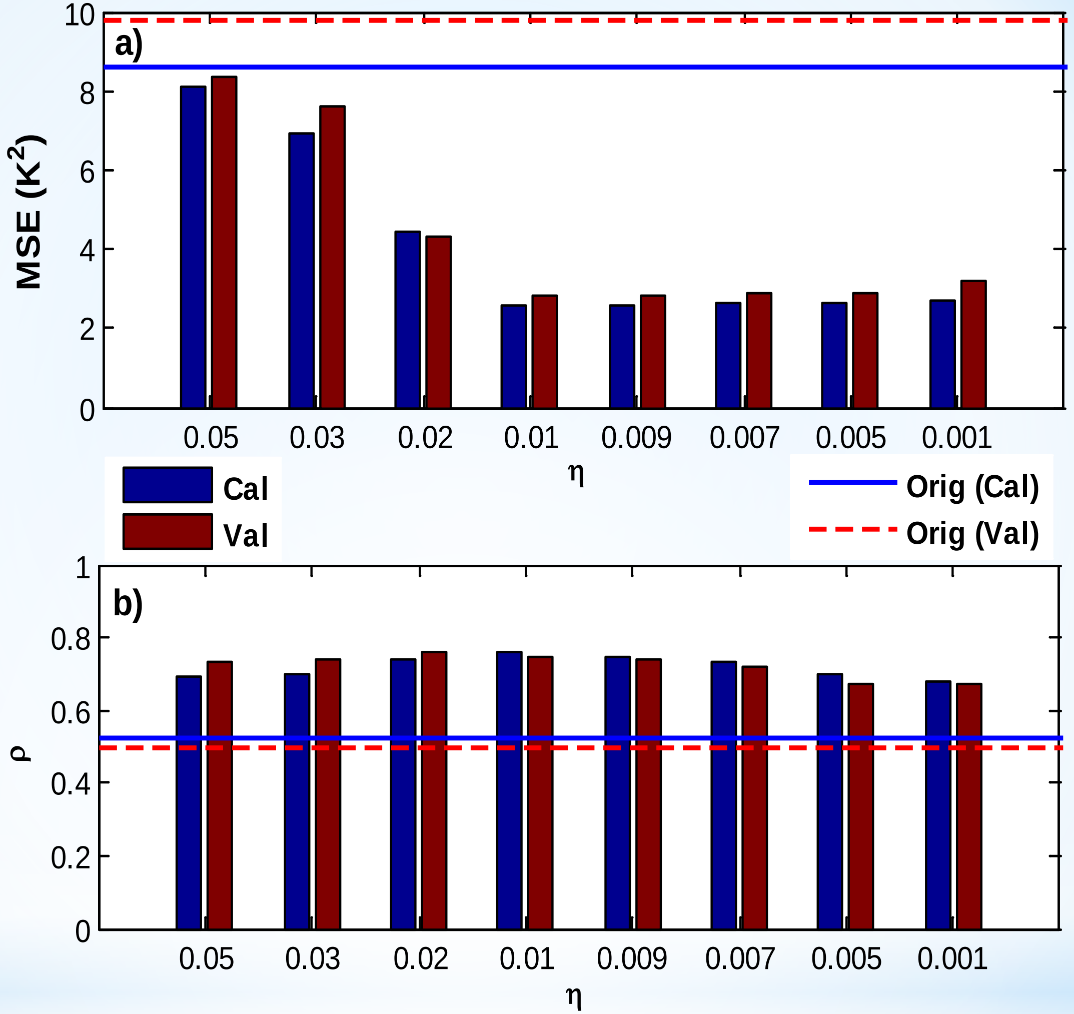
$$SNR = \frac{\sigma_T}{|\sigma_T - \sigma_O|}$$

σ_O : standard deviations of the ANN outputs

σ_T : standard deviations of the targets

$hn = 10$

Orig : temperature before bias correction and target

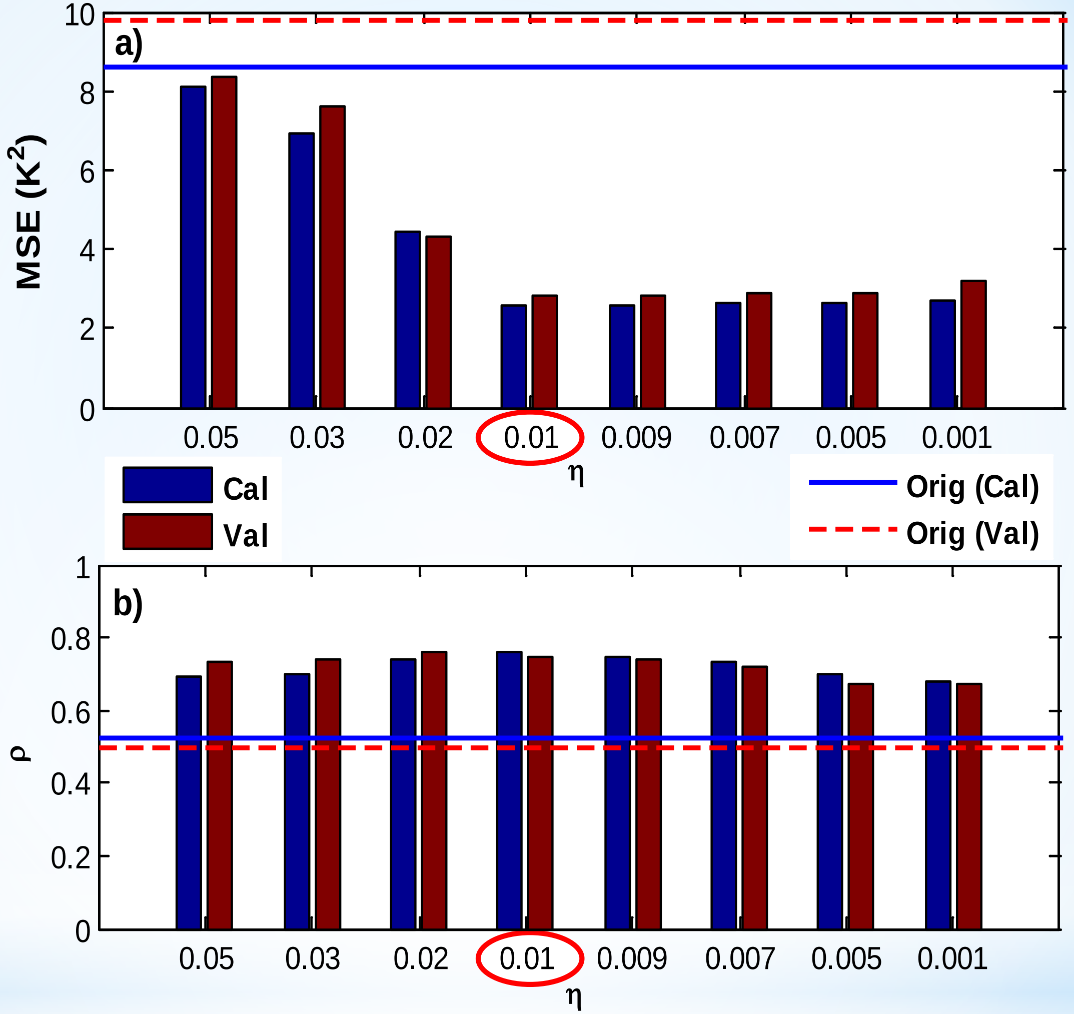


Mean Square Error $$MSE = E[(Tar - O)^2]$$

Correlation $$\rho = \frac{E[(Tar - E[Tar])(O - E[O])]}{\sqrt{E[(Tar - E[Tar])^2]\, E[(O - E[O])^2]}}$$

$hn = 10$

Orig : temperature before bias correction and target

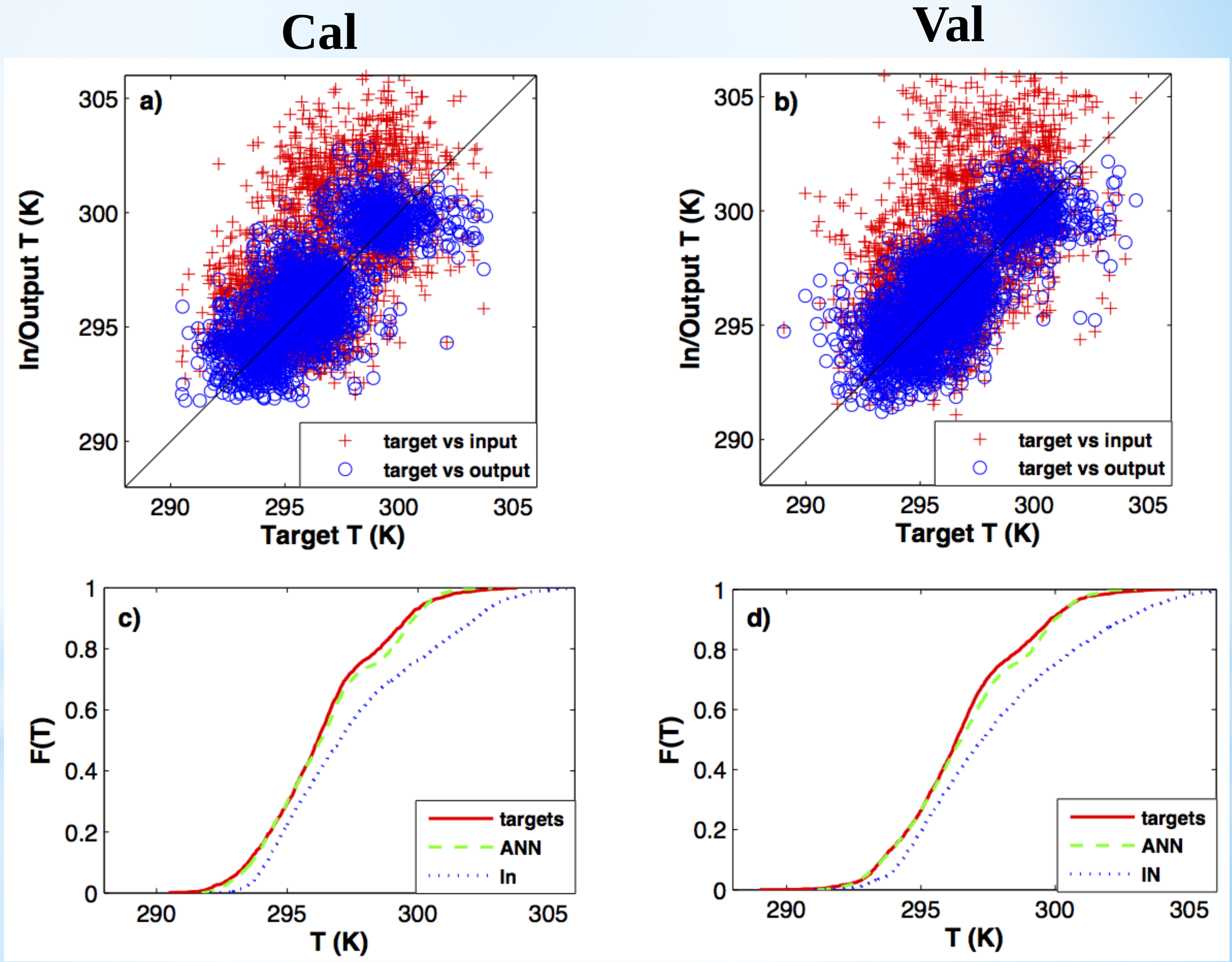


Mean Square Error $MSE = E[(Tar - O)^2]$

Correlation $$\rho = \frac{E[(Tar - E[Tar])(O - E[O])]}{\sqrt{E[(Tar - E[Tar])^2]\, E[(O - E[O])^2]}}$$

Performance of the ANN

$$hn = 10$$
$$\eta = 0.01$$



T	MSE		Bias		ρ		*KS*	
	Cal	Val	Cal	Val	Cal	Val	Cal	Val
In	8.59	9.77	2.30	2.40	0.52	0.50	0.17	0.19
CDF	5.31	5.56	1.85	1.88	0.50	0.49	0.004	0.03
EDCDF	5.25	5.95	1.84	1.93	0.50	0.48	0.005	0.02
LR	3.67	4.01	1.52	1.60	0.70	0.70	0.04	0.05
ANN	2.58	2.82	1.24	1.33	0.76	0.75	0.032	0.03

KS : Kolmogorov-Smirnov test

T	**MSE**		**Bias**		**ρ**		***KS***	
	Cal	**Val**	**Cal**	**Val**	**Cal**	**Val**	**Cal**	**Val**
In	8.59	9.77	2.30	2.40	0.52	0.50	0.17	0.19
CDF	**38%**	**43%**	1.85	1.88	0.50	0.49	0.004	0.03
EDCDF	**39%**	**39%**	1.84	1.93	0.50	0.48	0.005	0.02
LR	**57%**	**59%**	1.52	1.60	**35%**	**40%**	0.04	0.05
ANN	**70%**	**71%**	1.24	1.33	**46%**	**50%**	0.032	0.03

KS : Kolmogorov-Smirnov test

Pixel By Pixel Correction

Improvement Ratio (Imp)

