Multi-model Ensemble Analysis with Neural Network Gaussian Processes

Trevor Harris, Bo Li, Ryan Sriver

November 9, 2023

Texas A&M University
University of Illinois at Urbana-Champaign

Climate models

- Climate models are physical models of the earth system. Include atmosphere, ocean, land, sea ice, and (lately) biogeochemical processes.

- "Primary tool for investigating the response of the climate system to changes in forcings (increases in CO2)

  ... and for making *projections of the future climate*"

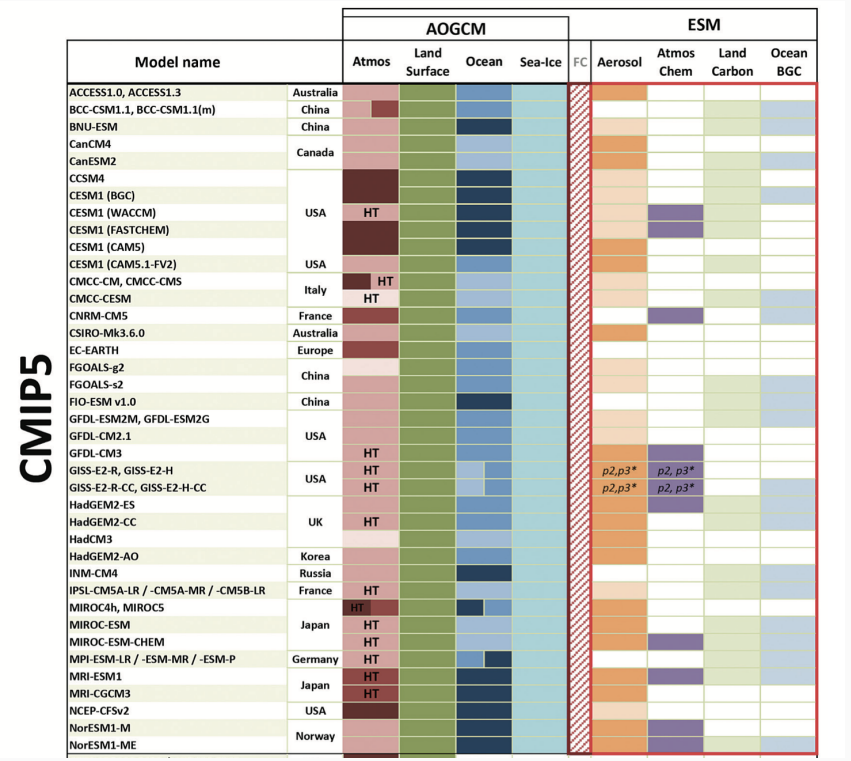


Figure 1: List of AOGCMs and ESMs participating in CMIP5 and their complexity (IPCC 5th assessment report)

Multi-model Ensembles

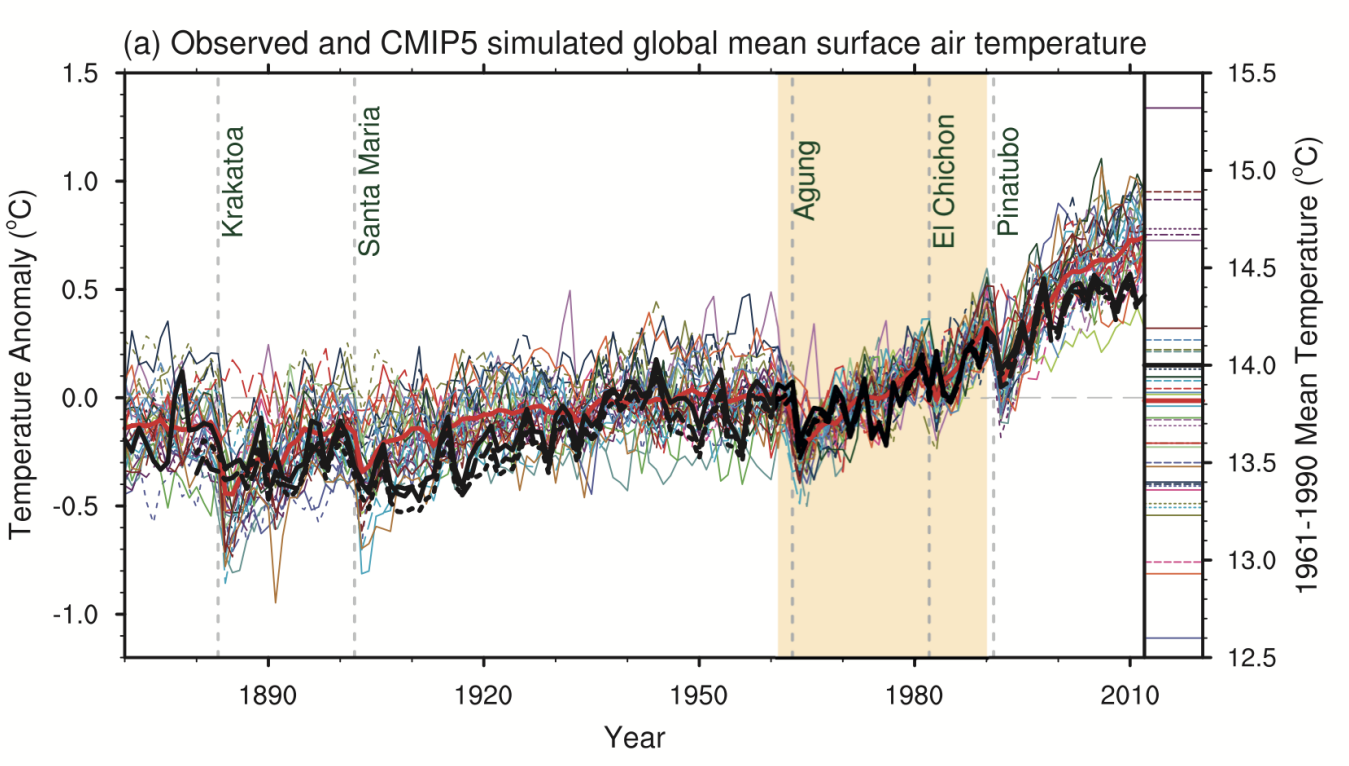


Figure 2: Gobal mean predictions for each CMIP5 model (colored lines), the model mean (red) and observations (black). Different models yield different predictions.

- Multi-model ensemble analysis – how to combine models to best resemble the actual climate?

Multi-model Ensemble Analysis

- Climate models produce spatio-temporal output (discretized to a grid). Combine directly?

