Neural Likelihood for Spatial Transport Error Models with Applications to Aerosol Parameter Constraints

Erik A. Bensen

Department of Statistics and Data Science
Statistical Methods for the Physical Sciences Research Center
Carnegie Mellon University

45th Conference on Stochastic Processes and their Applications

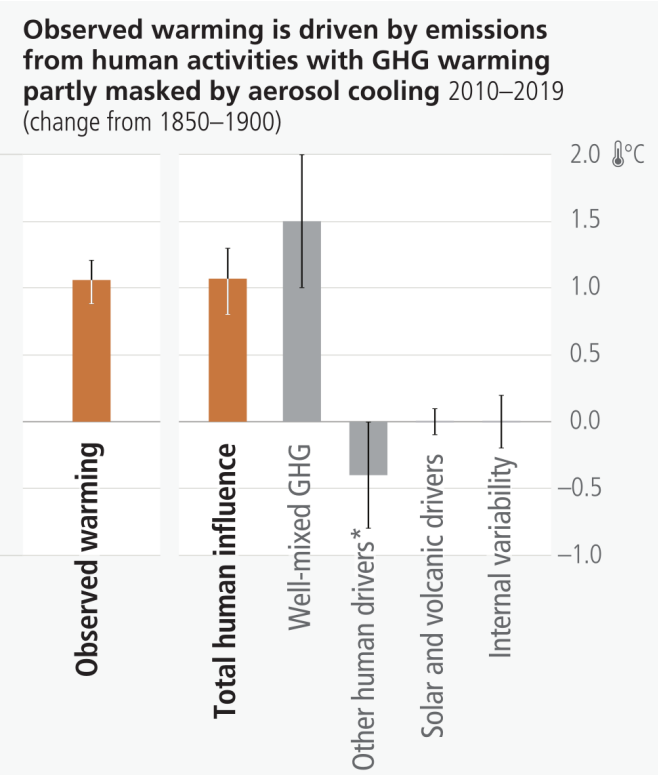
Cornell University, Ithaca, NY

June 18, 2026

Joint work with: Victor Sanchez, Hamish Gordon and Mikael Kuusela

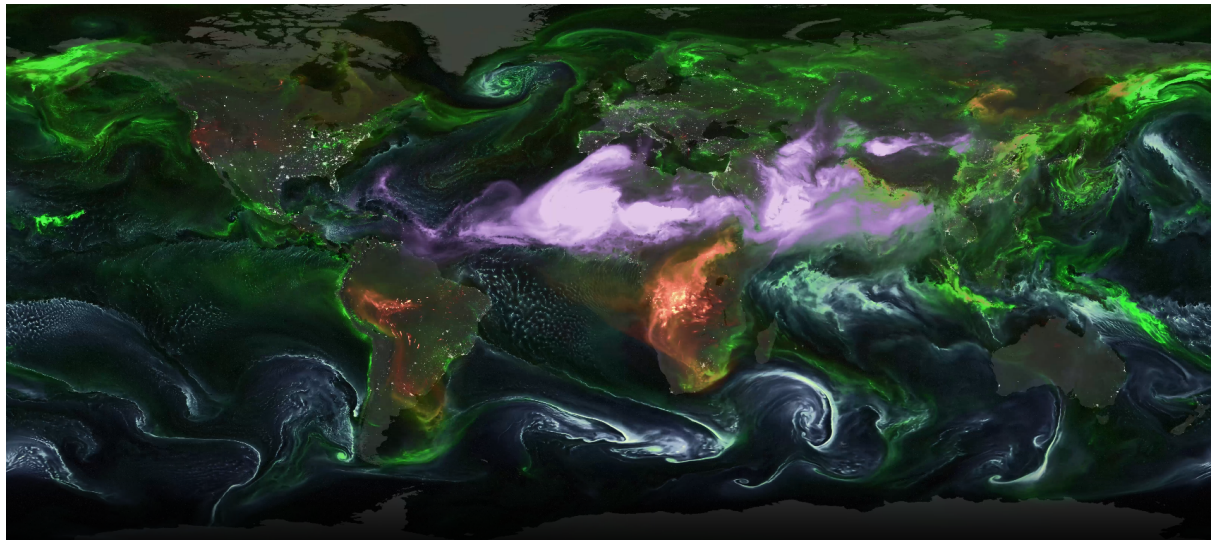
Aerosols play a key role in the climate system

From IPCC AR6:



*Other human drivers are predominantly cooling aerosols, but also warming aerosols, land-use change (land-use reflectance) and ozone.