$$ImpA = \frac{(A_{orig} - A_{out})}{A_{orig}}$$

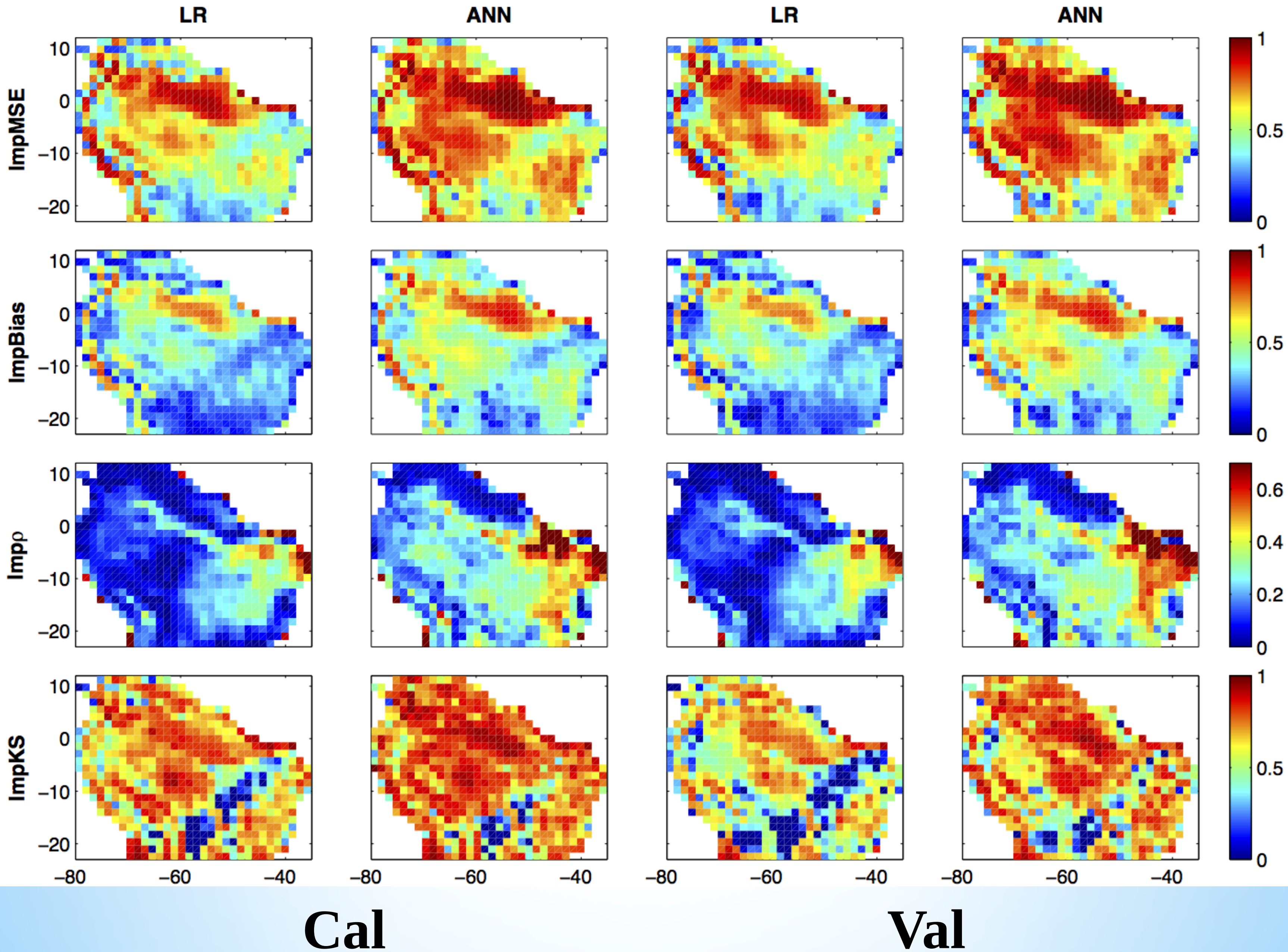
Imp : Improvement Ratio

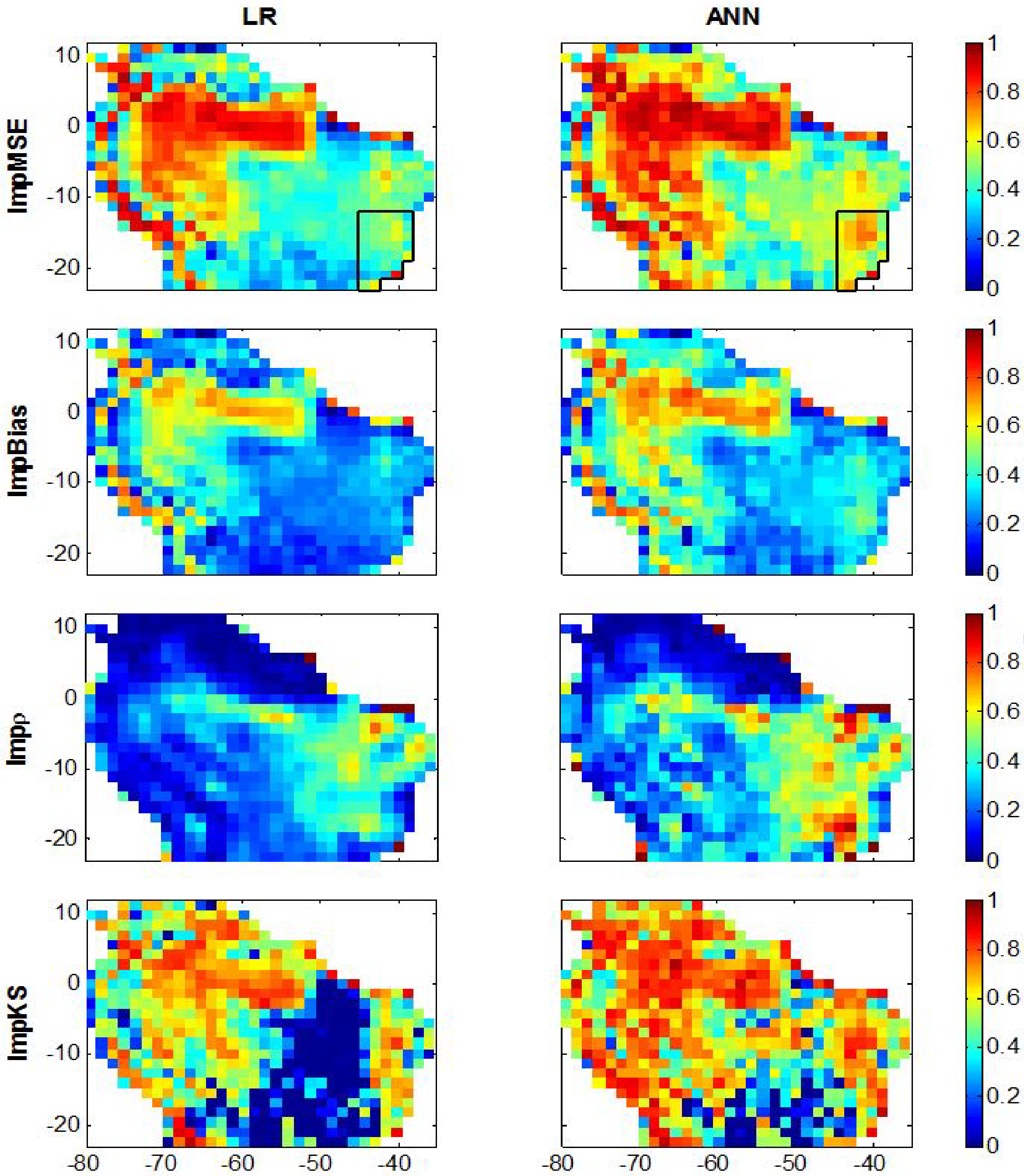
A : MSE, Bias, ρ, or KS

A_{orig} : statistics of original CCSM temperature relative to observations

A_{out} : statistics of regression models outputs (LR and ANN) relative to observations

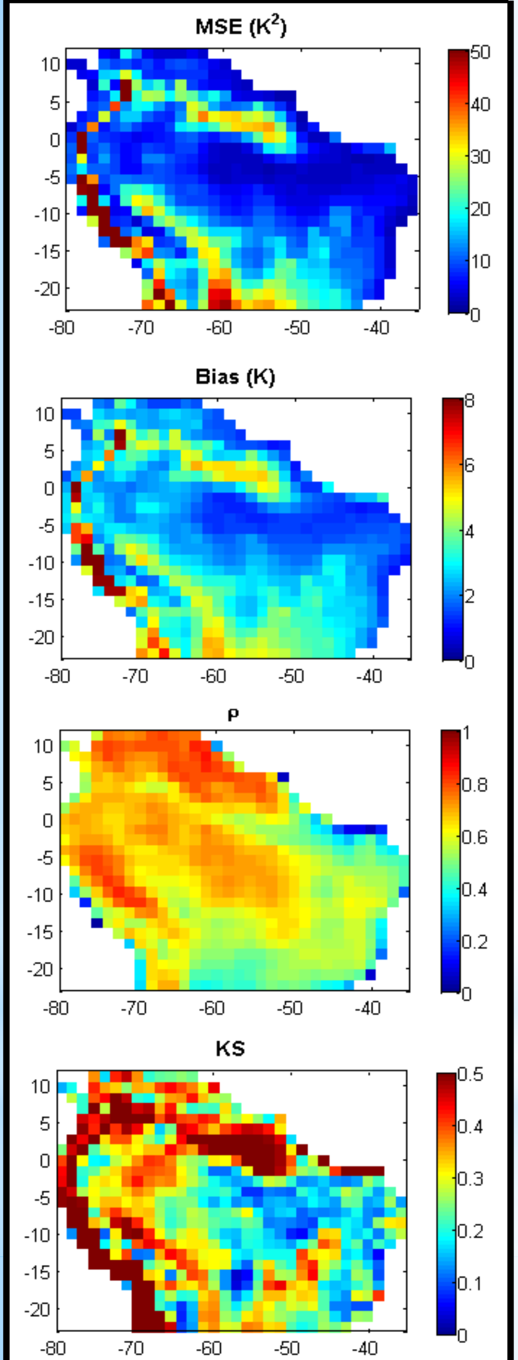
Results


LR
ANN
LR
ANN
ImpMSE
ImpBias
Impρ
ImpKS
10
0
−10
−20
−80
−60
−40
1
0.5
0
0.6
0.4
0.2
Cal
Val

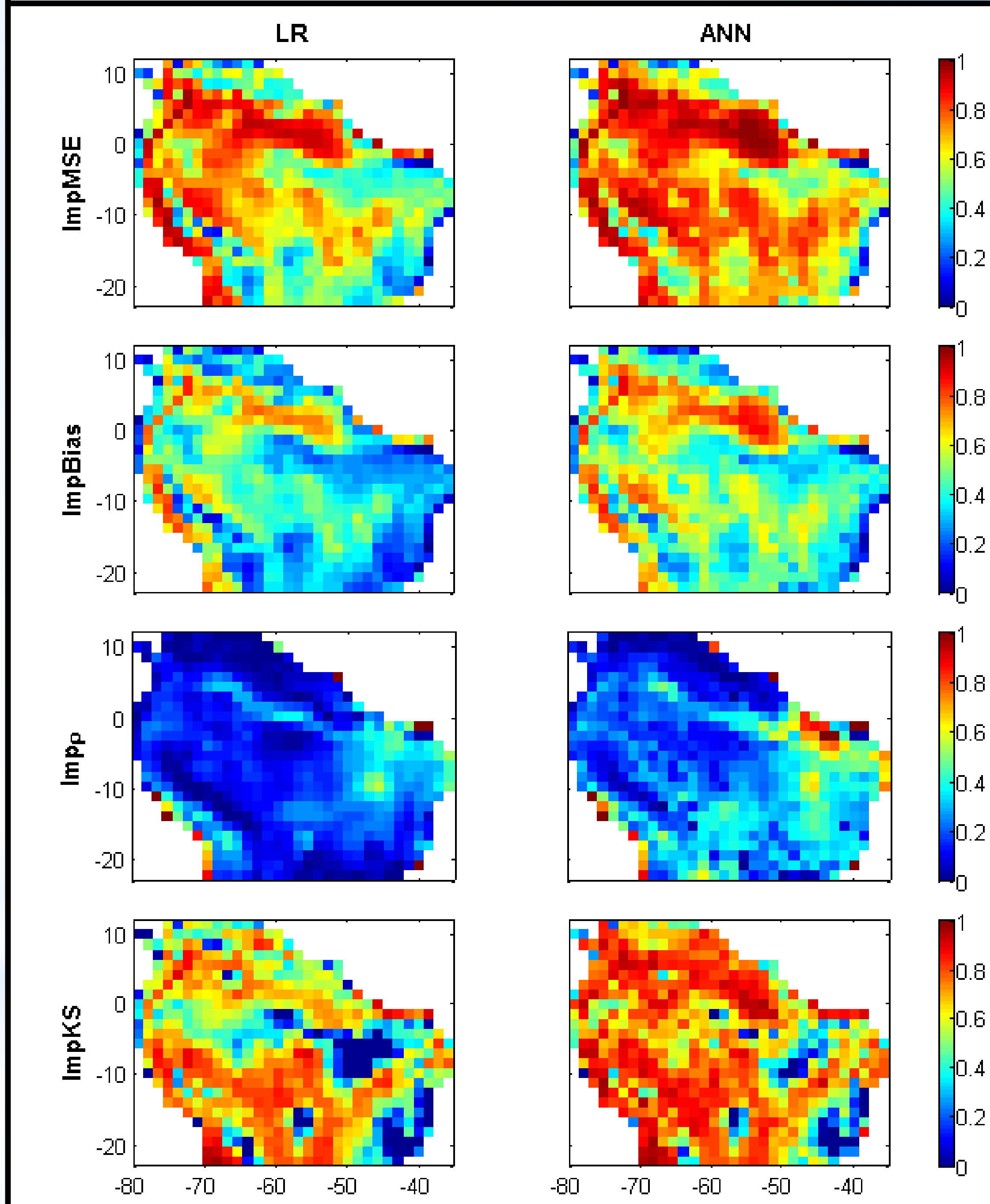


Black Closed Area	LR	ANN
ImpMSE	47%	61%
ImpBias	26%	37%
Imp$\boldsymbol{\rho}$	31%	55%
Imp*KS*	44%	55%

Val



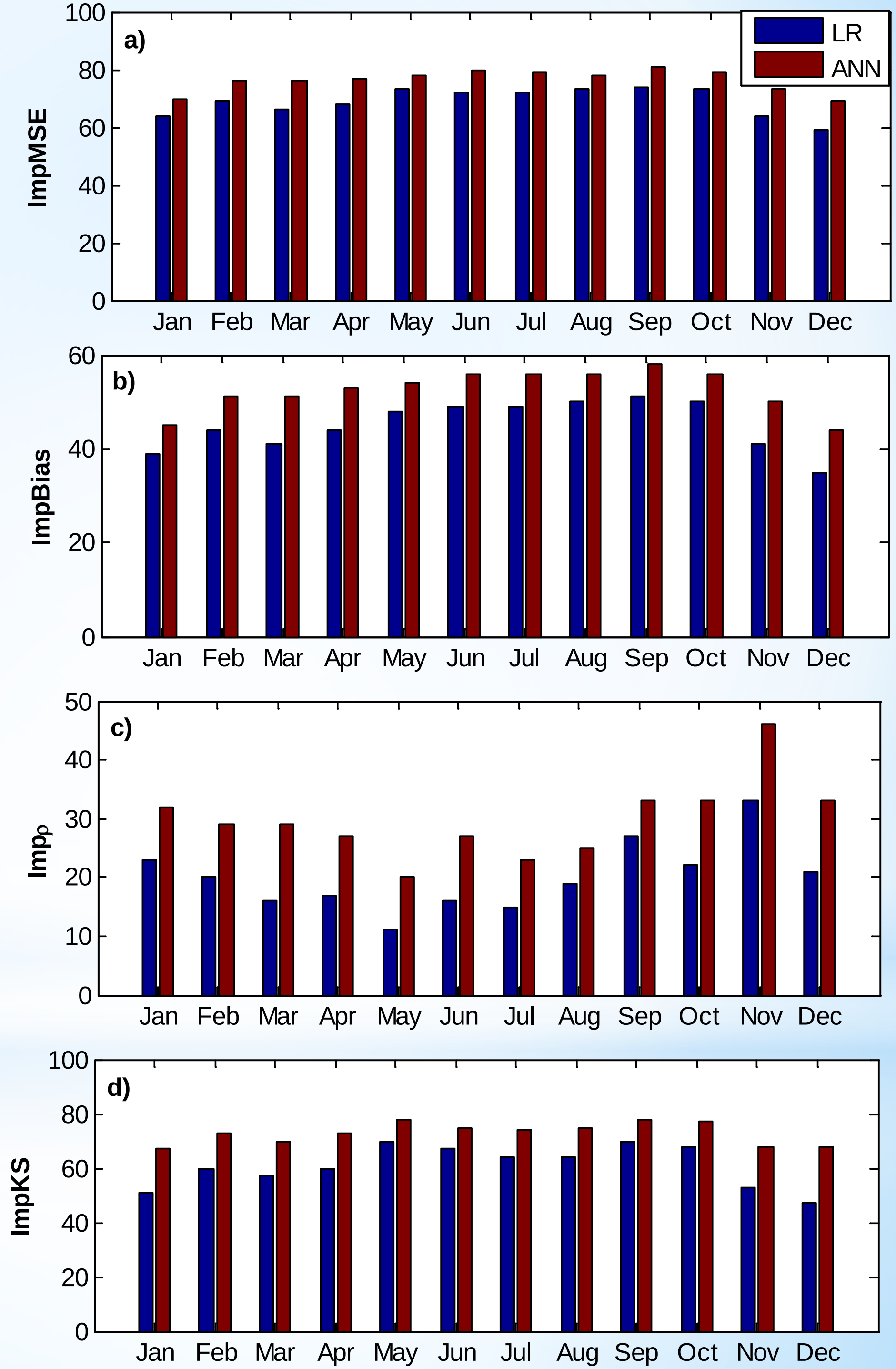
Orig



Val

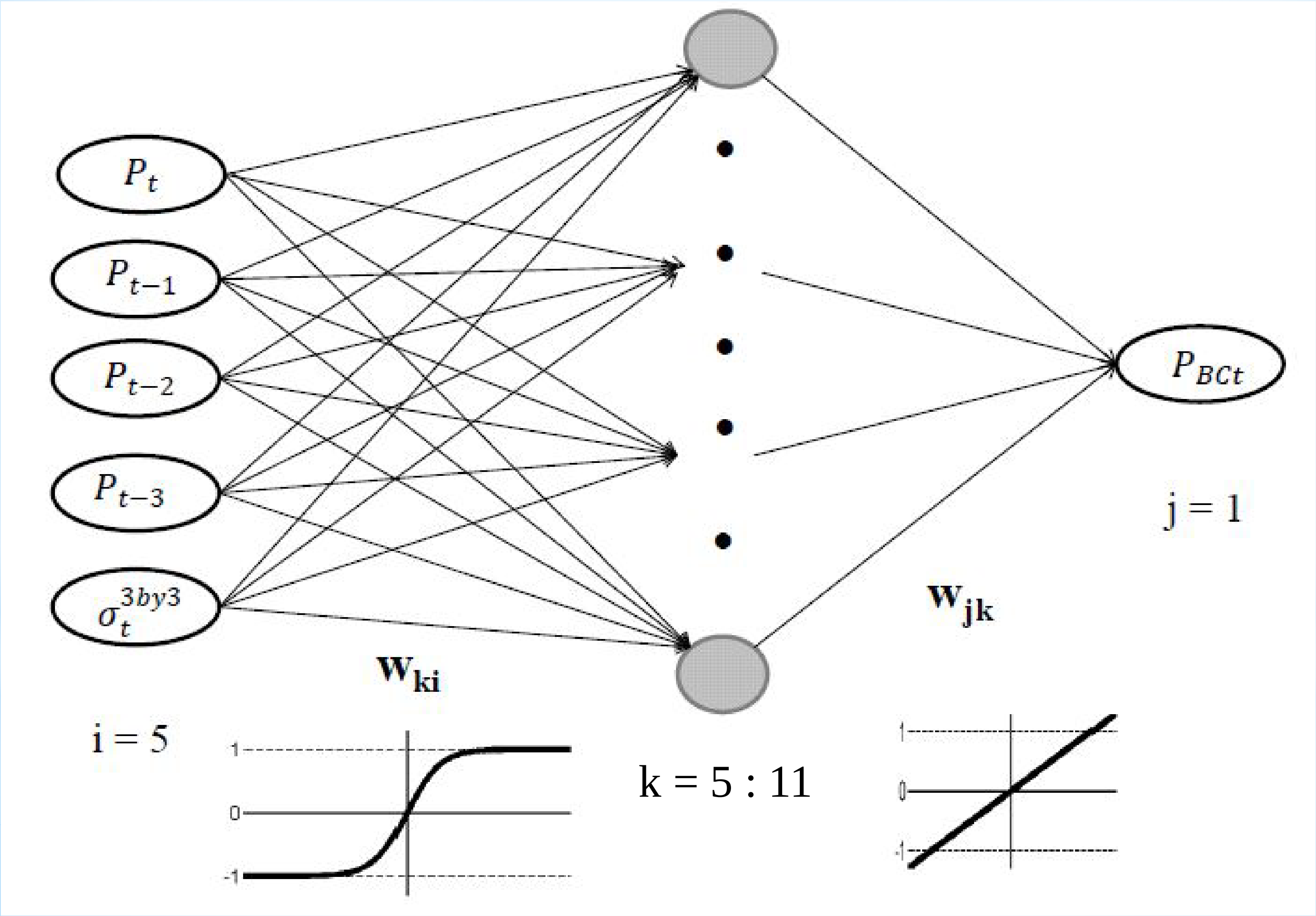
Domain Average Percent Improvements (Imp)

Time-Domain Average Improvement	LR	ANN
ImpMSE	68.92	76.33
ImpBias	45.08	52.5
Impρ	20	29.75
ImpKS	60.92	73



*Precipitation

Diagram of the ANN



Datasets

- Community Climate System Model (CCSM3)

Developed by the University Corporation for Atmospheric Research (UCAR).