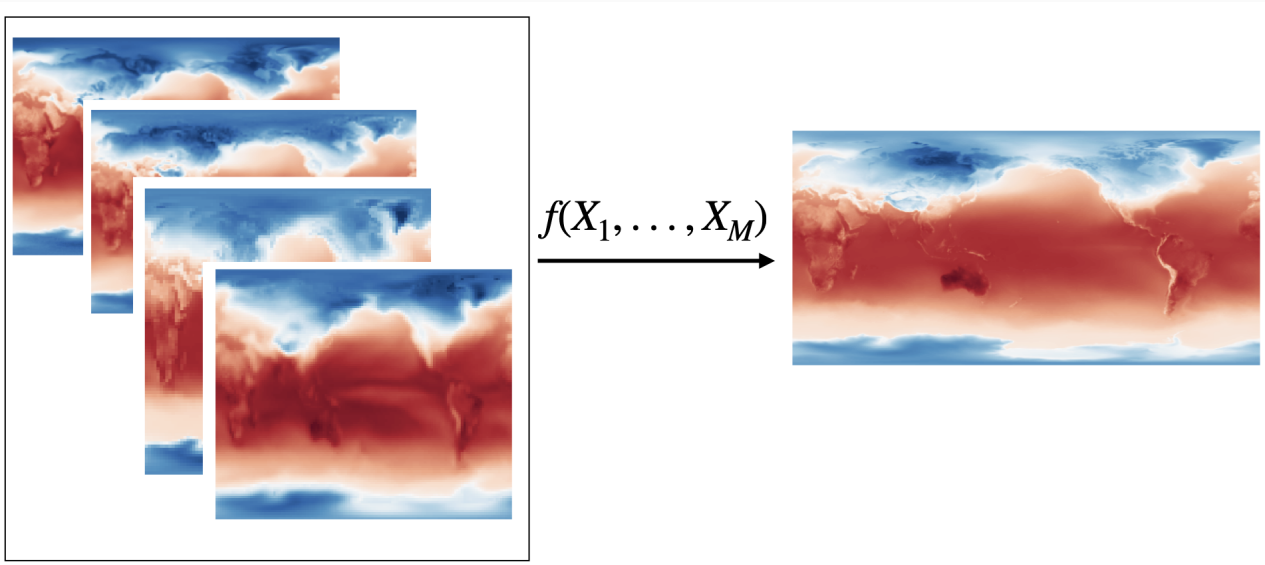


Figure 3: Goal: combine multiple climate fields into a single estimate

- More informative but much more difficult than averaging global means
  - Resize to common grid - introduces bias and lose information
  - Consider correlations between models and observations?
  - Spatially varying weights? Tons of parameters?

Problems with spatial analysis

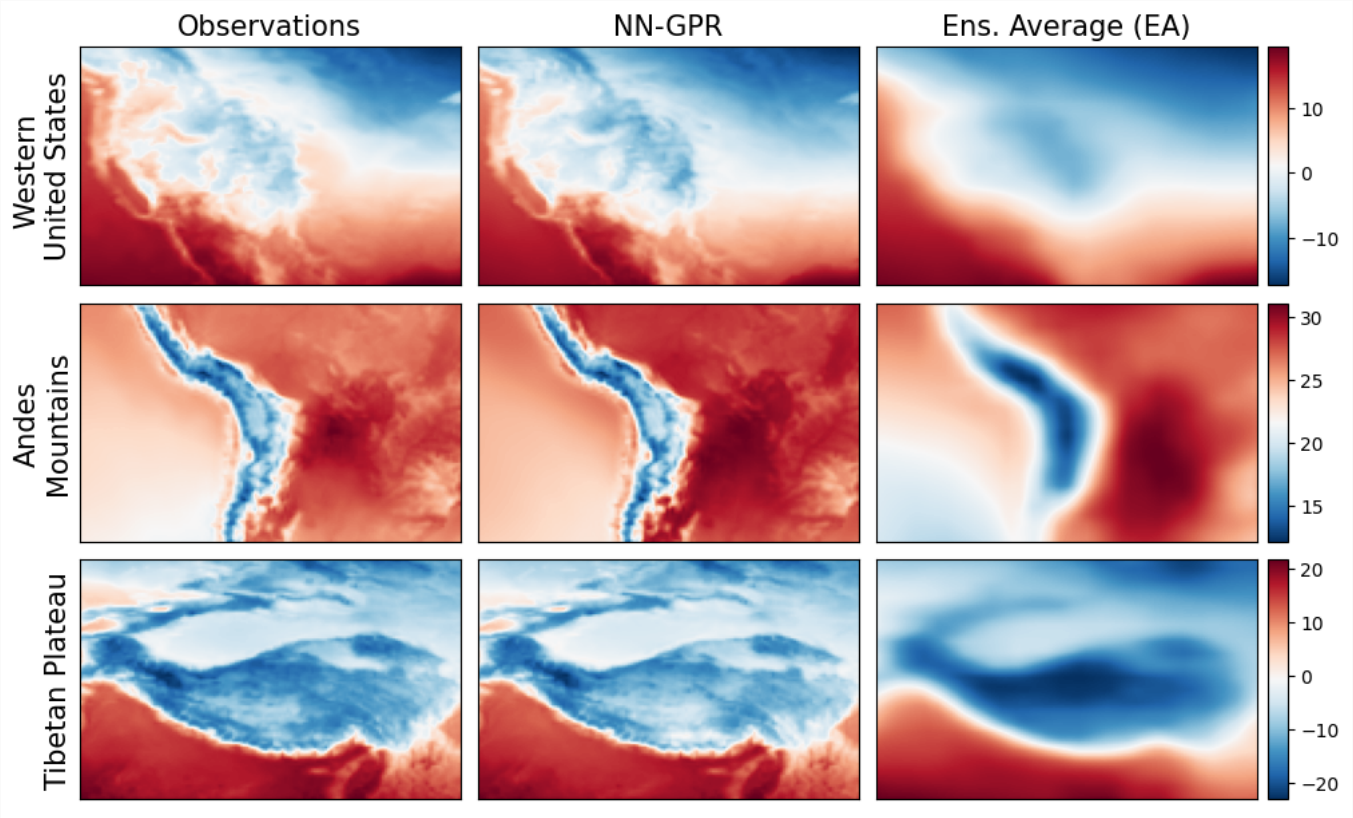


Figure 4: ERA5 reanalysis field for January 2015 (left), our proposed method NN-GPR's prediction (center) and an ensemble average of the 16 climate models (right)

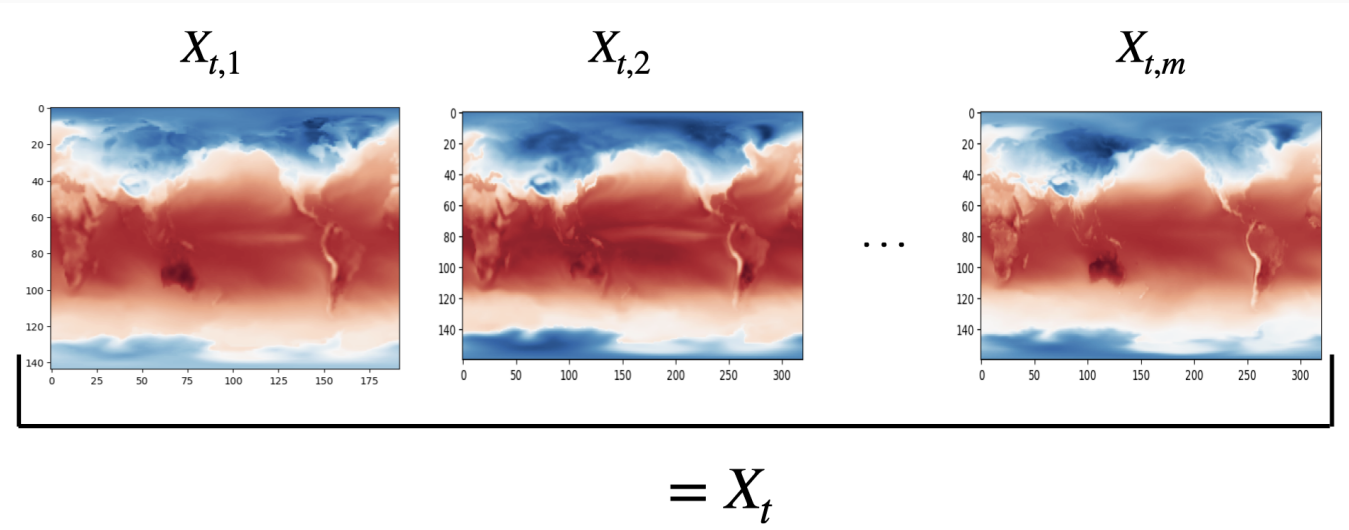
- Averaging in space destroys spatial detail. No longer represents the climate process well.