NASA GEOS Model:



Atmospheric aerosols (smoke, sea spray, dust, sulfurs,..) are one of the largest sources of uncertainty in understanding global climate change

Inferring aerosol climate parameters

Climate models have uncertain parameters describing the emissions, properties and interactions of aerosols

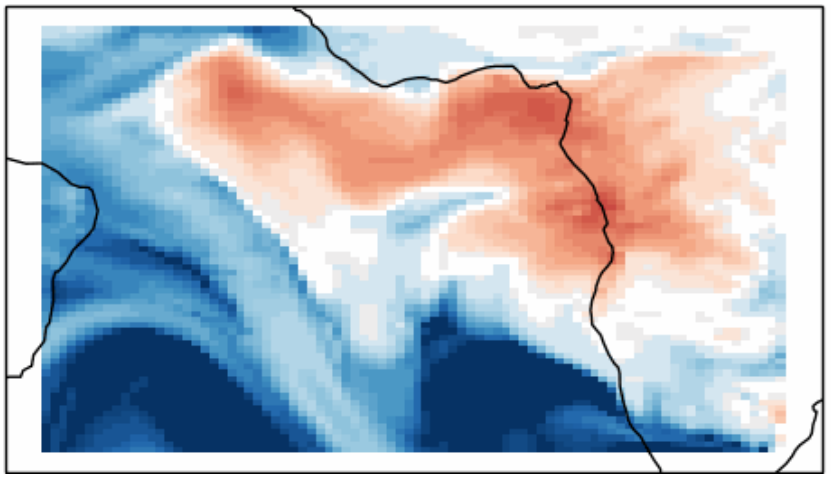
$\Rightarrow$ Aerosol uncertainties can be reduced by observationally constraining these parameters

Setting:

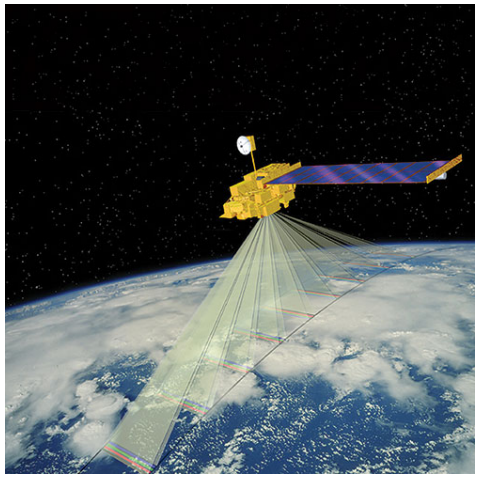
$$z(\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{u}) + \varepsilon_{\boldsymbol{x}},$$

where f is a (nudged) atmospheric model, $\boldsymbol{x}$ is a spatial location, $\boldsymbol{u}$ are parameters related to aerosols, z is a quantity observable by satellite and $\varepsilon_{\boldsymbol{x}}$ is an error term

Goal is to infer $\boldsymbol{u}$



UKESM1 model output



MODIS AOD observations

Inferring aerosol climate parameters

One can construct a *perturbed parameter ensemble* (PPE) to constrain the parameters $\boldsymbol{u}$

Idea: evaluate $f(\boldsymbol{x}; \boldsymbol{u})$ over target domain $\boldsymbol{x} \in D$ for selected $\boldsymbol{u}$'s, compare with observations and rule out implausible $\boldsymbol{u}$'s

Challenges:

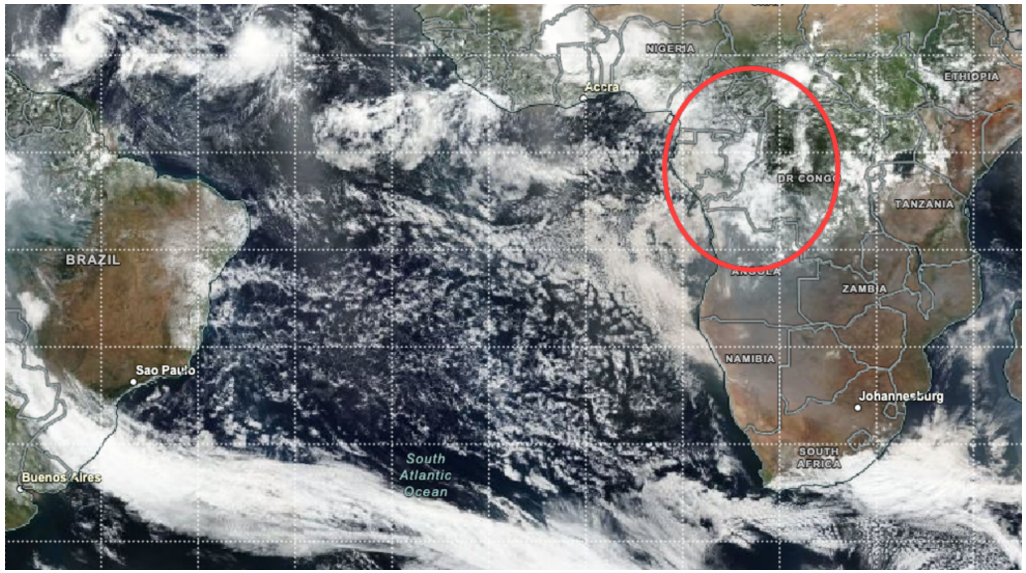
1. Can only afford to evaluate f for $n \approx 200$ parameters $\boldsymbol{u}$
2. f might be systematically misspecified