Spatial Resolution : 1.4° (T85)

Scenarios : 20C3M, A2

Temporal Coverage : 1990-2099

- Climate Research Unit (CRU)

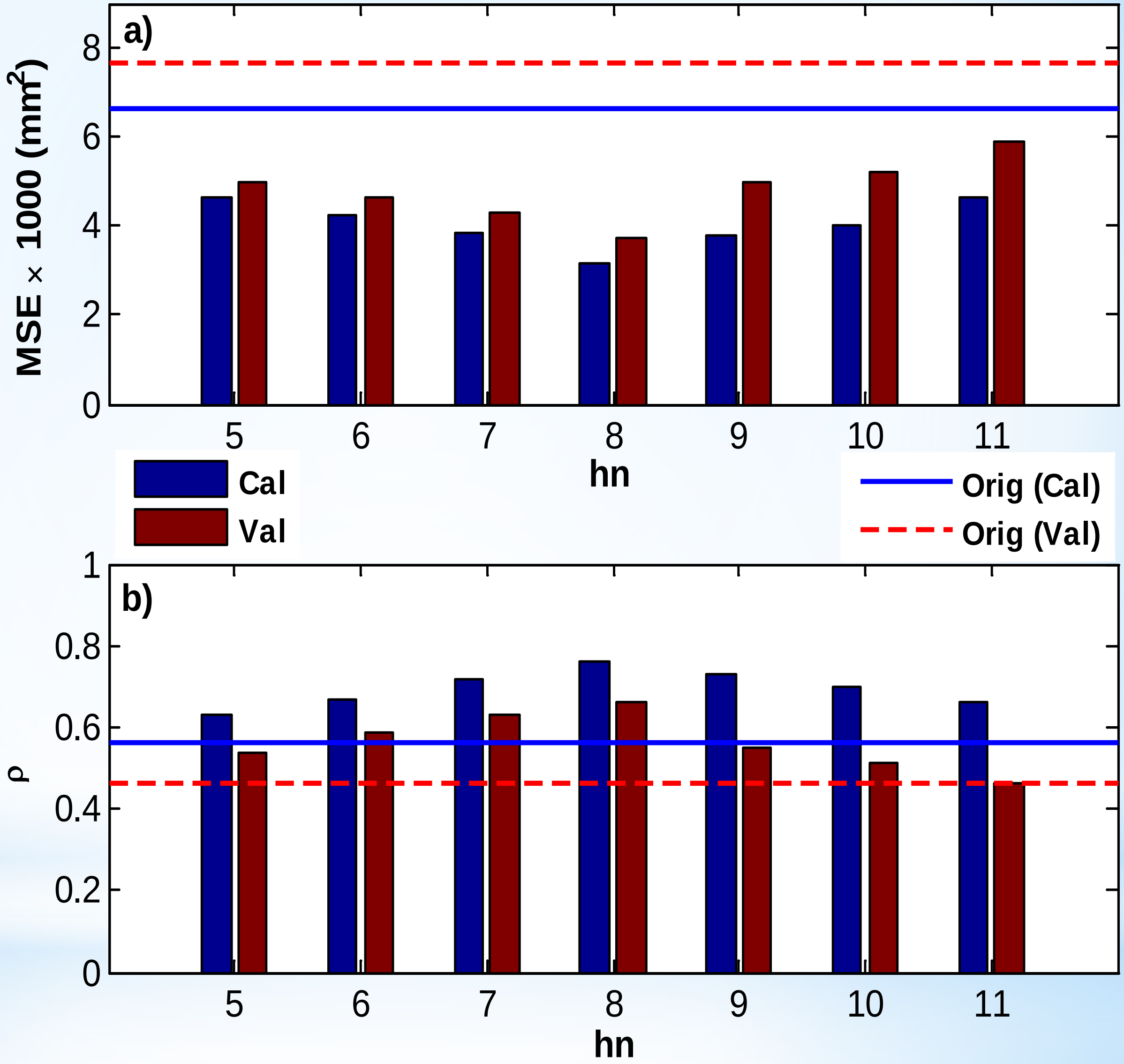
Developed by the University of the East Anglia, United Kingdom (Jones and Harris 2014)

Spatial Resolution : 0.5°

Temporal Coverage : 1901-2013

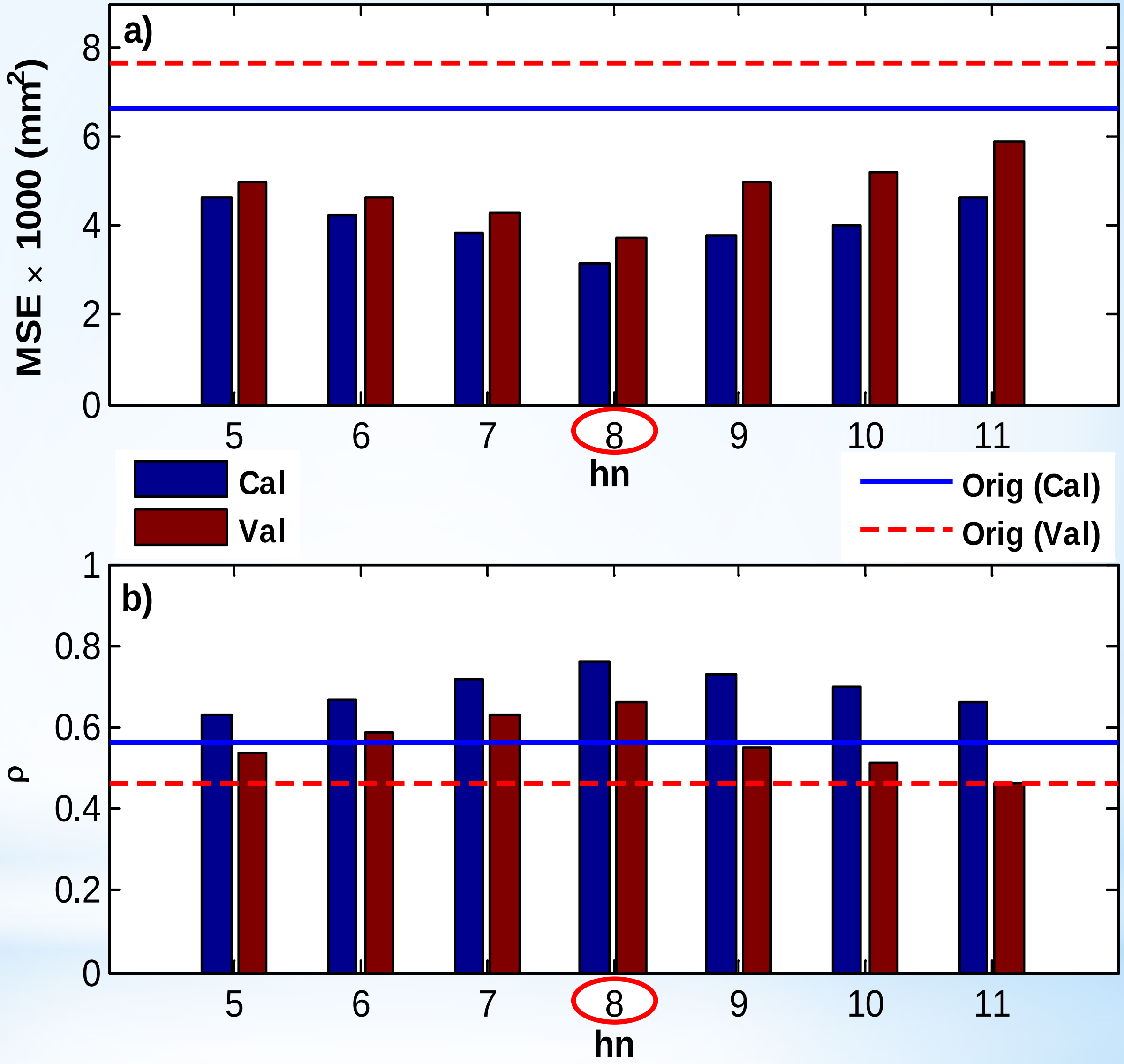
$\eta = 0.01$

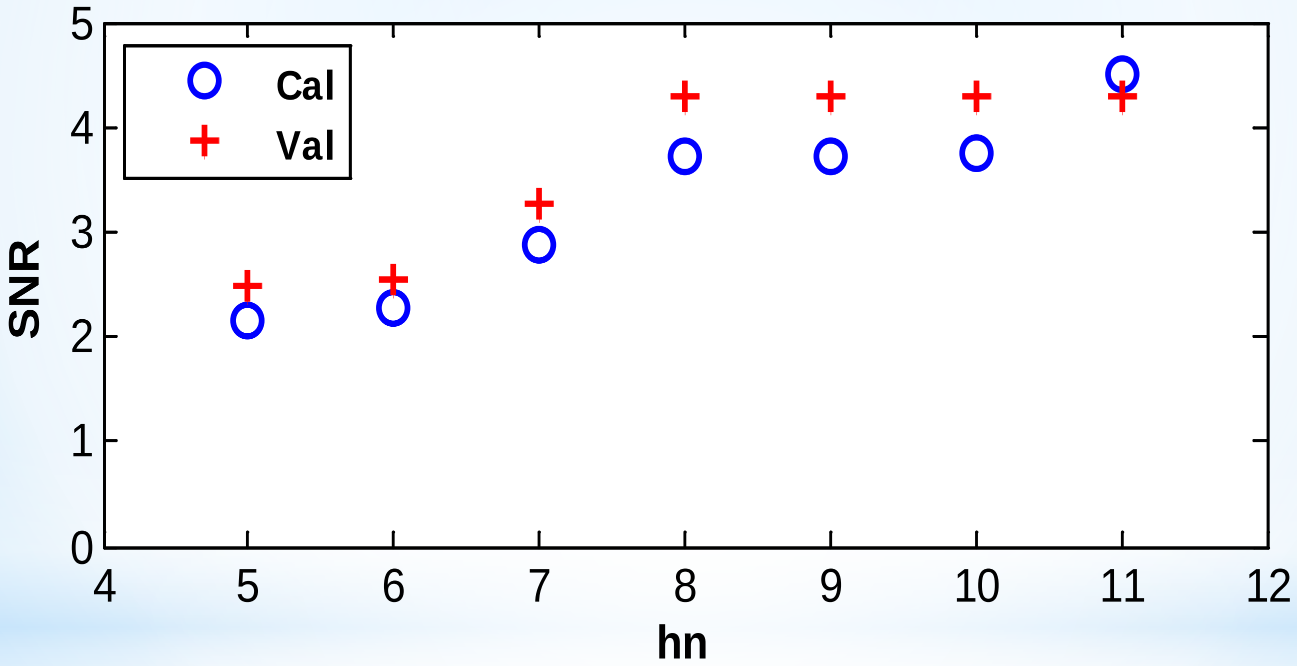
Orig : precipitation before bias correction and target
Cal: calibration
Val: Validation



$\eta = 0.01$

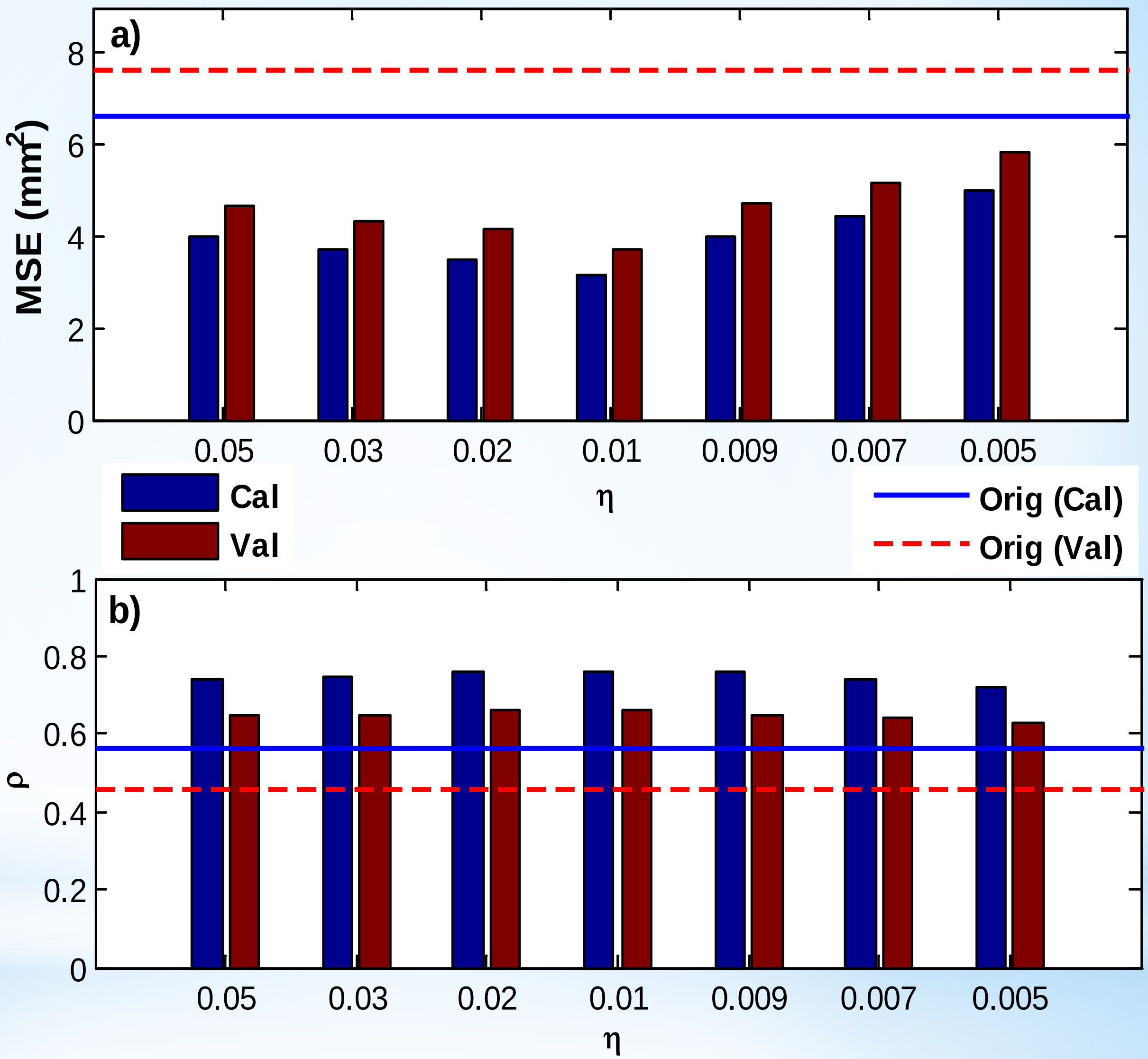
Orig : precipitation before bias correction and target