Previous Work

- How can we do this without an explosion of complexity?
- Model integration – combining multiple climate projections into a unified projection
  - **Ensemble averaging** – democratic and weighted (Giorgi and Mearns, 2002, 2003; Flato et al., 2014; Abramowitz et al., 2019)
  - **Bayesian methods** (Rougier et al., 2013; Sansom et al., 2017; Bowman et al., 2018)
  - **Regression** (Räisänen et al., 2010; Bracegirdle and Stephenson, 2012) and **Machine Learning** methods (Ghafarianzadeh and Monteleoni, 2013)
- We take a nonparametric regression approach (Gaussian process regression).
  - Climate models are used to predict observational data
  - The predictions constitute an "integration" or "analysis" of the climate models

Formulating the problem

- Reformulate multi-model ensemble analysis as a **prediction** problem
- $x_{t,i}$ – output of climate model i at time t for $i \in 1, .., m$ and $t \in 1, .., T$. Each $x_{t,i}$ is an entire gridded field of q_i grid points.
- x_t – an ensemble of m gridded climate model outputs observed at time t. $x_t \leftarrow \text{concat}(\text{vec}(x_{t,1}), ..., \text{vec}(x_{t,m}))$



Specifying the problem

- y_t reanalysis field at time t. Each y_t is a gridded field of d grid points. $y_t \leftarrow \text{vec}(y_t)$.

- Define our training dataset as the sequence $D = \{(x_t, y_t)\}_{t=1}^{T}$
  - x_t ensemble of m gridded climate model outputs observed at time t. $x_t \leftarrow \text{concat}(\text{vec}(x_{t,1}), ..., \text{vec}(x_{t,m}))$
  - y_t reanalysis field at time t on $d = \#lat \times \#lon$ grid points. $y_t \leftarrow \text{vec}(Y_t)$.

- Goal: Learn a function f s.t. $f(x_t) \approx y_t$.

- For a future model ensemble x_F we define the integration of the ensemble members as $f(x_F)$.

Specifying the problem

Can we consider deep learning to solve this problem?

Can we parameterize f as a deep neural network f_θ and simply learn θ by minimizing some loss, say $||y_t - f_\theta(x_t)||_2$?

- Good for fitting high dimensional, non-linear regression functions.
- Relatively "easy" to train with modern software (Pytorch, Tensorflow, etc.)
- ...but also data hungry, poor uncertainty quantification, poor robustness to covariate shift (climate change)

Compromise with an asymptotic approximation!

Infinite Width Neural Networks

Infinite width neural networks allow the number of hidden units (layer widths) to go to infinity

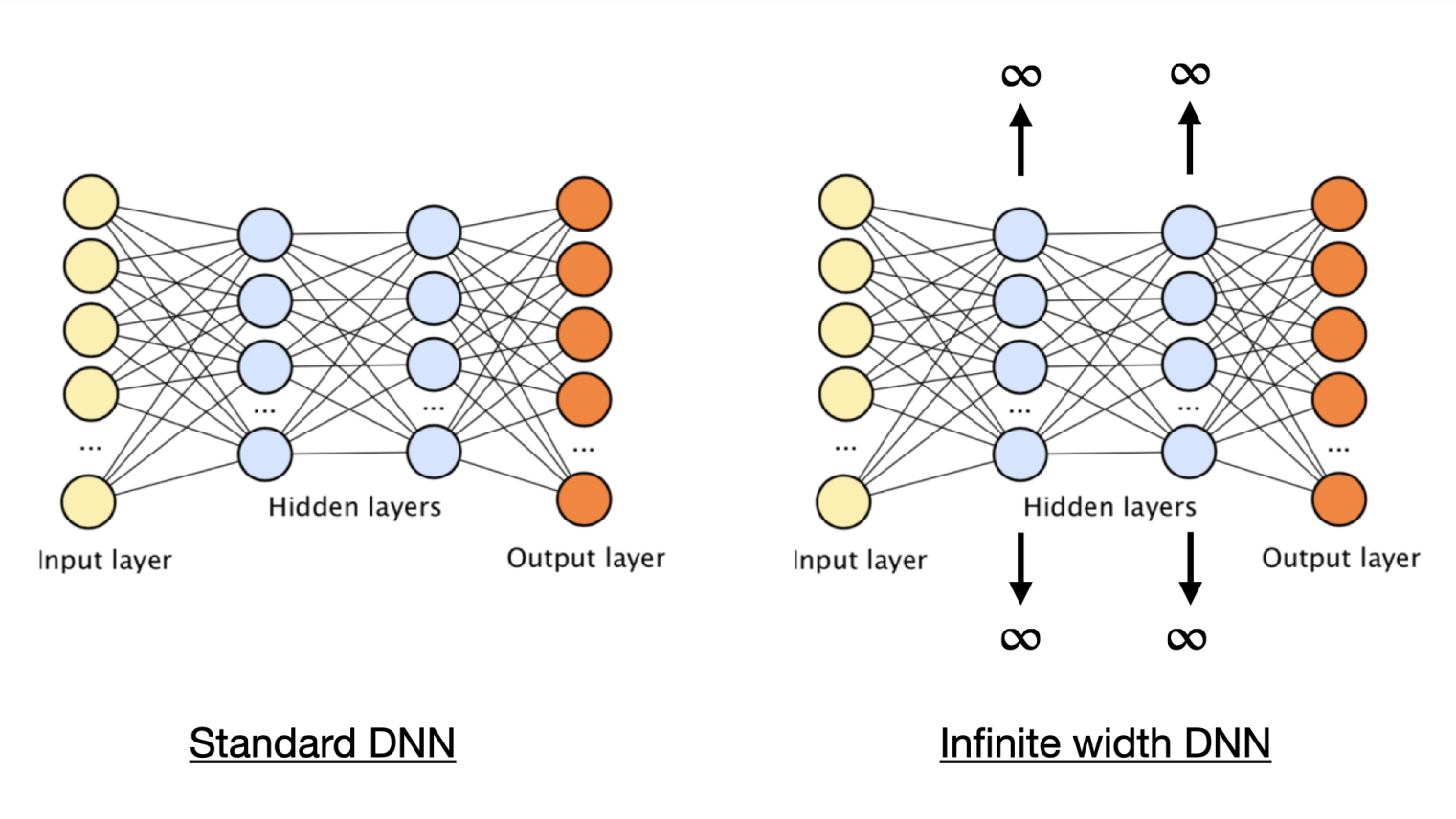


Figure 5: Left: Standard deep neural network (DNN). Right: The same DNN after letting the hidden layers become infinitely wide (infinite nodes).