Target domain

We will focus on a region over Central Africa and Central Atlantic

Important for studying the effect of smoke on climate due to seasonal agricultural burnings

We will constrain UKESM1 model parameters based on satellite aerosol optical depth (AOD) observations



Inferring aerosol climate parameters

It is customary to fit an emulator to the PPE to handle the small number of climate model evaluations

- Here we use Gaussian process regression over $\boldsymbol{u}$ fitted separately for each location $\boldsymbol{x}$

Carzon et al. (2023, 2025) developed a test inversion framework to constrain $\boldsymbol{u}$

This approach uses the model

$$z(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}; \boldsymbol{u}) + \varepsilon_{\boldsymbol{x},\boldsymbol{u},emu} + \varepsilon_{\boldsymbol{x},meas} + \varepsilon_{\boldsymbol{x},other},$$

where $\hat{f}$ is a GP emulator prediction, $\varepsilon_{\boldsymbol{x},\boldsymbol{u},emu}$ is emulator uncertainty, $\varepsilon_{\boldsymbol{x},meas}$ is measurement error and $\varepsilon_{\boldsymbol{x},other}$ is a data-driven model discrepancy term

Modeling the model discrepancy

Due to misspecified physics, meteorology, etc., there are going to be systematic discrepancies between the atmospheric model output and the observations

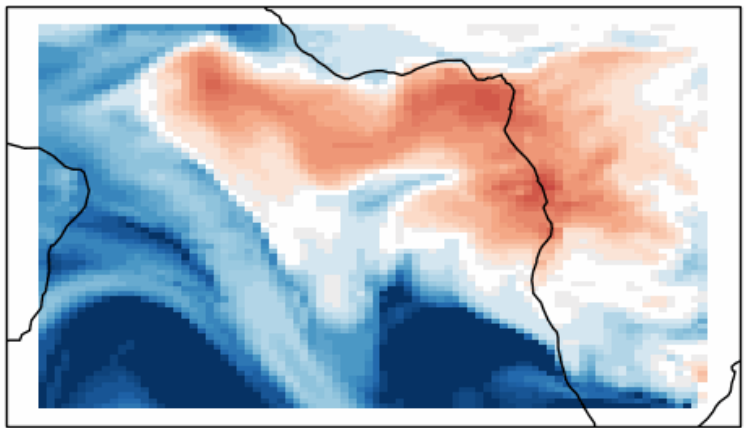
In the model

$$z(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}; \boldsymbol{u}) + \varepsilon_{\boldsymbol{x},\boldsymbol{u},emu} + \varepsilon_{\boldsymbol{x},meas} + \varepsilon_{\boldsymbol{x},other},$$

this is captured by the $\varepsilon_{\boldsymbol{x},other}$ term, which was modeled as an i.i.d. error term

This can be generalized to a spatially correlated additive error process as in Kennedy and O'Hagan (2001); Brynjarsdòttir and O'Hagan (2014) by assuming $\varepsilon_{\mathbf{x},other} \sim \mathrm{GP}(0, K)$ with standard deviation δ and length scales ℓ_x^v, ℓ_y^v

However, an additive error might not be the best way of modeling smoke plume model discrepancy