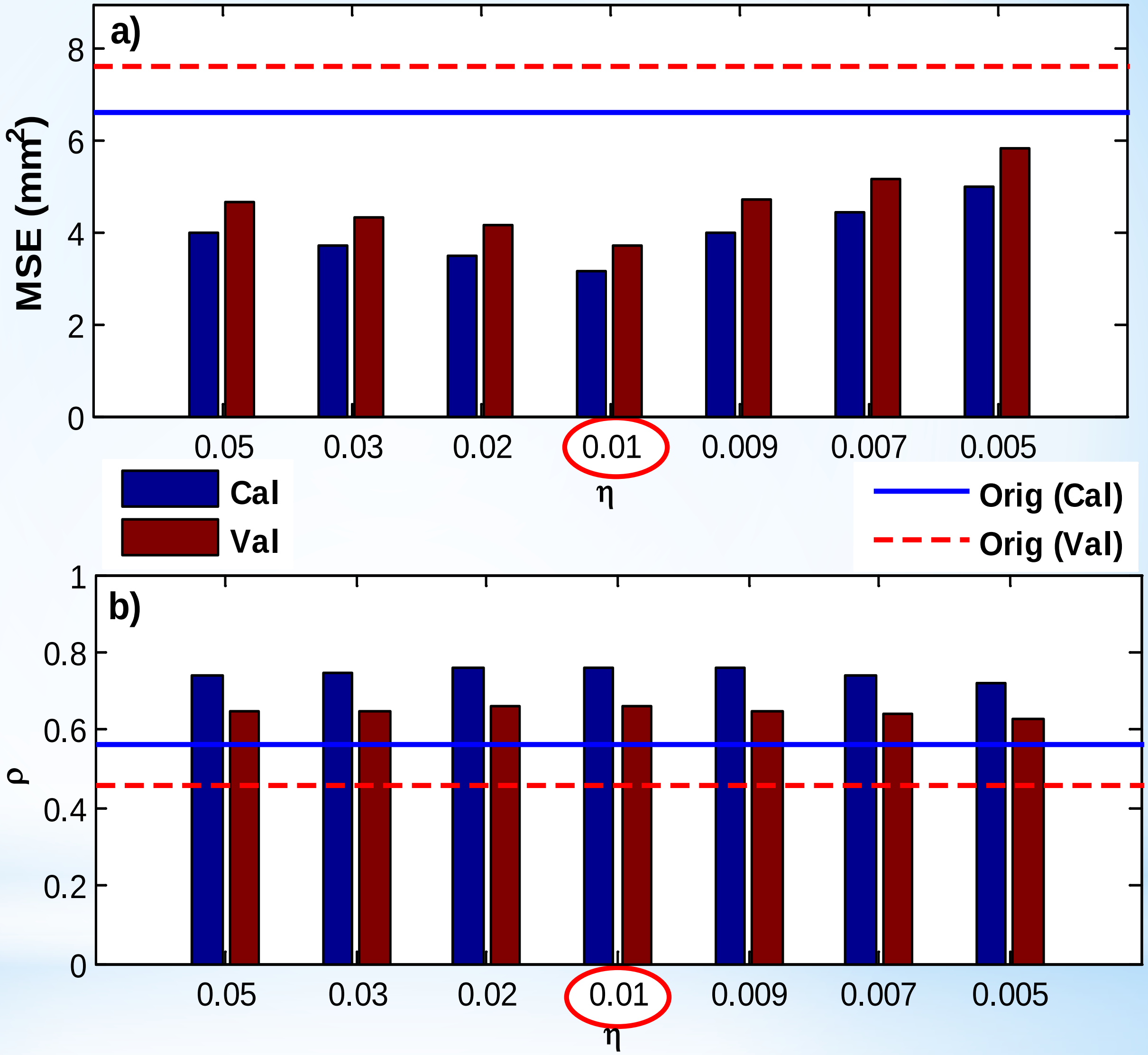

Cal
Val
SNR
hn
0
1
2
3
4
5
4
5
6
7
8
9
10
11
12

$hn = 8$

Orig : precipitation before bias correction and target



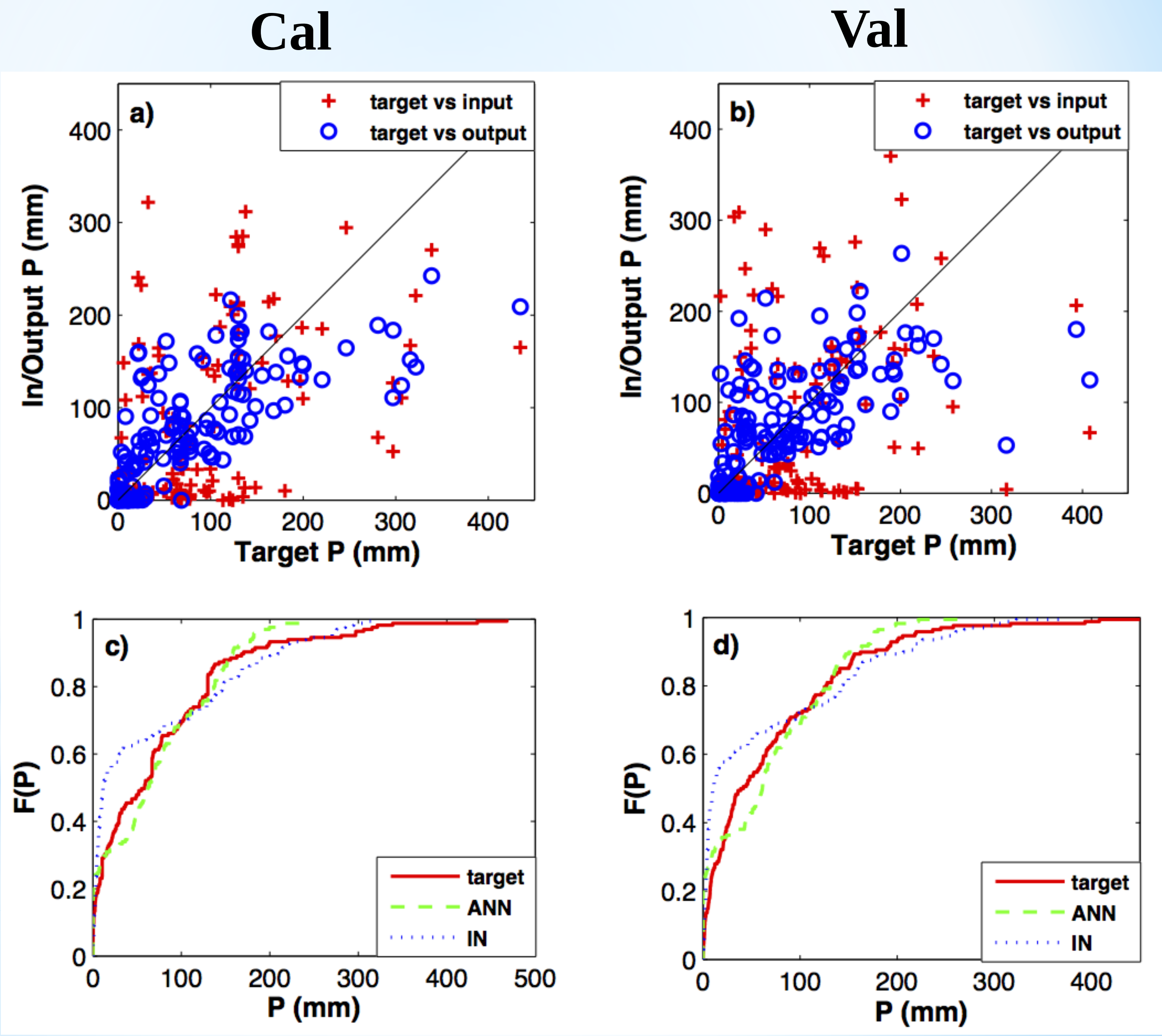
$hn = 8$



Orig : precipitation before bias correction and target

Performance of the ANN

$$hn = 8$$
$$\eta = 0.01$$

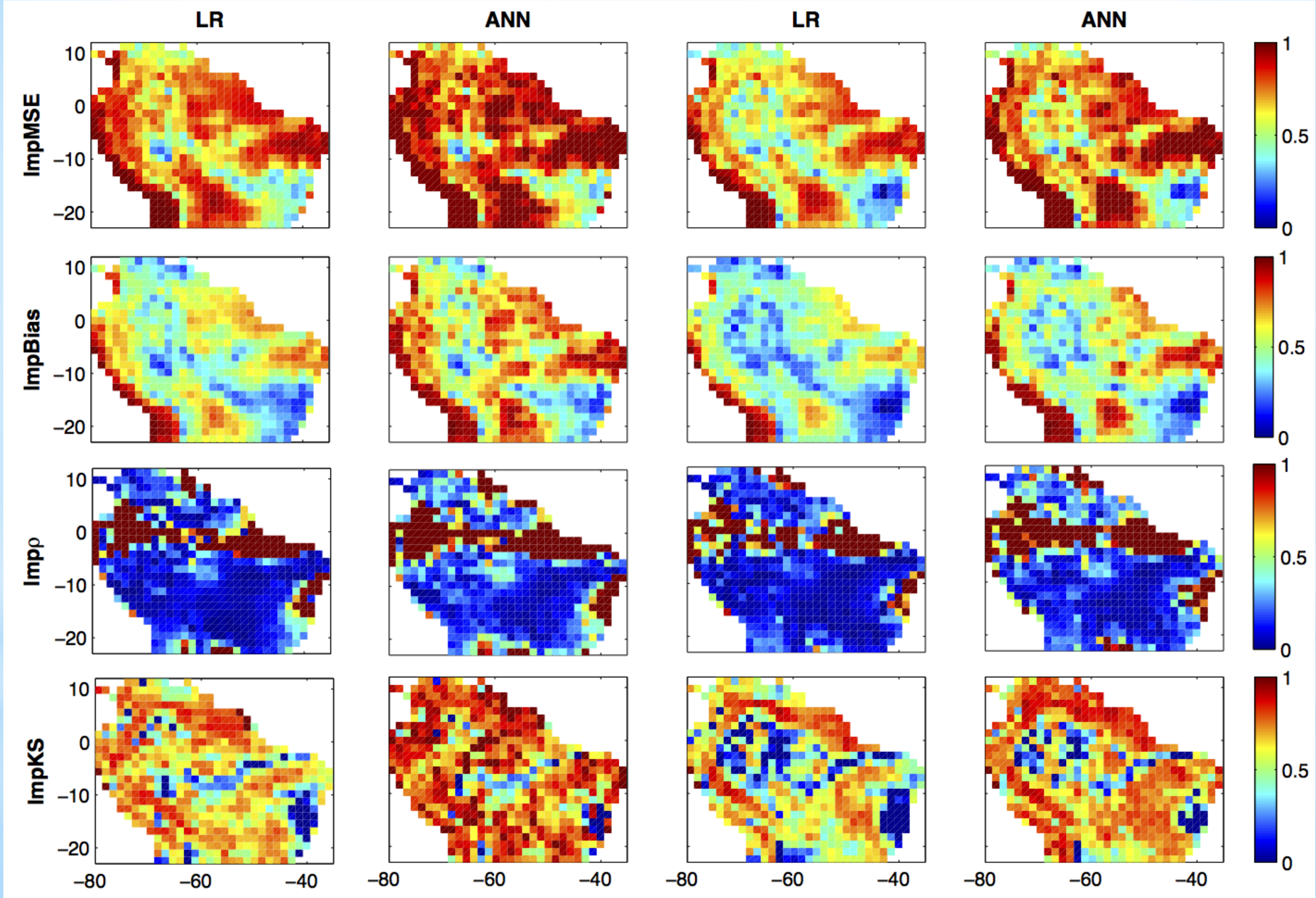


P	MSE		Bias		*ρ*		*KS*	
	Cal	**Val**	**Cal**	**Val**	**Cal**	**Val**	**Cal**	**Val**
In	6609	7652	55	56	0.56	0.46	0.24	0.27
CDF	7459	8532	54.97	58.28	0.51	0.40	0.06	0.09
EDCDF	7337	8695	54.53	57.99	0.51	0.40	0.05	0.07
LR	4130	4910	42.17	43.54	0.67	0.54	0.12	0.12
ANN	3166	3711	36.09	37.65	0.76	0.66	0.10	0.12

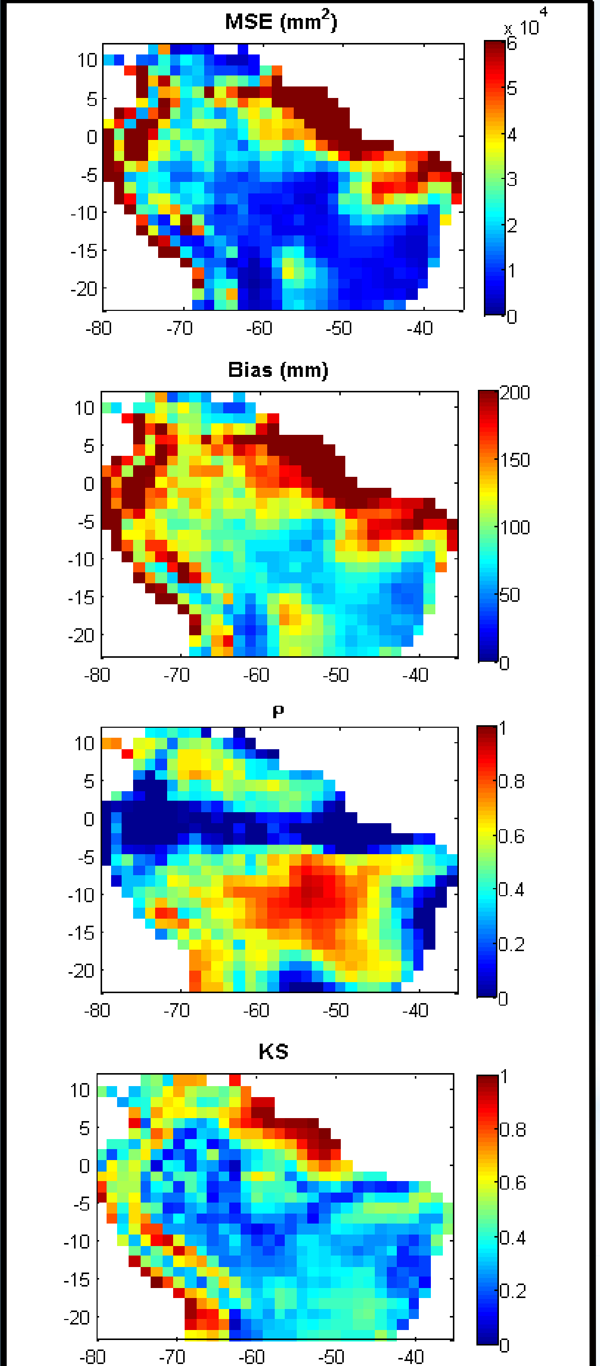
P	MSE		Bias		*ρ*		*KS*	
	Cal	**Val**	**Cal**	**Val**	**Cal**	**Val**	**Cal**	**Val**
In	6609	7652	55	56	0.56	0.46	0.24	0.27
CDF	7459	8532	54.97	58.28	0.51	0.40	0.06	0.09
EDCDF	7337	8695	54.53	57.99	0.51	0.40	0.05	0.07
LR	**37%**	**36%**	42.17	43.54	**20%**	**17%**	0.12	0.12
ANN	**52%**	**51%**	36.09	37.65	**36%**	**43%**	0.10	0.12

Pixel By Pixel Correction

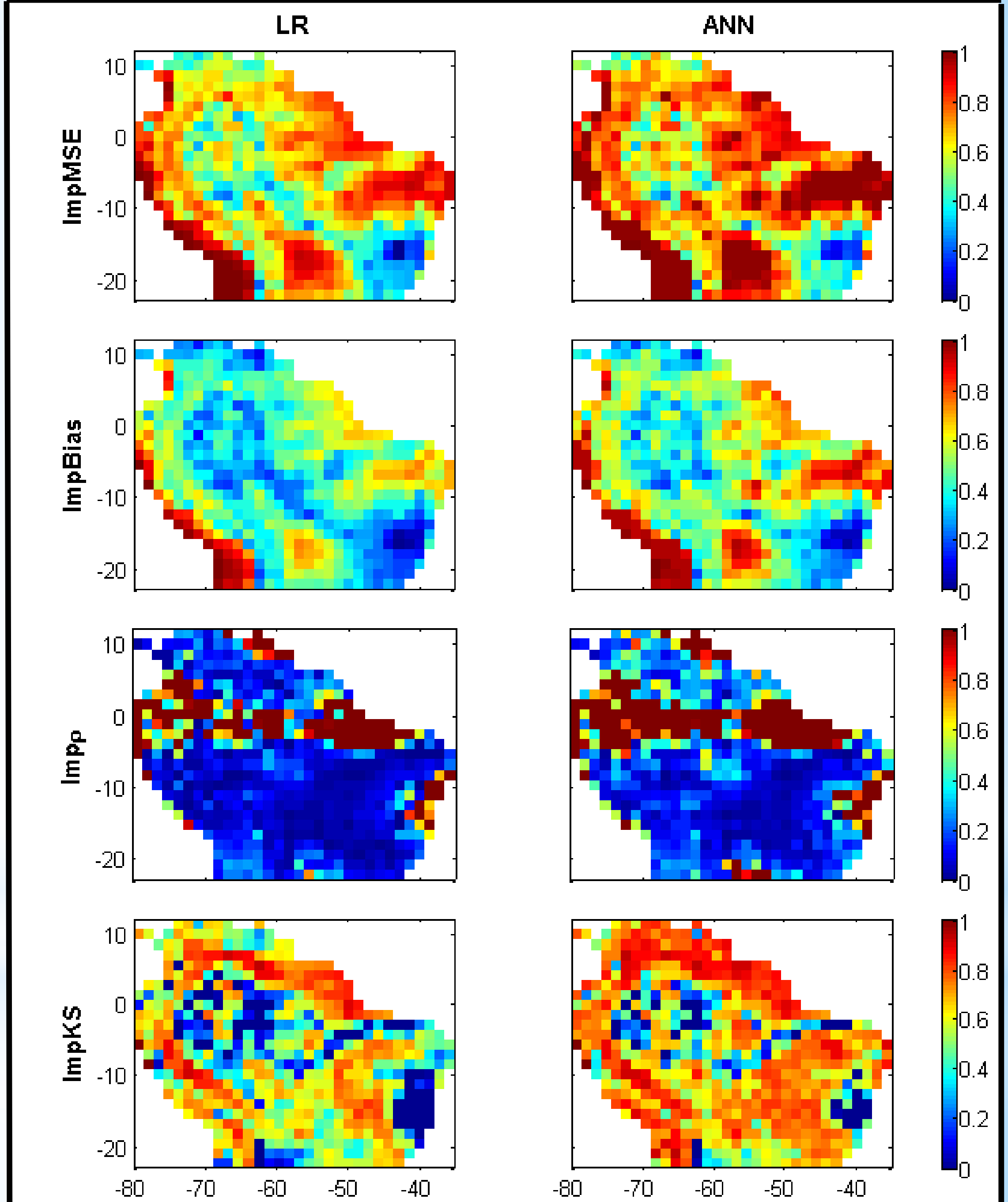
Improvement Ratio (Imp)