Neural Network Gaussian Processes (NNGP)

- Let $\Phi(\cdot)$ denote an untrained, fully connected, deep neural network with L layers.
- Lee et al. (2017) showed that if we place priors p (for all layers) on the weights and biases

$$W_{ij}^{l} \sim p(0, \sigma_w^2 / N_l)$$
$$b_i^{l} \sim p(0, \sigma_b^2)$$

then as layer width $N_l \to \infty$ for all layers

$$\Phi(x) \to \mathcal{GP}(0, K_\phi(x_t, x_s))$$

1. The output of an untrained $\Phi(x)$ converges to a zero mean Gaussian Process covariance function K_ϕ, $\phi = \{\sigma_b^2, \sigma_w^2\}$
2. Gaussian Process Regression (GPR) with K_ϕ exactly reproduces the predictions of a fully trained infinite width $\Phi(x)$

Neural Network Gaussian Processes (NNGP)

- K_ϕ – defined by layer type, number of layers, activations, and prior variances of $\Phi(\cdot)$.
- Given two climate model ensembles x_t and x_s, then for L layers...

$$
\begin{aligned}
K_\phi(x_t, x_s) &= K^L(x_t, x_s) \\
K^L(x_t, x_s) &= \sigma_b^2 + \sigma_w^2 F(K^{L-1}(x_t, x_t), K^{L-1}(x_t, x_s), K^{L-1}(x_s, x_s)) \\
K^{L-1}(x_t, x_s) &= \sigma_b^2 + \sigma_w^2 F(K^{L-2}(x_t, x_t), K^{L-2}(x_t, x_s), K^{L-2}(x_s, x_s)) \\
&\vdots \\
K^1(x_t, x_s) &= \sigma_b^2 + \sigma_w^2 F(K^0(x_t, x_t), K^0(x_t, x_s), K^0(x_s, x_s)) \\
K^0(x_t, x_s) &= \sigma_b^2 + \sigma_w^2 (x_t x_s^T / q)
\end{aligned}
$$

  where q is the dimension of x_t (and x_s), i..e $q = \sum_{i=1}^m q_i$.

- Recursively defined covariance function. Highly non-stationary.

Neural Network Gaussian Processes (NNGP)

- Importantly depends on the function F, which is uniquely specified by the activation function (non-linearity)

- In some cases F is computable and differentiable. Ex. ReLU

$$F(\cdot) = \frac{1}{2\pi}\sqrt{K^l(x_t, x_t)K^l(x_s, x_s)}\left(\sin(\theta^l_{x_t,x_s}) + (\theta^l_{x_t,x_s} - \pi)\cos(\theta^l_{x_t,x_s})\right)$$

where

$$\theta^l_{x_t,x_s} = \arccos\left(\frac{K^l(x_t, x_s)}{\sqrt{K^l(x_t, x_t)K^l(x_s, x_s)}}\right)$$

- Differentiability is essential for fast estimation of the hyperparameters σ_w^2 , σ_b^2, and σ^2.

Neural Network Gaussian Processes (NNGP)

- Posterior predictive distribution of GPR with K_ϕ exactly reproduces the predictions of a fully trained infinite width $\Phi(\cdot)$

- Asymptotic approximation to a fully trained, finite width $\Phi(\cdot)$.
  - Can be "trained" to learn $f(x_t) \approx y_t$ with very little data
  - Prediction uncertainty by default
  - Embed into a larger statistical model

- Robust to covariate shift?
  - Not really, but we know what to expect far away from training data (predictions $\mapsto$ prior)
  - NNGP kernel less affected than exponential and squared exponential kernels

Statistical Model - NN-GPR

- Recall the data $D = \{(x_t, y_t)\}_{t=1}^{T}$
  - x_t represents an *ensemble* of m gridded climate model fields all observed at time t. Vectorized into a $q = \sum_{i=1}^{m} q_i$ dimensional vector.
  - y_t represents the gridded reanalysis field at time t. Vectorized into a d dimensional vector.

- Since $d > 1$, we will model each dimension (spatial location) of y_t independently with its own GPR model.
  - Jointly modeling all d locations computationally infeasible since $d = 1,065,600$ in the application

- We use $y_t(s)$ to denote location $s \in s_1, ..., s_d$ in the reanalysis field y_t.

Statistical Model - NN-GPR

We model each $y_t(s)$ as

$$\begin{aligned} y_t(s) &= \bar{\mathbf{x}}_t \boldsymbol{\beta} + f_s(x_t) + \epsilon_{s,t}, \\ f_s &\sim \mathcal{GP}\left(0, K_\phi\right), \\ \epsilon_{s,t} &\sim \text{ iid } N(0, \sigma^2) \end{aligned} \tag{1}$$

- Where $\bar{\mathbf{x}}_t$ is the length-m vector of the climate model means at time t, i.e., $\bar{\mathbf{x}}_t = (\bar{x}_{t,1}, ..., \bar{x}_{t,m})^T$ for which $\bar{x}_{t,i}, 1 \leq i \leq m$ represents the spatial mean of the ith climate model output at time t
- $\mathbf{K}_\phi$ is a $T \times T$ matrix whos entries are determined by the NNGP covariance function
- We assume $\mathbf{K}_\phi$ is shared across all locations $s \in 1, ..., d$ (same ϕ at all locations).

Statistical Model - NN-GPR

Data level — $y_t(s) = \bar{\mathbf{x}}_t \boldsymbol{\beta} + f_s(x_t) + \epsilon_{s,t}$

1. $\bar{\mathbf{x}}_t \boldsymbol{\beta}$ is a trend term that helps keep the prediction of $y_t(s)$ biased towards the weighted ensemble mean.

2. $f_s(x_t)$ maps the entire climate ensemble x_t to a single point $y_t(s)$ after accounting for the trend.

3. $\epsilon_{s,t}$ represents the white noise residual after accounting for the trend and the Gaussian process defined on $x_t(s)$.

Prior level —

1. $f_s \sim \mathcal{GP}(\mathbf{0}, \mathbf{K}_\phi)$ is the Gaussian process prior corresponding to an infinitely wide Bayesian neural network with a pre-specified architecture.

Parameter Estimation

- Only $m + 3$ parameters, where $m =$ number of climate models.
- Additional hyperparameters such as activation function, layer type, and network depth (number of layers).
  - Activation function – ReLU. Common and K_ϕ is differentiable with respect to σ_b^2 and σ_w^2.
  - Layer type – Fully connected layers. Works fine and has simple covariance expression. Convolutional requires model resizing / regridding.
  - Network depth – 10 layers. Predictions relatively insensitive for 3-30 layers.
- Maximum marginal likelihood estimation of β, σ_w^2, σ_b^2, and σ^2. We optimize these parameters rather than place additional priors on them.

Integrating ensembles

- Given a new multi-model ensemble, say x_F for a future time $F > T$, we integrate the ensemble by computing the posterior predictive distribution for $y_F(s)$ for all $s \in 1, ..., d$.