Cal **Val**



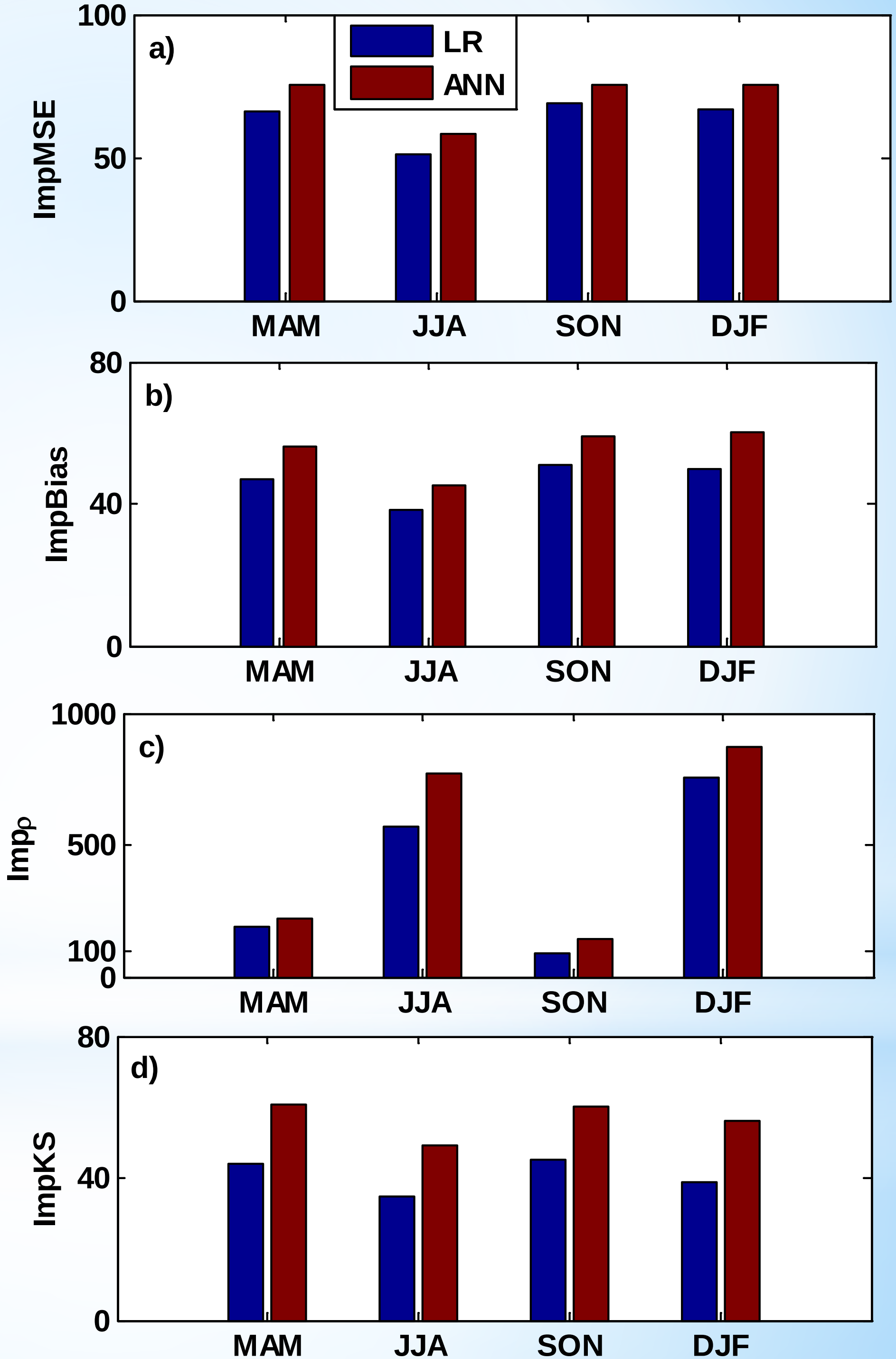
Orig



Val

➤ Domain Average Percent Improvements (Imp)

Time-Domain Average Improvement	**LR**	**ANN**
ImpMSE	63.25	70.75
ImpBias	46.5	55
Impρ	402	501
ImpKS	40.75	56.5



- ✓ The trained ANN model performs well for both calibration and validation periods.

- ✓ The trained ANN model performs well for both calibration and validation periods.

Temporal Generalization √

Temporal Generalization √

- ✓ The trained ANN model performs well for both calibration and validation periods.

➢ QUESTION

Is the ANN model useful over locations where observations are not available?

Temporal Generalization √

- ✓ The trained ANN model performs well for both calibration and validation periods.

➢ QUESTION

Is the ANN model useful over locations where observations are not available?

Spatial Generalization ?

Regionalization

Goal of Regionalization

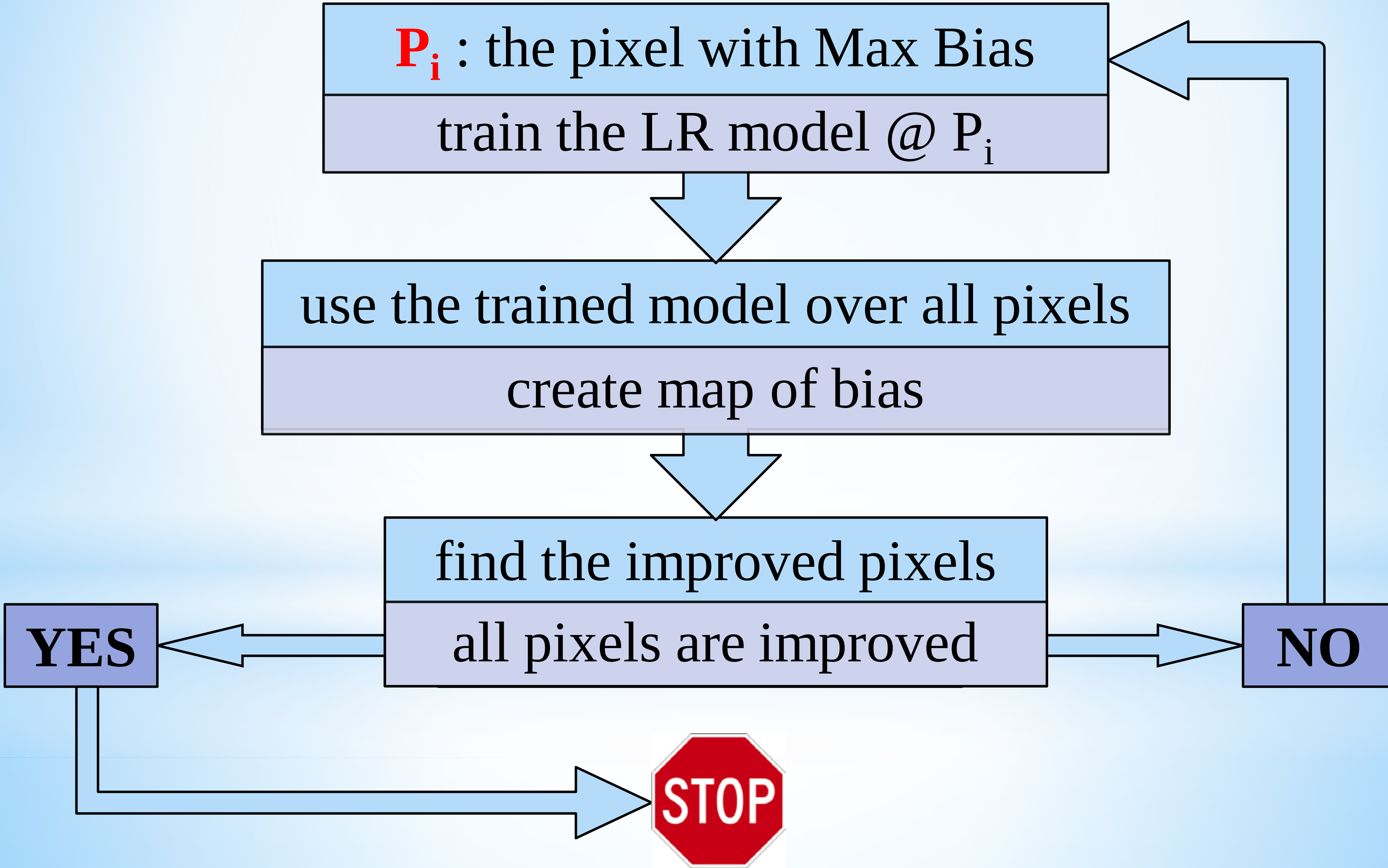
Identify a finite number of training pixels (calibrations) that would result in a desired response (sufficiently close to the performance of the model calibrated at all domain pixels).

Criteria:
$$\begin{cases} 1.3 \times Bias_{PbyP} & For\ 70\%\ performance \\ 1.2 \times Bias_{PbyP} & For\ 80\%\ performance \\ 1.1 \times Bias_{PbyP} & For\ 90\%\ performance \end{cases}$$

$Bias_{PbyP}$: bias between the bias-corrected temperature and the target, when the biases are corrected pixel by pixel.

Pixel with Bias < Criteria ⟶ Improved Pixel

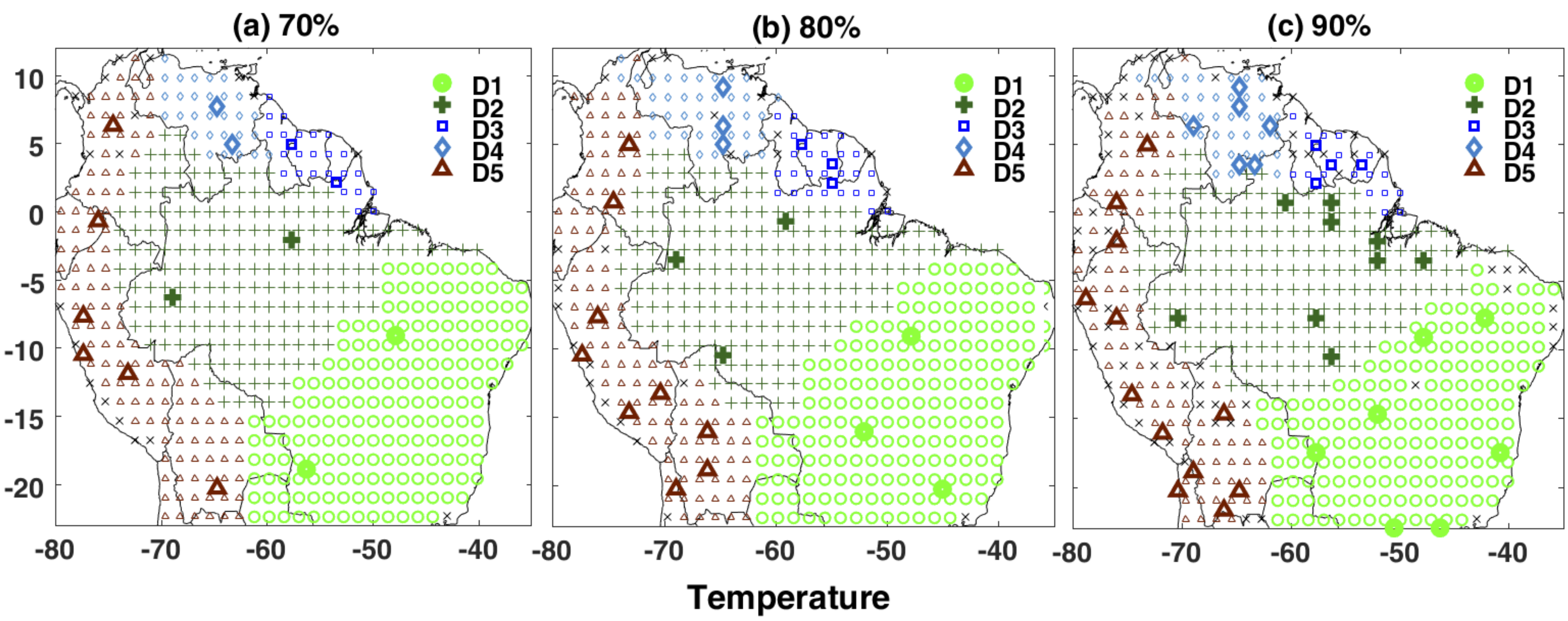
Diagram of the Process



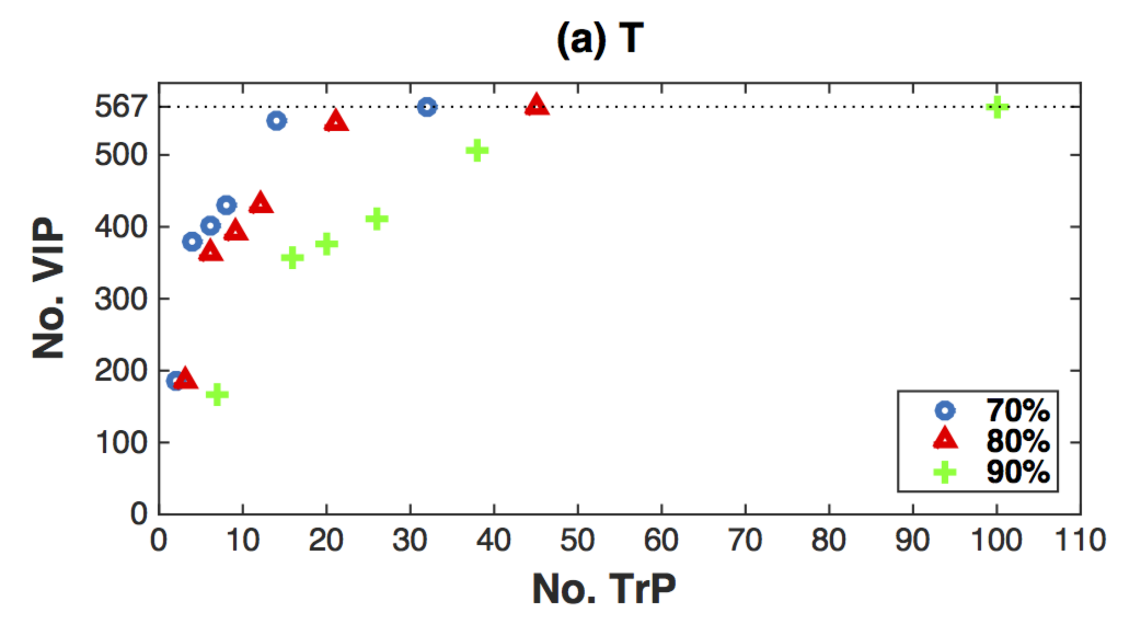
*Temperature

Smaller Marker Symbol: Validating Pixel (VlP)
Larger Marker Symbol: Training Pixels (TrP)
Black Cross Symbol: Independent Training Pixel (IndTrP)

March

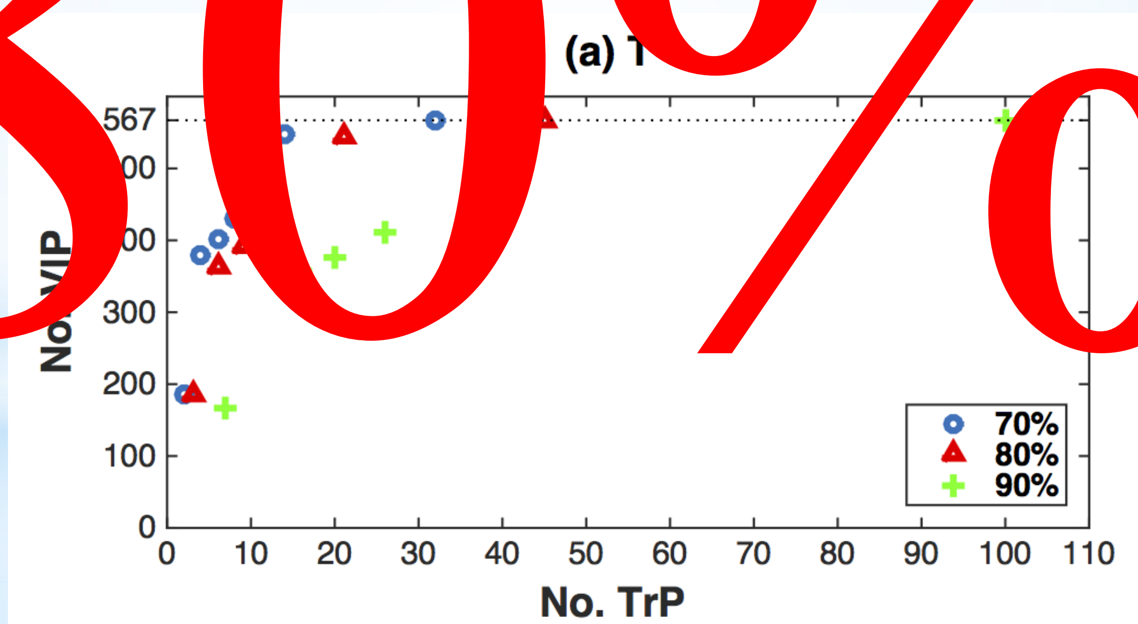


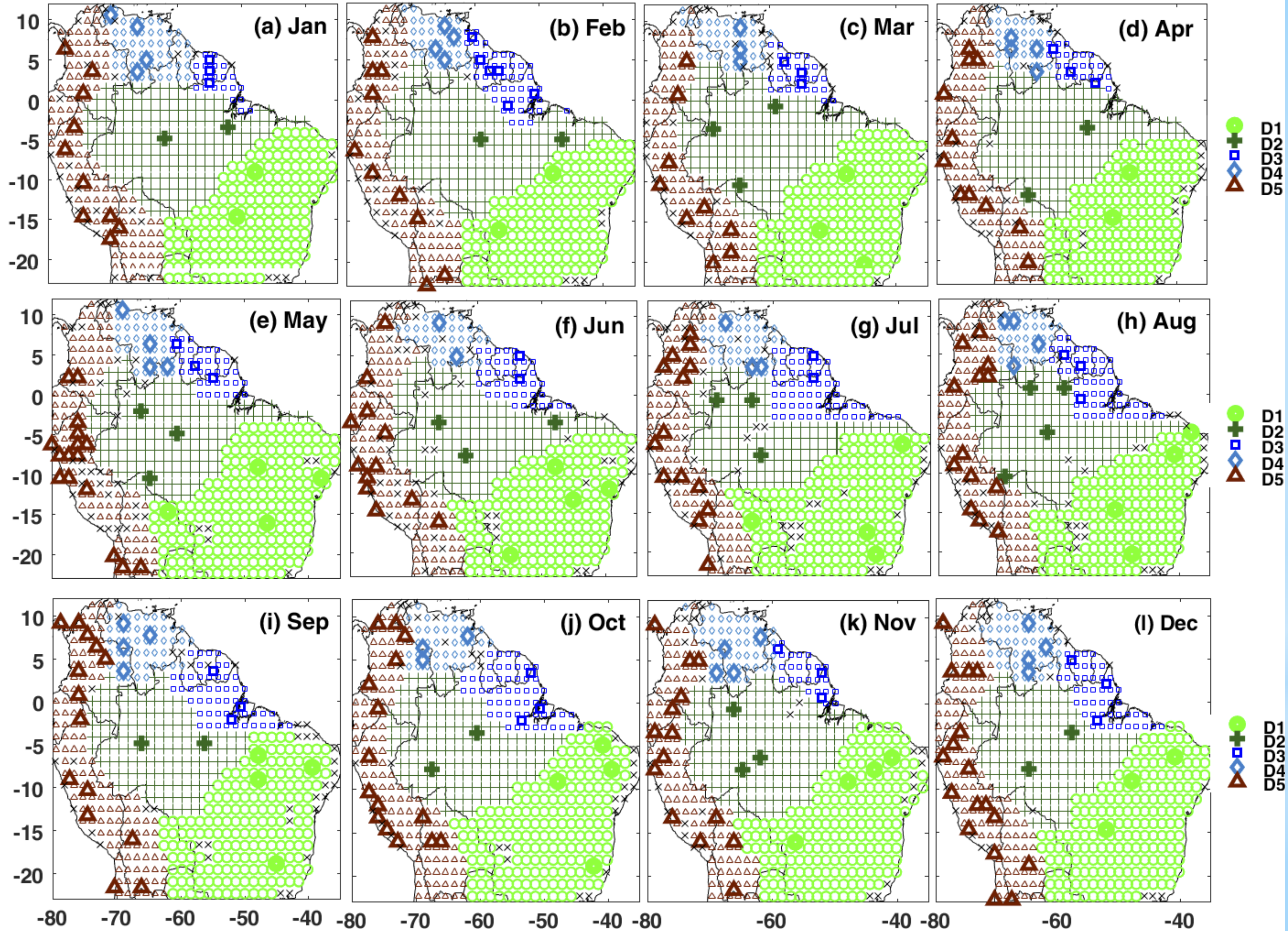
	D1	D2	D3	D4	D5	IndTrP
70%	2	2	2	2	6	21
80%	3	3	3	3	9	27
90%	7	9	4	6	12	63



	D1	D2	D3	D4	D5	IndTrP
70%	2	2	2	2	6	35
80%	3	3	3	3	9	48
90%	7	[illegible]	4	[illegible]	12	[illegible]

80%





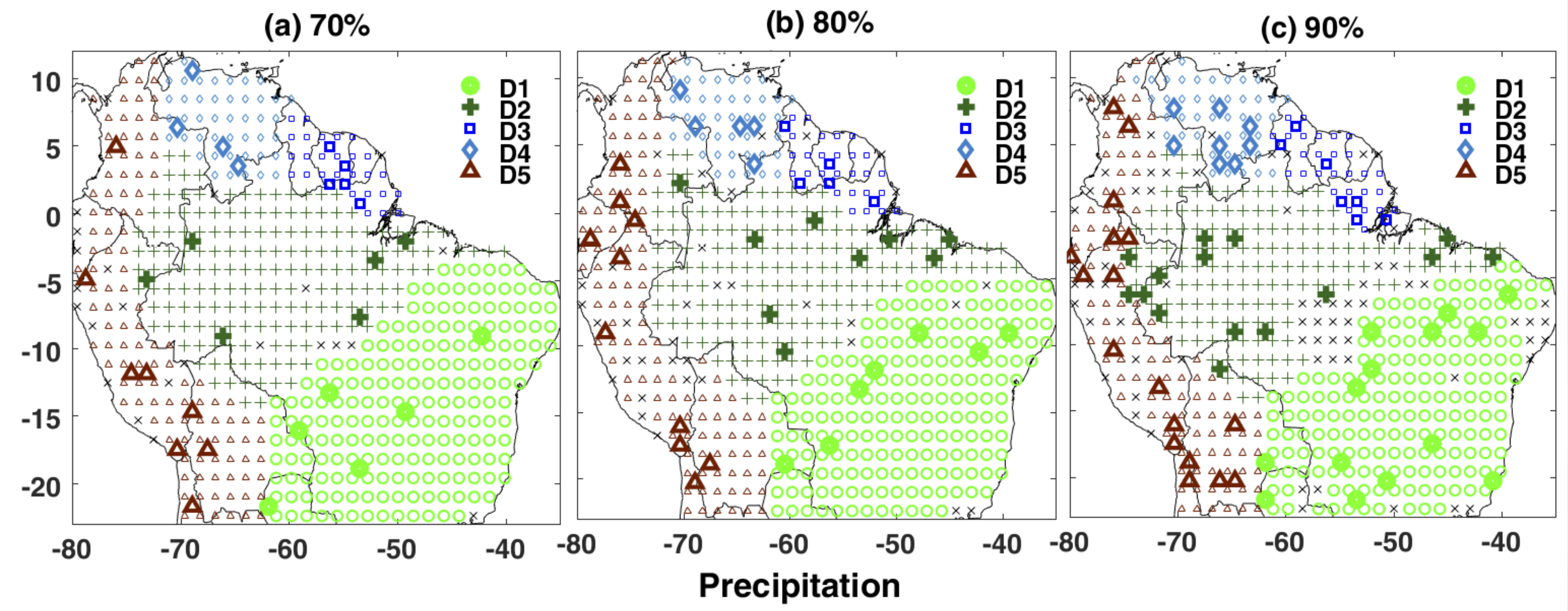
(a) Jan
(b) Feb
(c) Mar
(d) Apr
(e) May
(f) Jun
(g) Jul
(h) Aug
(i) Sep
(j) Oct
(k) Nov
(l) Dec
D1
D2
D3
D4
D5
10
5
0
-5
-10
-15
-20
-80
-70
-60
-50
-40

	Jan			Feb			Mar			Apr		
	TrP	VlP	IndTrP	TrP	VlP	IndTrP	TrP	VlP	IndTrP	TrP	VlP	IndTrP
D1	2	175	4	2	164	5	3	186	2	2	169	11
D2	2	175	0	2	184	2	3	179	0	2	187	1
D3	3	21	2	6	46	0	3	27	4	3	23	3
D4	4	49	4	4	32	0	3	37	5	4	31	3
D5	10	121	16	10	118	18	9	114	13	11	127	12
	May			Jun			Jul			Aug		
	TrP	VlP	IndTrP	TrP	VlP	IndTrP	TrP	VlP	IndTrP	TrP	VlP	IndTrP
D1	4	201	7	4	172	14	4	183	16	4	166	15
D2	3	155	4	3	161	6	3	130	5	4	160	5
D3	3	34	1	2	40	0	2	60	0	3	54	3
D4	4	36	1	2	34	4	3	40	2	4	32	1
D5	16	104	25	12	114	22	14	115	16	13	116	15
	Sep			Oct			Nov			Dec		
	TrP	VlP	IndTrP	TrP	VlP	IndTrP	TrP	VlP	IndTrP	TrP	VlP	IndTrP
D1	4	176	19	4	180	15	4	190	9	3	184	4
D2	2	134	0	2	145	0	3	167	2	2	149	1
D3	3	51	5	3	55	0	3	23	0	3	45	1
D4	4	46	3	3	37	7	4	49	2	4	46	5
D5	14	124	9	16	120	8	14	113	12	17	123	9

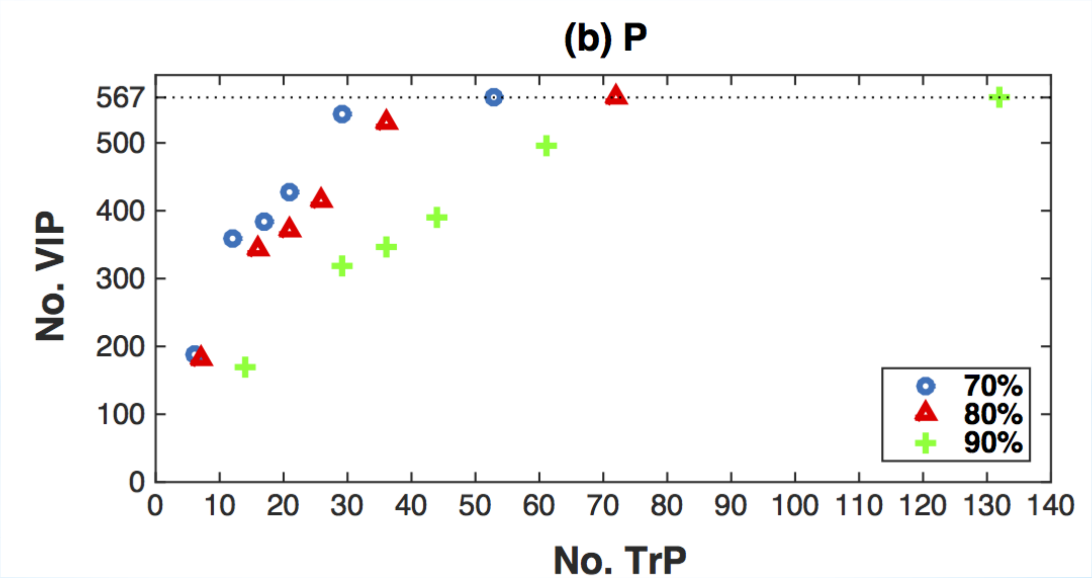
*Precipitation

Smaller Marker Symbol: Validating Pixel (VlP)
Larger Marker Symbol: Training Pixels (TrP)
Black Cross Symbol: Independent Training Pixel (IndTrP)

MAM

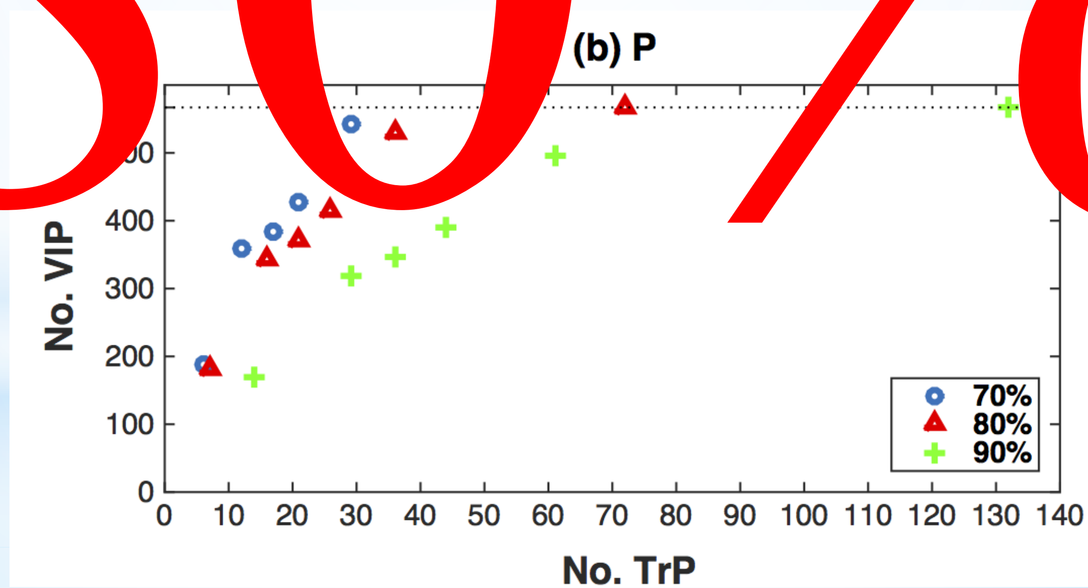


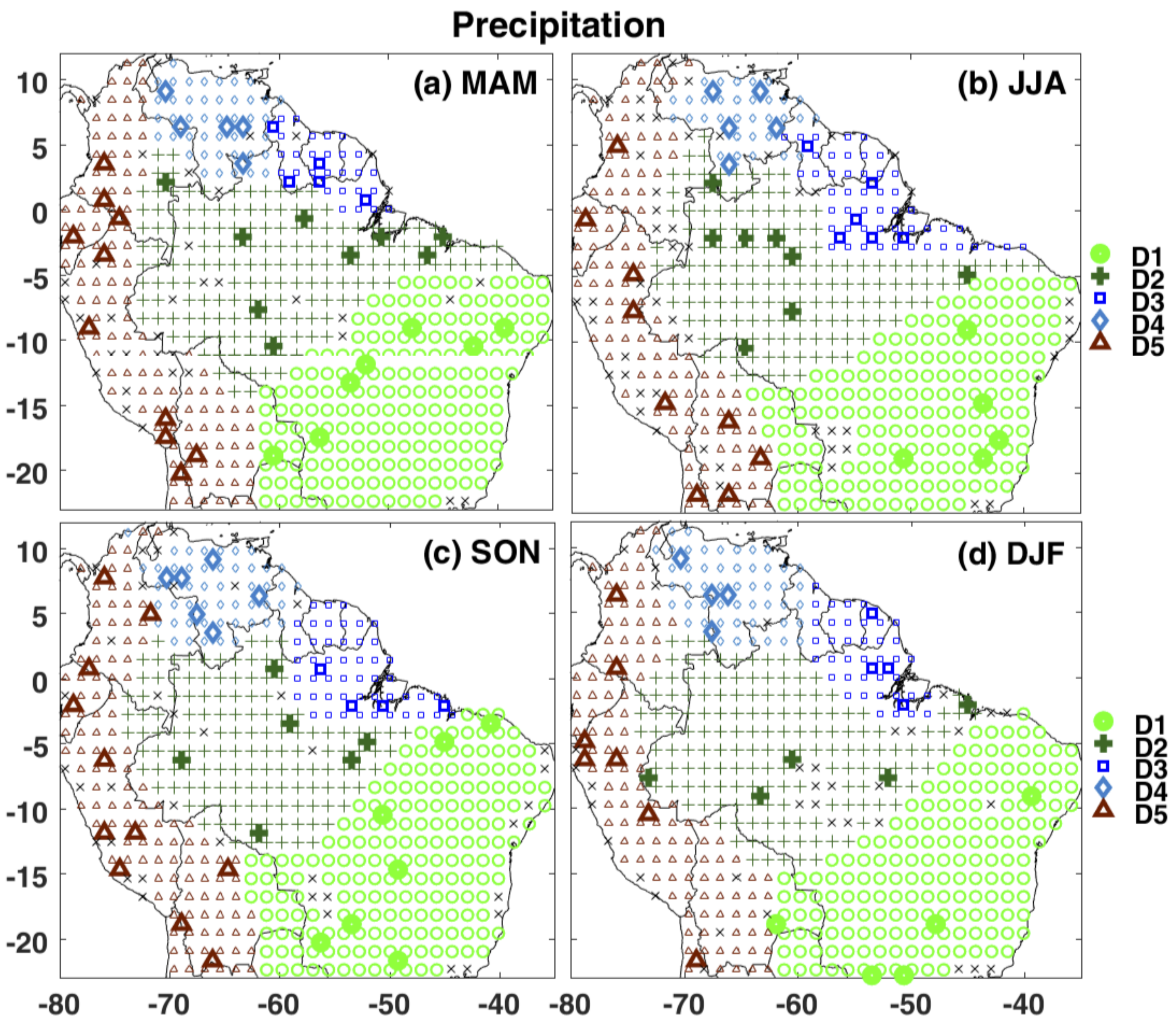
	D1	D2	D3	D4	D5	IndTrP
70%	6	6	5	4	8	26
80%	7	9	5	5	10	38
90%	14	15	7	8	17	74



	D1	D2	D3	D4	D5	IndTrP
70%	6	6	5	4	8	26
80%	7	9	[illegible]	5	[illegible]	38
[illegible]	[illegible]	[illegible]	7	8	17	74

80%





Precipitation
(a) MAM
(b) JJA
(c) SON
(d) DJF
D1
D2
D3
D4
D5
10
5
0
-5
-10
-15
-20
-80
-70
-60
-50
-40

	MAM			JJA			SON			DJF		
	TrP	VlP	IndTrP	TrP	VlP	IndTrP	TrP	VlP	IndTrP	TrP	VlP	IndTrP
D1	7	184	6	5	169	12	7	182	12	5	171	6
D2	9	162	8	8	148	10	6	134	7	5	155	14
D3	5	27	3	6	50	3	4	49	1	4	39	1
D4	5	43	2	5	32	3	6	48	5	4	50	2
D5	10	116	17	9	117	23	11	117	12	7	118	13