- $P(y_F(s) \mid x_F, D) = N(\mu_F(s), K_F + \sigma^2)$

$$\mu_F(s) = \bar{\mathbf{x}}_F \hat{\boldsymbol{\beta}} + \mathbf{k}_\phi(x_F) \left(\mathbf{K}_\phi + \sigma^2 \mathbf{I}_T\right)^{-1} \mathbf{y}(s),$$
$$K_F = K_\phi(x_F, x_F) - \mathbf{k}_\phi(x_F) \left(\mathbf{K}_\phi + \sigma^2 \mathbf{I}_T\right)^{-1} \mathbf{k}_\phi(x_F)^T.$$

  where

$$\mathbf{y}(s) = (y_1(s), ..., y_n(s))$$
$$\mathbf{k}_\phi(x_F) = (K_\phi(x_F, x_1), ..., K_\phi(x_F, x_n))$$

- Integration of x_F defined as $\hat{y}_F = \text{concat}(\mu_F(1), ..., \mu_F(T))$

Numerical experiments

- Test methods ability to accurately predict future climate under many "perfect model" scenarios
  - Given 16 global climate models. Treat one model as the "truth". Treat other 15 as multi-model ensemble.
  - Cycle through / repeat for all models as the "truth".
- We compare our method against Ensemble Averaging, Reliability Ensemble Averaging, and Pointwise Regression, a Convolutional Neural Network (CNN), Gaussian process regression with stationary kernels, and the delta method.
  - Compare 2-meter surface temperature (T2M) and Total Precipitation (PR) estimates
  - Train on historical period (1979-2021) match reanalysis data availability
  - Test on future simulations (2021-2100) based on SSP245
  - SSP245 – Shared Socioeconomic Pathway 2 with Representative Concentration Pathway (RCP) 4.5 (medium plausible scenario)

Prediction metrics

Four metrics for measuring different qualities of our predictions.

- Point estimation
  1. **MSE** - Mean Squared Error. Standard reconstruction metric based on the $L2$ norm between prediction and observation.
  2. **SSIM** - Structural Similarity Index Measure. Image reconstruction metric that compares the "structural content" of two images.
     - High SSIM means the prediction is not blurred, noisy, skewed, or otherwise distorted, even at very fine scales.

- Uncertainty Quantification (in paper, skipped here)
  3. **CRPS** - Continuous Ranked Probability Score. Scoring rule for assessing the quality of ensemble forecasts.
  4. **Coverage** - Percent of prediction intervals that cover the target.

Point Estimation (T2M)

(↓) Mean Squared Error (MSE) - T2M

Model	2030	2040	2050	2060	2070	2080	2090	2100
NN-GPR	**1.91**	**1.97**	**2.10**	**2.27**	**2.37**	**2.53**	2.68	2.84
LM	2.29	2.28	2.38	2.51	2.54	2.57	**2.62**	**2.71**
WEA	3.29	3.27	3.40	3.54	3.54	3.60	3.62	3.67
EA	5.98	5.87	5.96	6.04	6.00	6.03	5.97	5.99
GPSE	**1.91**	**2.01**	2.26	2.57	2.85	3.23	3.60	3.96
GPEX	**1.89**	**1.97**	2.19	2.44	2.65	2.90	3.16	3.40
CNN	2.78	2.75	2.79	2.95	2.94	2.97	3.01	3.08
DELT	3.07	3.05	3.17	3.31	3.30	3.36	3.40	3.46

Table 1: Point estimation metrics, averaged over perfect model experiments, broken out by decade. NN-GPR is our method, LM is pointwise linear regression, WEA is a reliability weighted ensemble average, and EA is the naive ensemble average. Arrows indicate whether higher (↑) or lower (↓) numbers are better.

Point Estimation (T2M)

(↑) Structural Similarity Index (SSIM) - T2M

Model	2030	2040	2050	2060	2070	2080	2090	2100
NN-GPR	**0.92**	**0.92**	**0.91**	**0.91**	**0.90**	**0.89**	**0.89**	0.88
LM	0.91	0.91	0.91	0.90	0.90	**0.90**	**0.89**	**0.89**
WEA	0.89	0.88	0.88	0.88	0.87	0.87	0.87	0.87
EA	0.84	0.84	0.83	0.83	0.83	0.82	0.82	0.82
GPSE	**0.92**	**0.91**	0.90	0.89	0.88	0.87	0.86	0.86
GPEX	**0.92**	**0.92**	**0.91**	0.90	0.89	0.88	0.88	0.87
CNN	0.89	0.89	0.89	0.88	0.88	0.88	0.88	0.88
DELT	0.89	0.89	0.88	0.88	0.88	0.87	0.87	0.87

Table 2: Point estimation metrics, averaged over perfect model experiments, broken out by decade. NN-GPR is our method, LM is pointwise linear regression, WEA is a reliability weighted ensemble average, and EA is the naive ensemble average. Arrows indicate whether higher (↑) or lower (↓) numbers are better.

Point Estimation (PR)

(↓) Mean Squared Error (MSE) - PR

Model	2030	2040	2050	2060	2070	2080	2090	2100
NN-GPR	**3.84**	**3.97**	**4.05**	**4.28**	**4.47**	**4.59**	**4.68**	**4.83**
LM	4.41	4.58	4.64	4.84	4.99	5.08	5.16	5.29
WEA	4.97	5.14	5.22	5.43	5.58	5.65	5.76	5.85
EA	5.84	6.03	6.13	6.33	6.48	6.57	6.70	6.76
GPSE	**3.88**	**4.02**	**4.08**	**4.33**	**4.52**	**4.63**	**4.73**	**4.87**
GPEX	**3.86**	**3.99**	**4.06**	**4.31**	**4.49**	**4.61**	**4.71**	**4.86**
CNN	4.70	4.87	4.92	5.15	5.34	5.41	5.49	5.63
DELT	5.15	5.31	5.40	5.60	5.74	5.85	5.97	6.05

Table 3: Point estimation metrics, averaged over perfect model experiments, broken out by decade. NN-GPR is our method, LM is pointwise linear regression, WEA is a reliability weighted ensemble average, and EA is the naive ensemble average. Arrows indicate whether higher (↑) or lower (↓) numbers are better.

Point Estimation (PR)

(↑) Structural Similarity Index (SSIM) - PR

Model	2030	2040	2050	2060	2070	2080	2090	2100
NN-GPR	**0.59**	**0.58**	**0.58**	**0.57**	**0.56**	**0.56**	**0.56**	**0.55**
LM	0.55	0.55	0.54	0.54	0.54	0.54	0.53	0.53
WEA	0.50	0.49	0.49	0.49	0.49	0.49	0.48	0.48
EA	0.48	0.47	0.47	0.47	0.47	0.47	0.46	0.47
GPSE	**0.58**	**0.58**	**0.57**	**0.56**	**0.55**	**0.55**	**0.54**	**0.54**
GPEX	**0.58**	**0.58**	**0.57**	**0.56**	**0.55**	**0.55**	**0.54**	**0.54**
CNN	0.51	0.51	0.51	0.50	0.50	0.50	0.49	0.49
DELT	0.51	0.51	0.51	0.51	0.50	0.50	0.50	0.50

Table 4: Point estimation metrics, averaged over perfect model experiments, broken out by decade. NN-GPR is our method, LM is pointwise linear regression, WEA is a reliability weighted ensemble average, and EA is the naive ensemble average. Arrows indicate whether higher (↑) or lower (↓) numbers are better.

Application - Global multi-model analysis

Goal — Combine future simulations (under SSP245) from a multi-model ensemble into a single projection

- Use NN-GPR to make future projections under SSP245.
  - Project 2-meter surface temperature (T2M) and Total Precipitation (PR) monthly averages
  - Take the same 16 model ensemble from the experiments as input
  - Predict the corresponding ERA5 reanalysis fields.
  - ERA5 much higher resolution than any of the models ($d = 1440 \times 720$ grid points).