Physical Origins of Regionalization

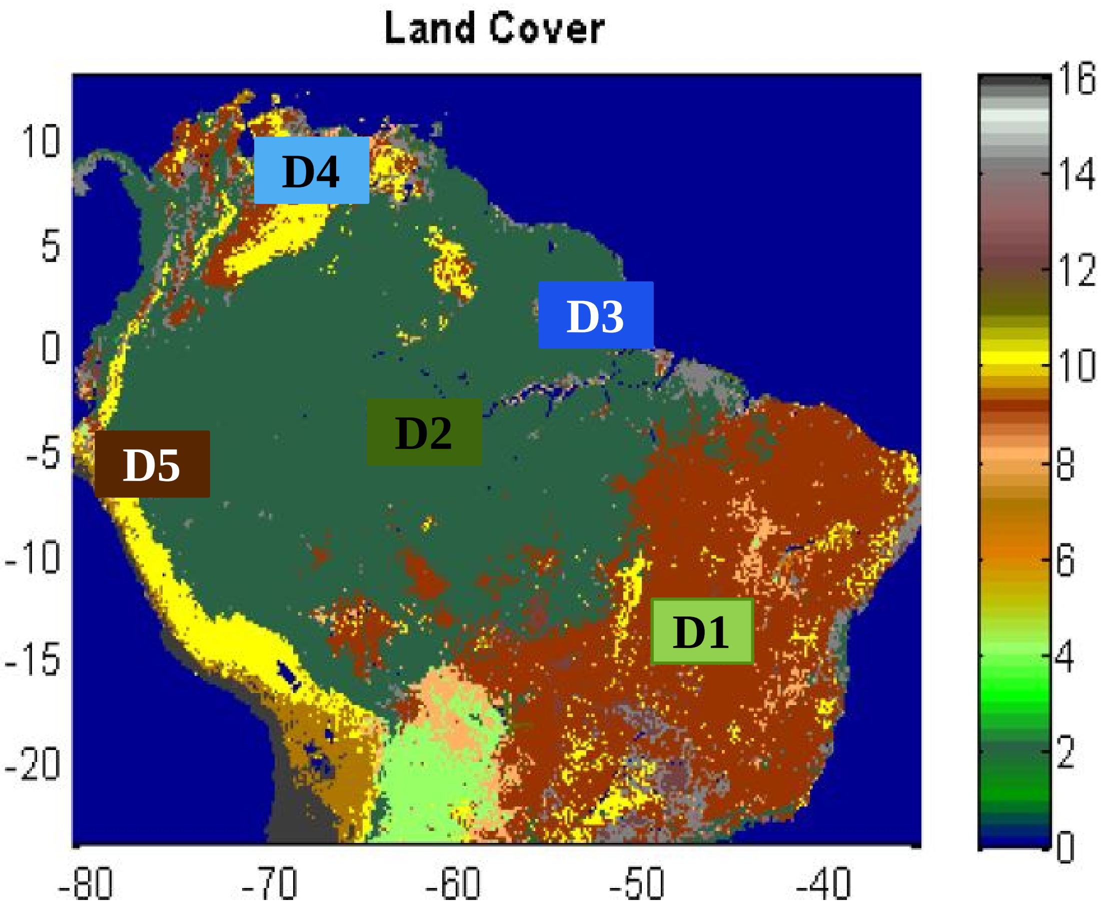
- Investigate consistency and relationship between delineated domains and physical features of the domain such as:

Physical Origins of Regionalization

- Investigate consistency and relationship between delineated domains and physical features of the domain such as:

  - **Land Cover Type**
  - **Elevation**
  - **Climatology**
    - **Temperature**
    - **Precipitation**

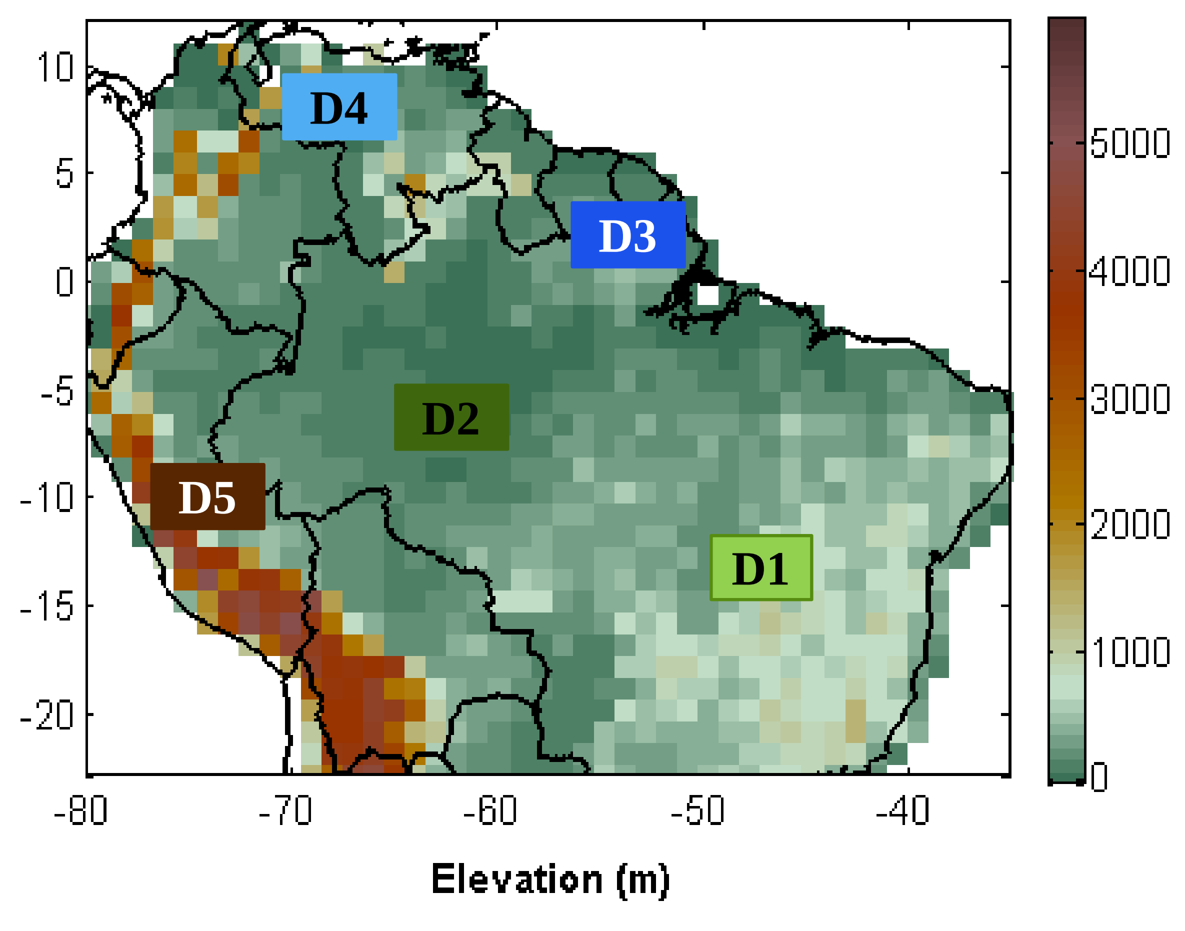
• Land Cover Type



Group	Class
16	barren or sparsely vegetated
15	snow and ice
14	cropland/natural vegetation mosaic
13	urban and built-up
12	croplands
11	permanent wetlands
10	grasslands
9	savannas
8	woody savannas
7	open shrub lands
6	closed shrub land
5	mixed forests
4	deciduous broadleaf forest
3	deciduous needleleaf forest
2	evergreen broadleaf forest
1	evergreen needleleaf forest
0	water

Data: Land Cover Type Climate Modeling Grid (CMG) product derived from Terra and Aqua MODIS data

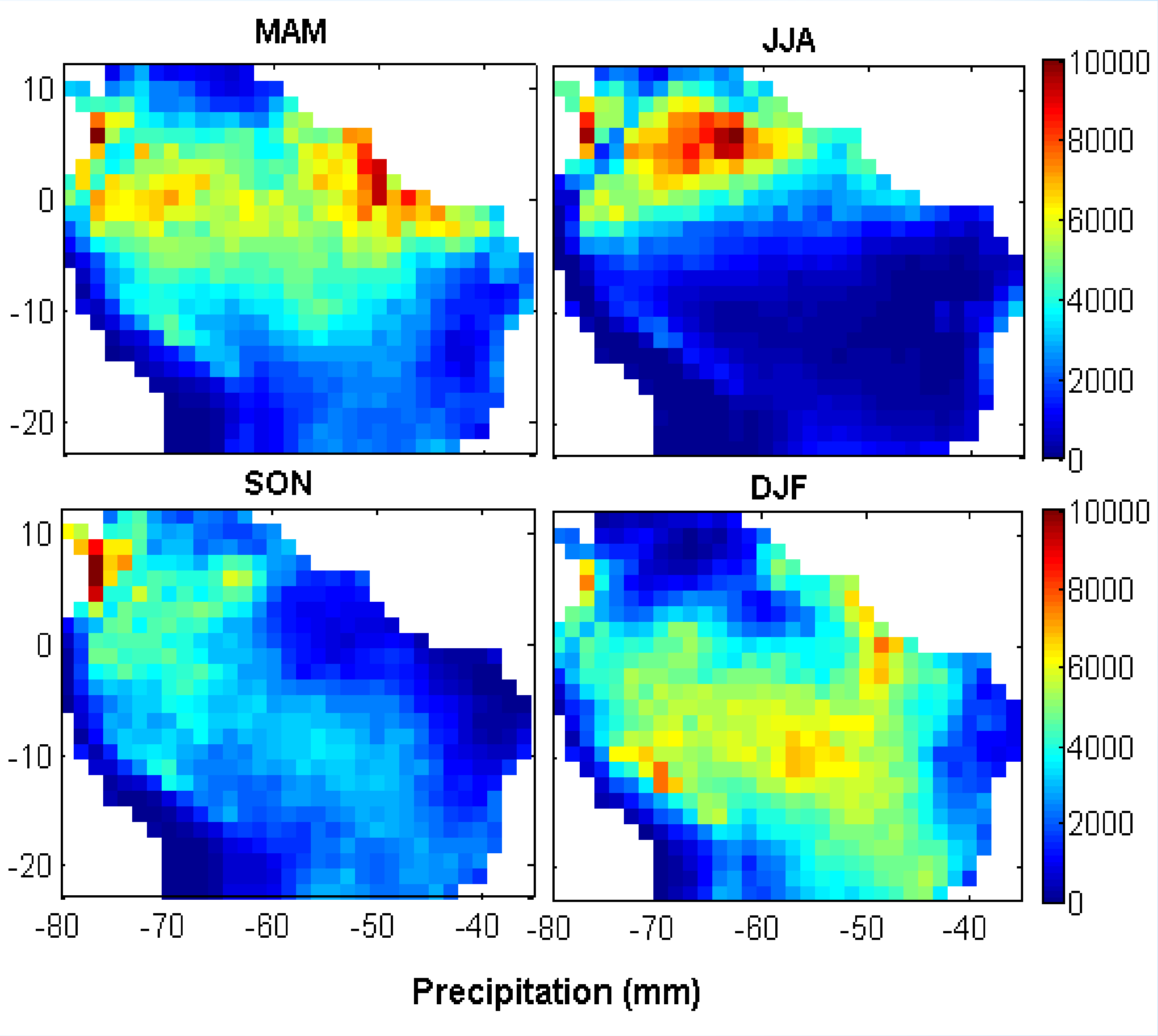
• Elevation



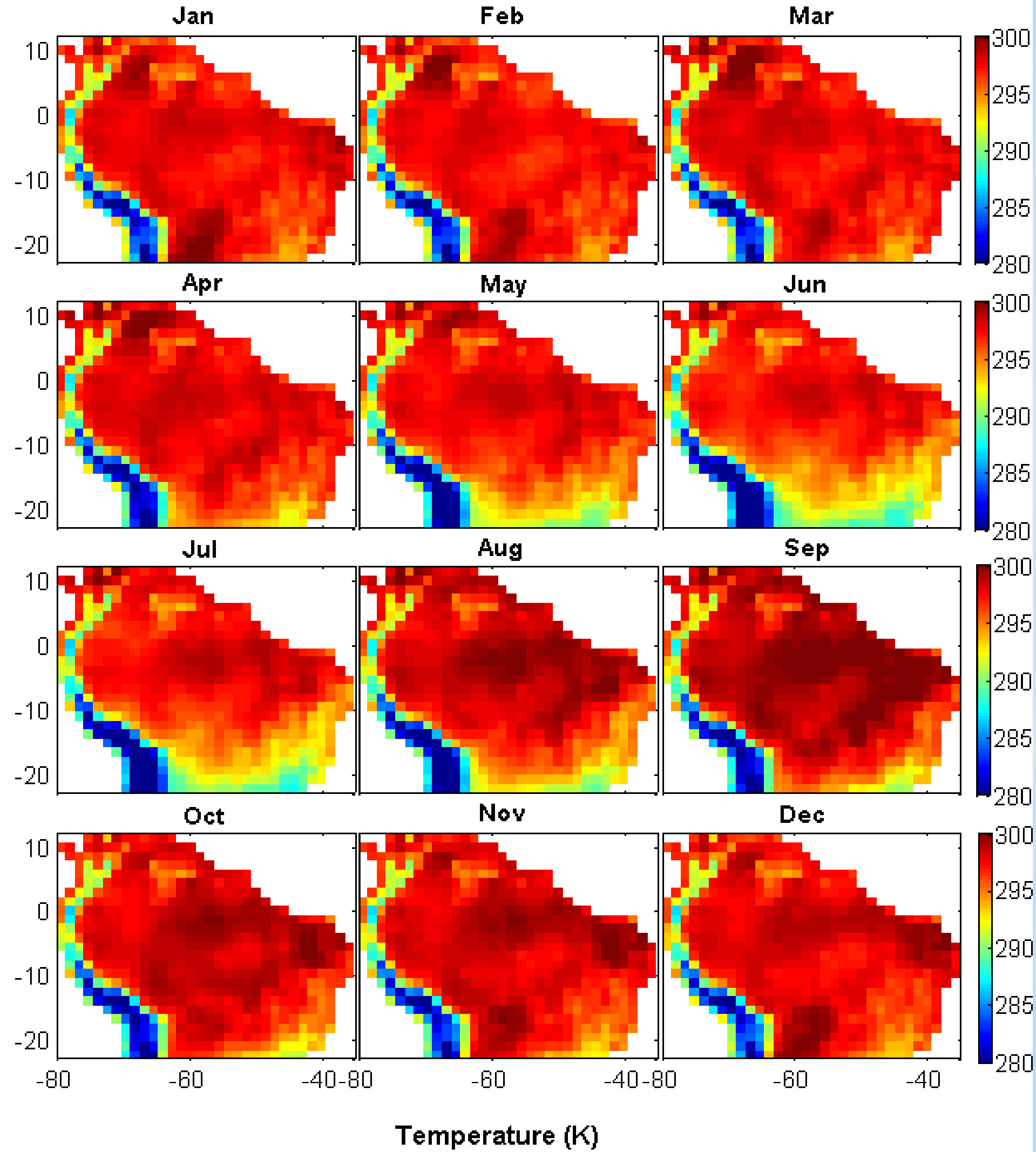
Data: 30 arc-second DEM of South America by USGS