- Compare with a pointwise Linear Model (LM) and Ensemble Average (EA)
  - Train on historical period (1979-2015)
  - Test on SSP245 simulations (2015-2021)
  - Project further (2022-2100) and compare qualitatively

Global multi-model analysis

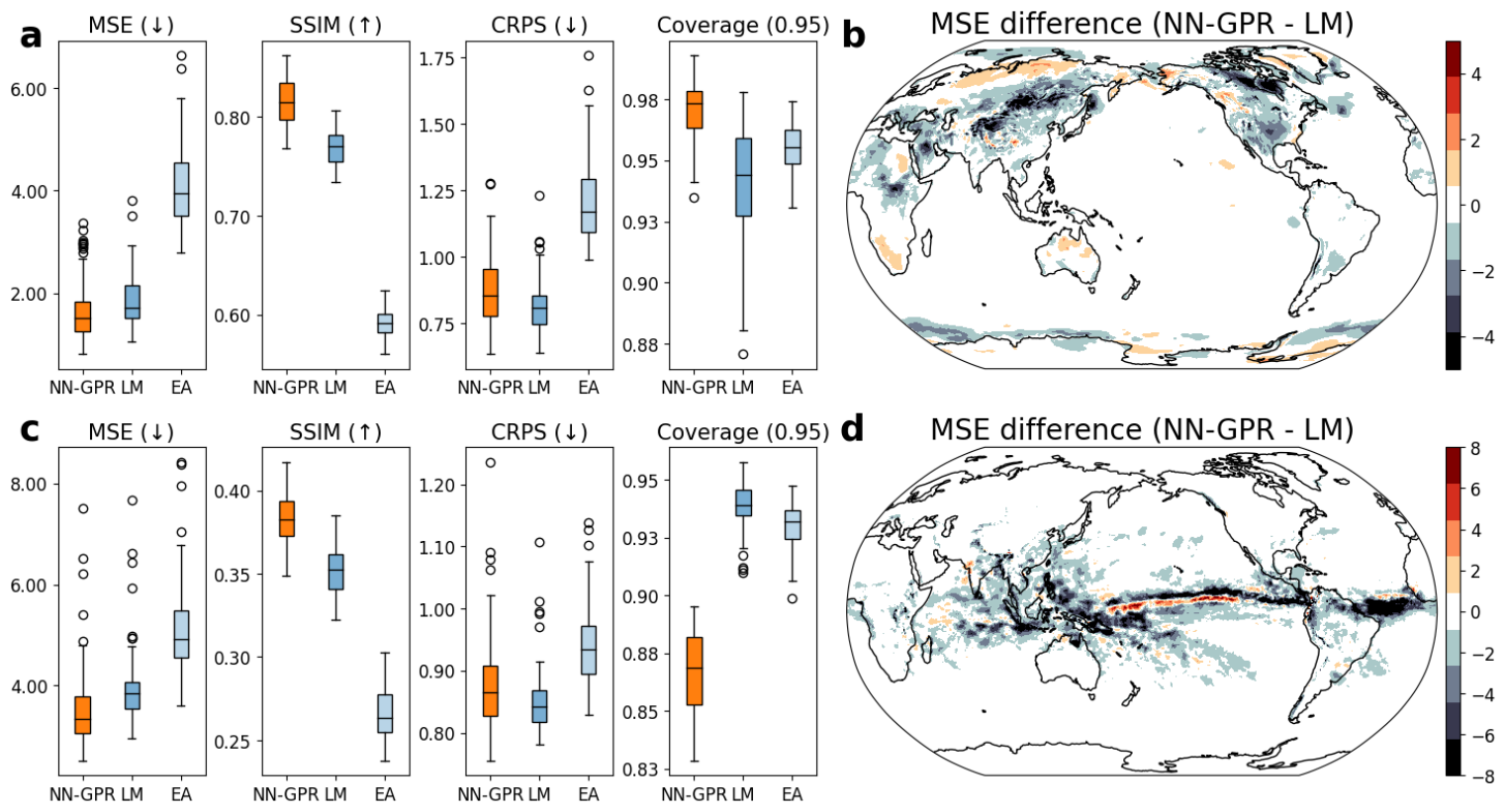


Figure 6: Boxplots for T2M (a) and PR (c) comparing the accuracy and UQ measures for each method on the reanalysis test data (2015-2021). Panels (b) and (d) show spatial differences in the MSE (averaged over time) of NN-GPR and LM for T2M(b) and PR(d).