• Climatology

- Precipitation



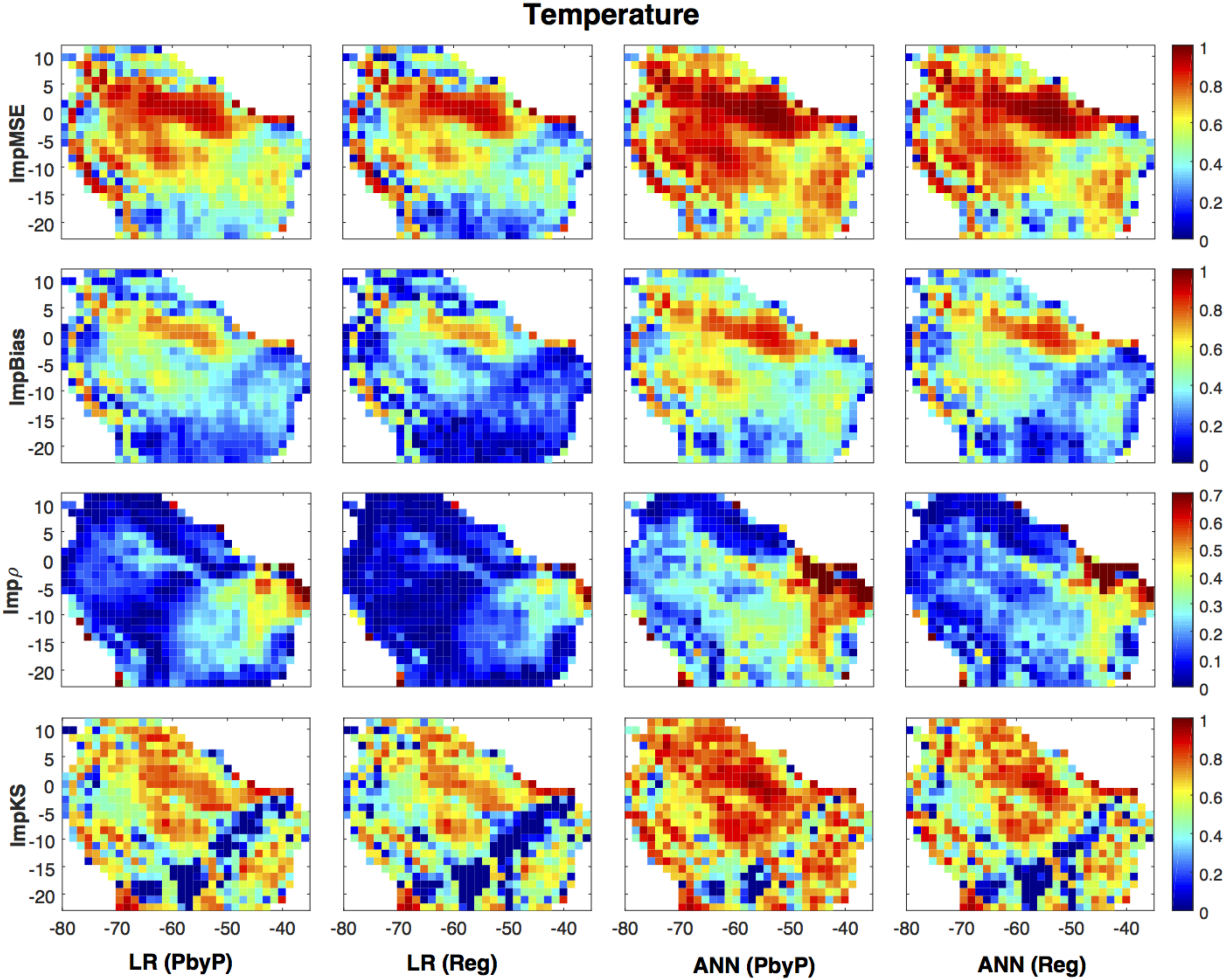
- Temperature



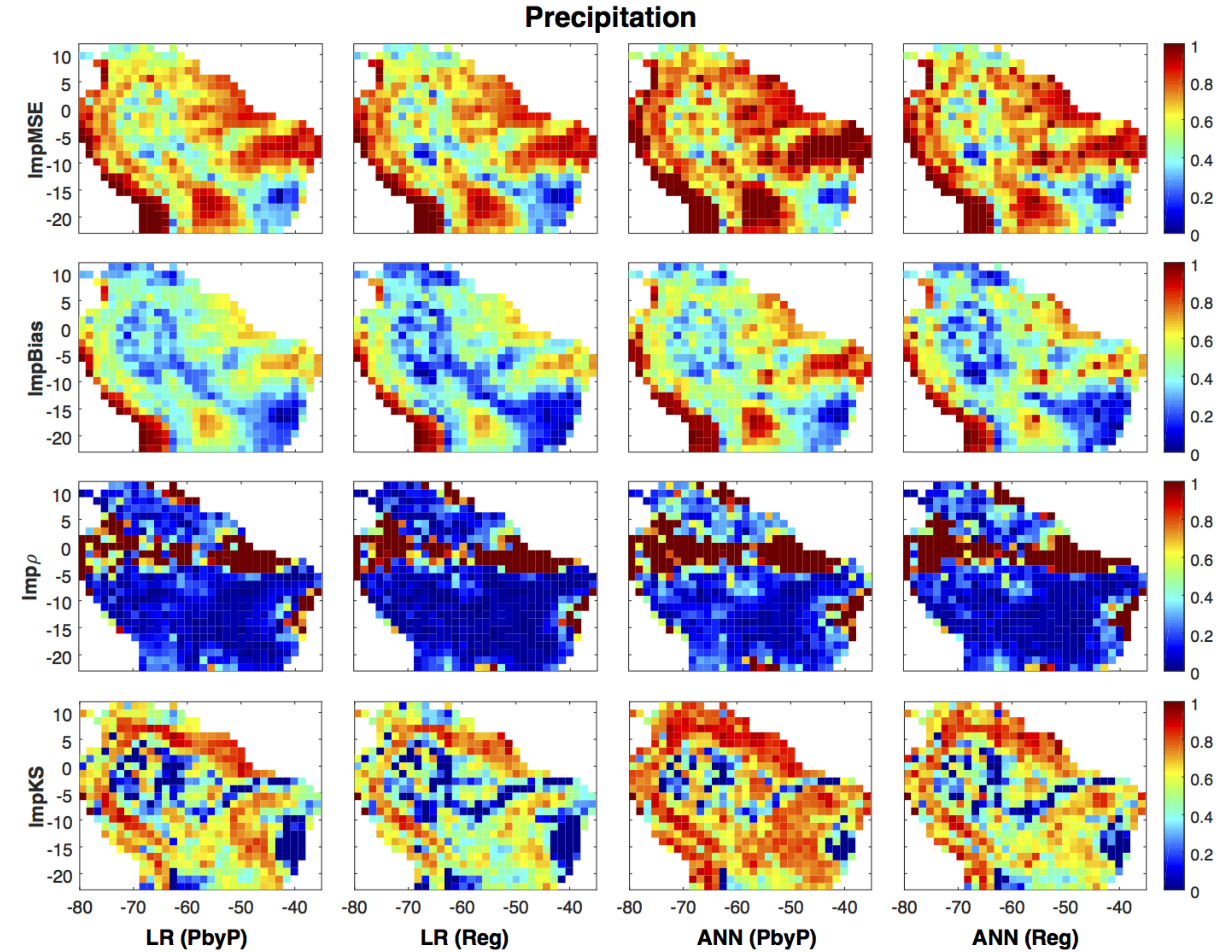
- **Train ANN Model @ defined Training Pixels**

- **Use the Trained Model @ Pixels Within Delineated Domains**

*Temperature


Temperature
ImpMSE
ImpBias
Impρ
ImpKS
10
5
0
-5
-10
-15
-20
-80
-70
-60
-50
-40
LR (PbyP)
LR (Reg)
ANN (PbyP)
ANN (Reg)
1
0.8
0.6
0.4
0.2
0
0.7
0.6
0.5
0.4
0.3
0.2
0.1

*Precipitation



Precipitation
ImpMSE
ImpBias
Impρ
ImpKS
LR (PbyP)
LR (Reg)
ANN (PbyP)
ANN (Reg)
10
5
0
-5
-10
-15
-20
-80
-70
-60
-50
-40
1
0.8
0.6
0.4
0.2
0

- ✓ The trained ANN model performs well for both calibration and validation periods.

Temporal Generalization √

- ✓ Is the ANN model useful over locations where observations are not available?

Spatial Generalization √

Summary & Conclusions

- A three layer feedfoward neural network has generalization ability in time and space to reduce the biases of climate variables (temperature and precipitation).
- Inputs to the temperature network: T, TS, Q, LW_n, SW_n
- Inputs to the precipitation network: $P_t, P_{t-1}, P_{t-2}, P_{t-3}, \sigma_t^{3by3}$
- Trained model can improve all statistics (MSE, Bias, ρ, KS) in calibration and validation periods at all pixels.
- ANN outperforms LR in bias correction of temperature and precipitation.
- The larger improvements by the regression models occur over the regions with larger (smaller) original error/KS (ρ).

- Delineation of the region based on the systematic errors of CCSM3 outputs into 5 domains, which is consistent with the physical features of the regions (land cover, elevation, climatology):

1. D1 over southeast of Brazil; covered by savannas, small patches of deciduous broadleaf forest; with drier-cooler climate than D2.
2. D2 over the Amazon basin in Brazil; covered by evergreen broadleaf forest; with wetter-warmer climate than D1.
3. D3 over Amazon River Delta region; covered by evergreen broadleaf forest and large wetlands.
4. D4 over north; covered by evergreen forest, grass, cropland; has contrast in rainfall regime with D2 (D3).
5. D5 over west; contains the Andes Mt; covered by evergreen and deciduous forest, grass, shrub, cropland, barren; most heterogeneous region with the driest-coolest climate and highest elevation.

- Precipitation is the best indicator.
- The highest variability of temperature and elevation in D5 can clearly separate this domain from the others.

- Many observations are not required to construct a robust relationship between the input-output.
    - ✓ 10% of the domain pixels for temperature network.
    - ✓ 13% of the domain pixels for precipitation network.

This reduces the overall computational requirements, time and memory usage significantly.

Future Work

- Apply the proposed method for bias correction of other climate model outputs.

- Test the regionalization ability of the method over other domains.

- Evaluate the performance of ANN model using other training approaches.

Thank you

BPGDR

$$net_k = \sum_i w_{ki} x_i \qquad O_k = f_k(net_k) \qquad net_j = \sum_k w_{jk} O_k \qquad O_j = f_j(net_j)$$

- For the weights that are connected to the output nodes (output weights):

$$\Delta w_{jk} = \eta \frac{\partial E}{\partial w_{jk}} = \eta \frac{\partial E}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{jk}} = \eta \frac{\partial E}{\partial O_j} \cdot \frac{\partial O_j}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{jk}}$$

$$\delta_j = \frac{\partial E}{\partial net_j}$$

$$\frac{\partial E}{\partial O_j} = \frac{\partial \left[\frac{1}{2} (Tar_j - O_j)^2\right]}{\partial O_j} = -(Tar_j - O_j)$$

$$\frac{\partial O_j}{\partial net_j} = \frac{\partial f_j(net_j)}{\partial net_j} = f_j'(net_j)$$

$$\frac{\partial net_j}{\partial w_{jk}} = \frac{\partial (\sum_k w_{jk} O_k)}{\partial w_{jk}} = O_k$$

O_k : output from unit k in the hidden layer

$$\Delta w_{jk} = \eta \left(Tar_j - O_j\right) f_j'(net_j)\, O_k$$

- For the weights that are connected to the hidden nodes (hidden weights):

$$\Delta w_{ki} = \eta \frac{\partial E}{\partial w_{ki}} = \eta \frac{\partial E}{\partial net_k} . \frac{\partial net_k}{\partial w_{ki}} = \eta \frac{\partial E}{\partial O_k} . \frac{\partial O_k}{\partial net_k} . \frac{\partial net_k}{\partial w_{ki}}$$

$$\delta_k = \frac{\partial E}{\partial net_k}$$

$$\frac{\partial E}{\partial O_k} = \frac{\partial E}{\partial net_j} . \frac{\partial net_j}{\partial O_k} = \delta_j w_{jk}$$

$$\frac{\partial O_k}{\partial net_k} = \frac{\partial f(net_k)}{\partial net_k} = f_k'(net_k)$$

$$\frac{\partial net_k}{\partial w_{ki}} = \frac{\partial (\sum_i w_{ki} x_i)}{\partial w_{ki}} = x_i$$

$$\Delta w_{ki} = \eta \, \delta_j w_{jk} \, f_k'(net_k) \, x_i$$

Experimental Setup of the ANN

UNKNOWN	METHOD
Structure	literature , T & E [+]
Number of the hidden layers	literature
Number of the hidden nodes	constructive algorithm [++]
Transfer functions	literature , T & E
Learning rate	literature , T & E
Performance function	literature
Best set of inputs	physics , T & E
Training function (Weights)	literature , T & E

[+] T & E : trial and error

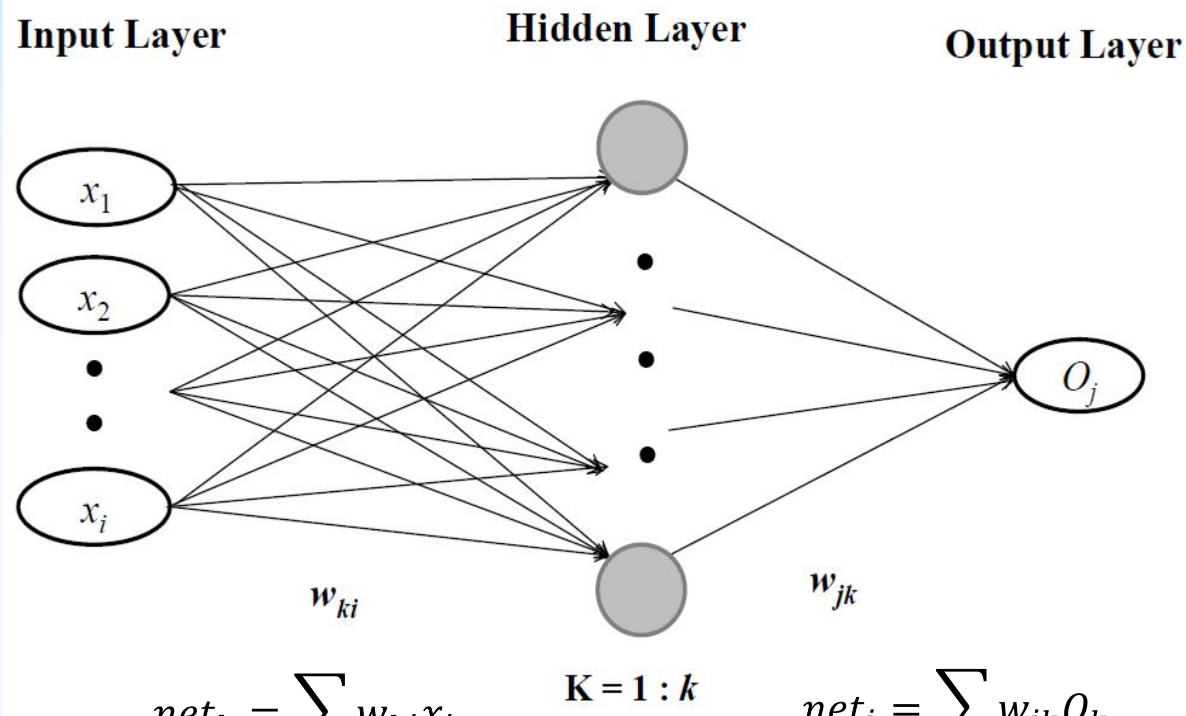
[++] constructive algorithm (Kwok and Yeung 1997)

Proposed ANN

Structure	FNN
Number of the hidden layers	1
Number of the hidden nodes	5-35
Transfer functions	Hyperbolic tangent - Linear
Learning rate	0.05-0.001
Performance function	mse +
Best set of inputs	T, TS, Q, LWn, SWn ++
Training function (Weights)	BPGDR algorithm (incremental)

+ mse : mean square error

++ T : air temperature, TS: skin temperature, Q : specific humidity
LWn : net longwave radiation, SWn : net shortwave radiation



$$net_k = \sum_i w_{ki} x_i$$
$$\downarrow$$
$$O_k = f_k(net_k)$$

$$net_j = \sum_k w_{jk} O_k$$
$$\downarrow$$
$$O_j = f_j(net_j)$$

- For the output weights:

$$\Delta w_{jk} = \eta \frac{\partial E}{\partial w_{jk}} = \eta \frac{\partial E}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{jk}} = \eta \frac{\partial E}{\partial O_j} \cdot \frac{\partial O_j}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{jk}}$$

η : learning rate (0-1)

$$\Delta w_{jk} = \eta \left(Tar_j - O_j\right) f_j'(net_j)\, O_k$$

- For the hidden weights:

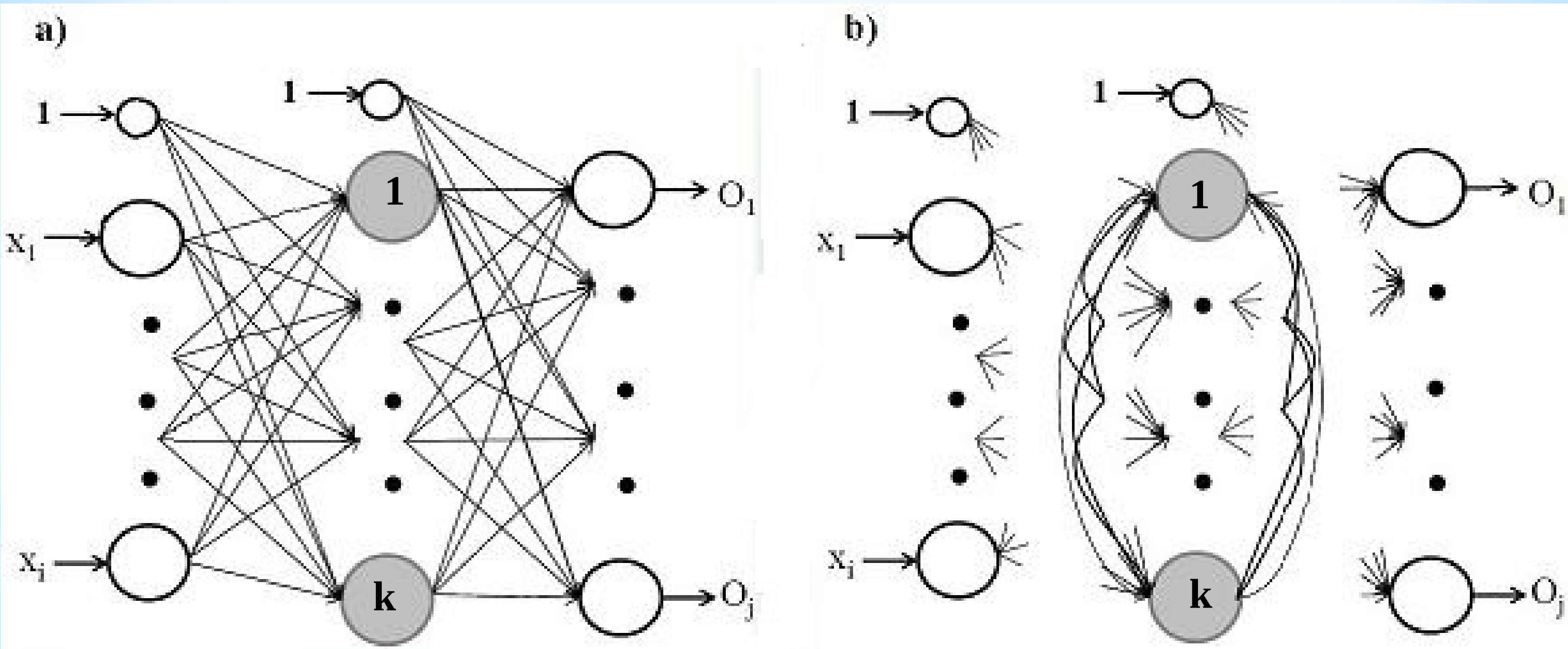
$$\Delta w_{ki} = \eta \frac{\partial E}{\partial w_{ki}} = \eta \frac{\partial E}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{ki}} = \eta \frac{\partial E}{\partial O_k} \cdot \frac{\partial O_k}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{ki}}$$

$$\Delta w_{ki} = \eta \frac{\partial E}{\partial net_j} w_{jk}\, f_k'(net_k)\, x_i$$

Determination of the ANN's Architecture

Feedforward Neural Network
FNN

Recurrent Neural Network
RNN



Input layer
1,…., i

Hidden layer

Output layer
1,…., j

Input layer
1,…., i

Hidden layer

Output layer
1,…., j