Global multi-model analysis

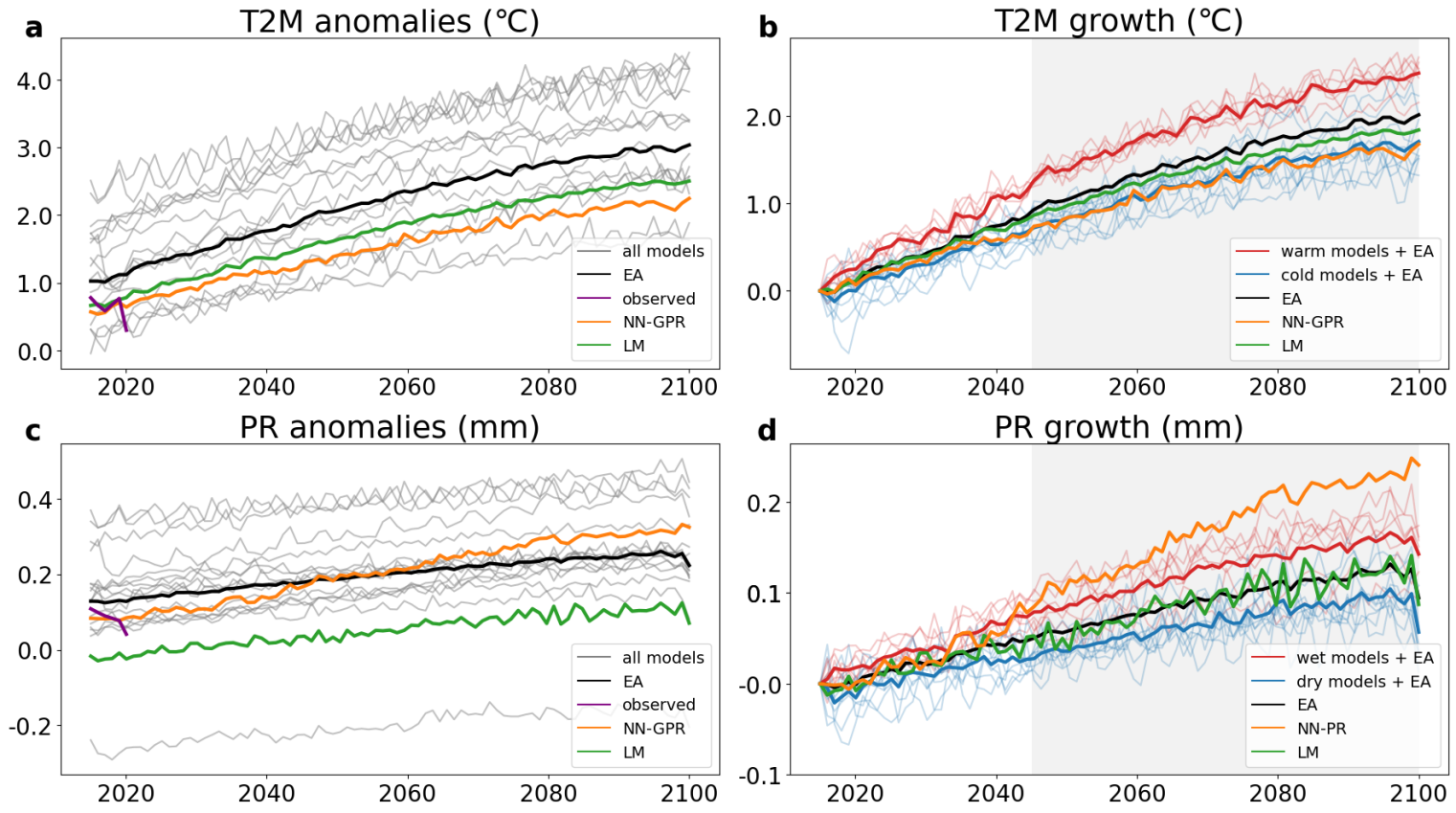


Figure 6: Panels (a) and (c) show yearly T2M and PR averages of each method, individual climate model averages (all models), and the observations. Panels (b) and (d) show projected T2M and PR growth of each method starting from 2015. In panel (b), a "warm" model has T2M growth over 1°C by 2045 and in panel (d) a "wet" model has PR growth over 0.05mm by 2045.

Application - Regional Climate Modeling

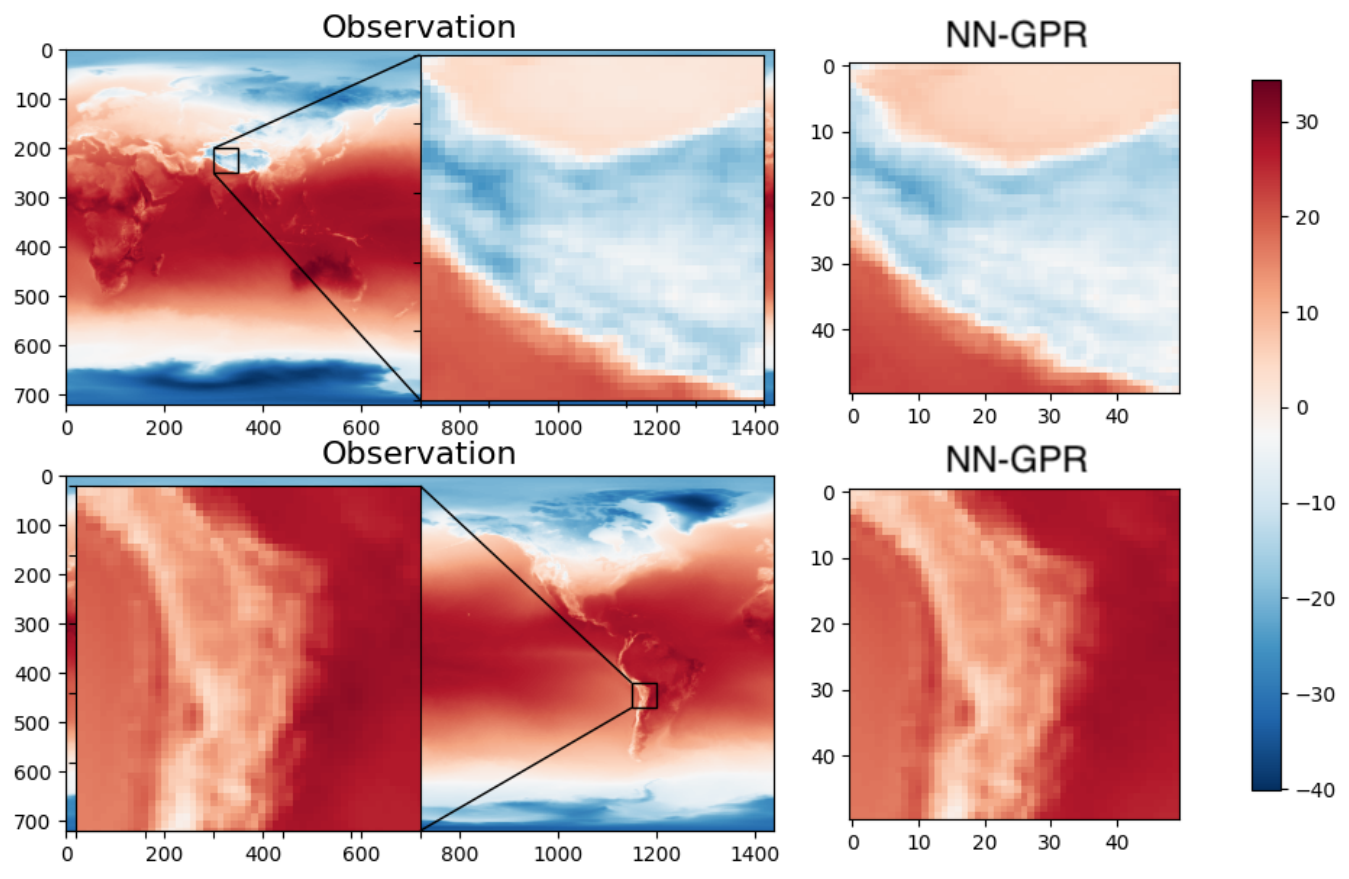


Figure 7: NN-GPR prediction vs the observed reanalysis field.

NN-GPR shows remarkable detail at the local level for a global model.

Can be used for regional climate modeling?

Regional Climate Modeling

- NN-GPR automatically converts low resolution climate models into high resolution predictions.
  - Indirect statistical downscaling of an entire ensemble

- Compare our *regional* predictions (based on global inputs) against regional climate models (RCMs) in North America.
  - Real dynamical RCMs are relatively expensive and not available everywhere

- RCM output comes from the NA-CORDEX simulations (Mearns et al., 2017)
  - CanESM2 projections downscaled via CRCM5-OUR and CanRCM4. T2M and PR.

Regional Climate Modeling

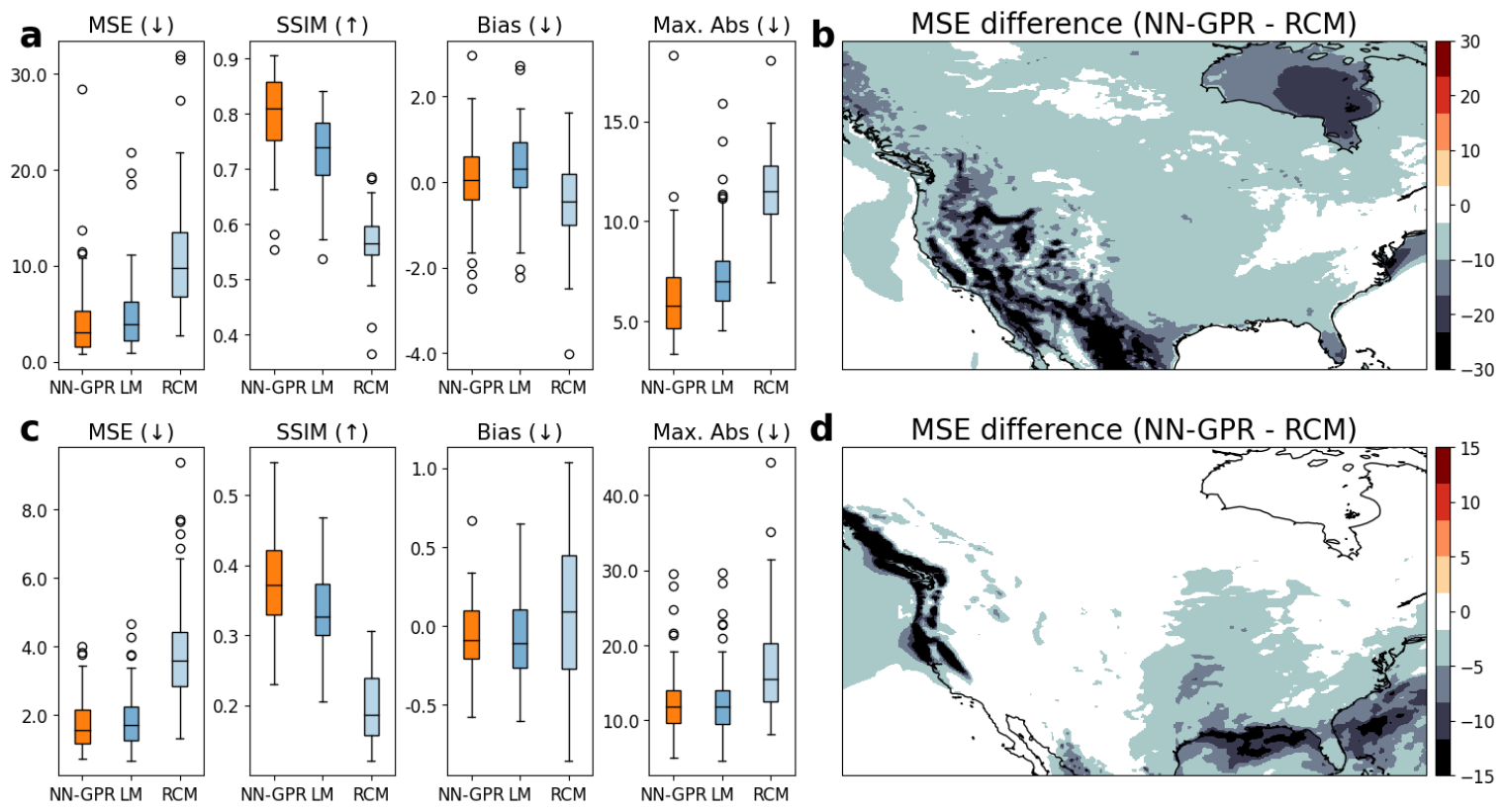


Figure 8: Skill metrics for NN-GPR, LM and the regional climate model average (RCM) for T2M (a) and PR (c) forecasting over 2015-2021. Panels (b) and (d) – spatial MSE differences (NN-GPR - RCM) of T2M (a) and PR (c) projection.

Regional Climate Modeling

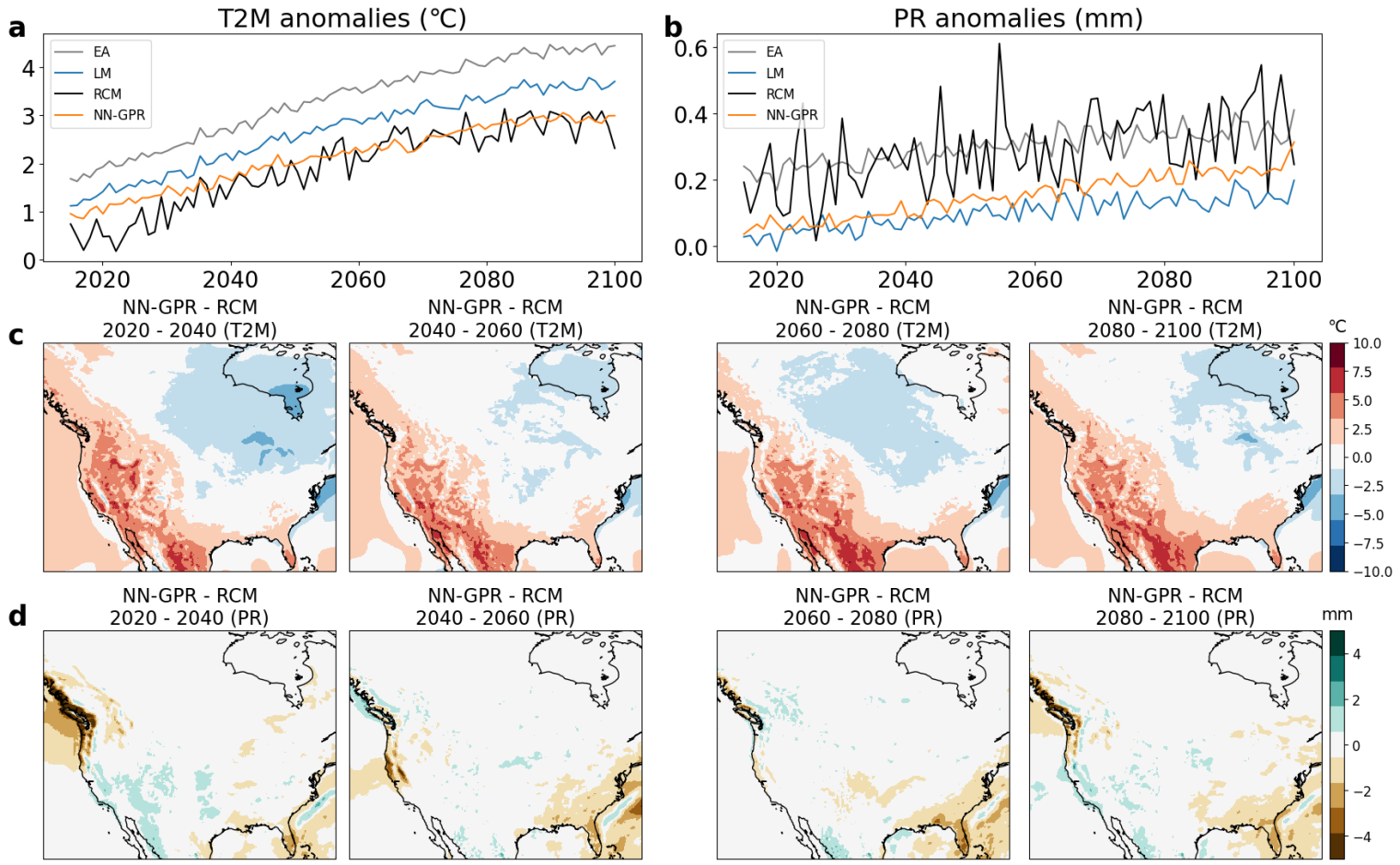


Figure 8: Panels (a) and (b) – yearly average anomaly projection (with respect to the 1950-2015 average) for each method and the RCM mean for T2M and PR. For T2M, NN-GPR closely tracks the RCM anomalies, while being consistently lower than RCM in PR.

Conclusion

- Neural Network Gaussian Process Regression (NN-GPR) approach for multi-model analysis / combining spatio-temporal fields
- NN-GPR improves over linear models and model averaging over short time horizons.
  - Weak improvement / deterioration over longer time horizons in T2M
  - Consistently more accurate than the existing approaches in PR
  - Similar overall mean predictions as ensemble mean but significantly improved the fine-scale detail.
- Downscaling like behavior
  - Comparable to RCM, could be a cheap surrogate in regions with no RCMs.
- Paper - https://arxiv.org/submit/4836661/view
  - Now accepted to AOAS!