UKESM1 model output

Additive error vs. transport error

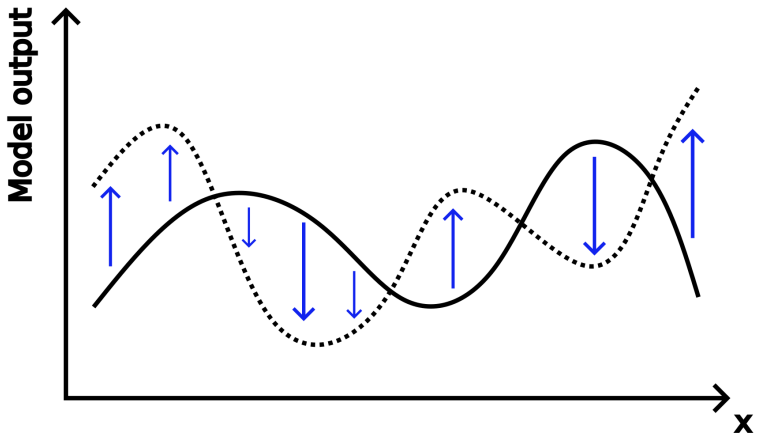


Figure: Vertical additive error

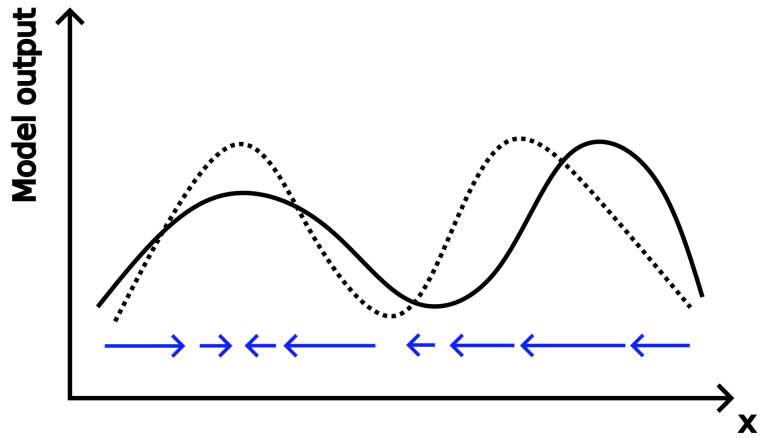


Figure: Horizontal transport error

Spatial transport error model

Our idea: Capture model discrepancy by transport the spatial field using a random transport map

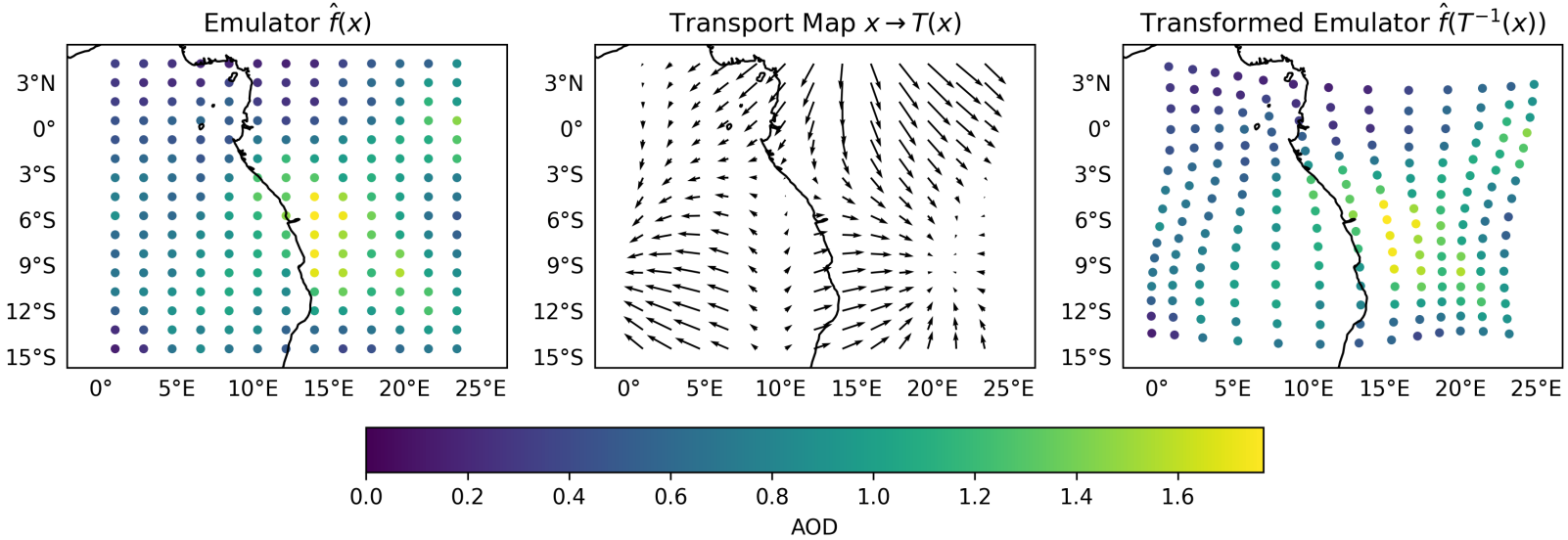
So we replace the original model

$$z(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}; \boldsymbol{u}) + \varepsilon_{\boldsymbol{x},\boldsymbol{u},emu} + \varepsilon_{\boldsymbol{x},meas} + \varepsilon_{\boldsymbol{x},other}$$

with the following model:

$$z(\boldsymbol{x}) = |\det \mathbf{J}[T^{-1}(\boldsymbol{x})]| \left(\hat{f}(T^{-1}(\boldsymbol{x}); \boldsymbol{u}) + \varepsilon_{T^{-1}(\boldsymbol{x}),\boldsymbol{u},emu}\right) + \varepsilon_{\boldsymbol{x},meas} + \varepsilon_{\boldsymbol{x},other},$$

where $T(\cdot)$ is a random transport map



Transport maps from convex Gaussian processes

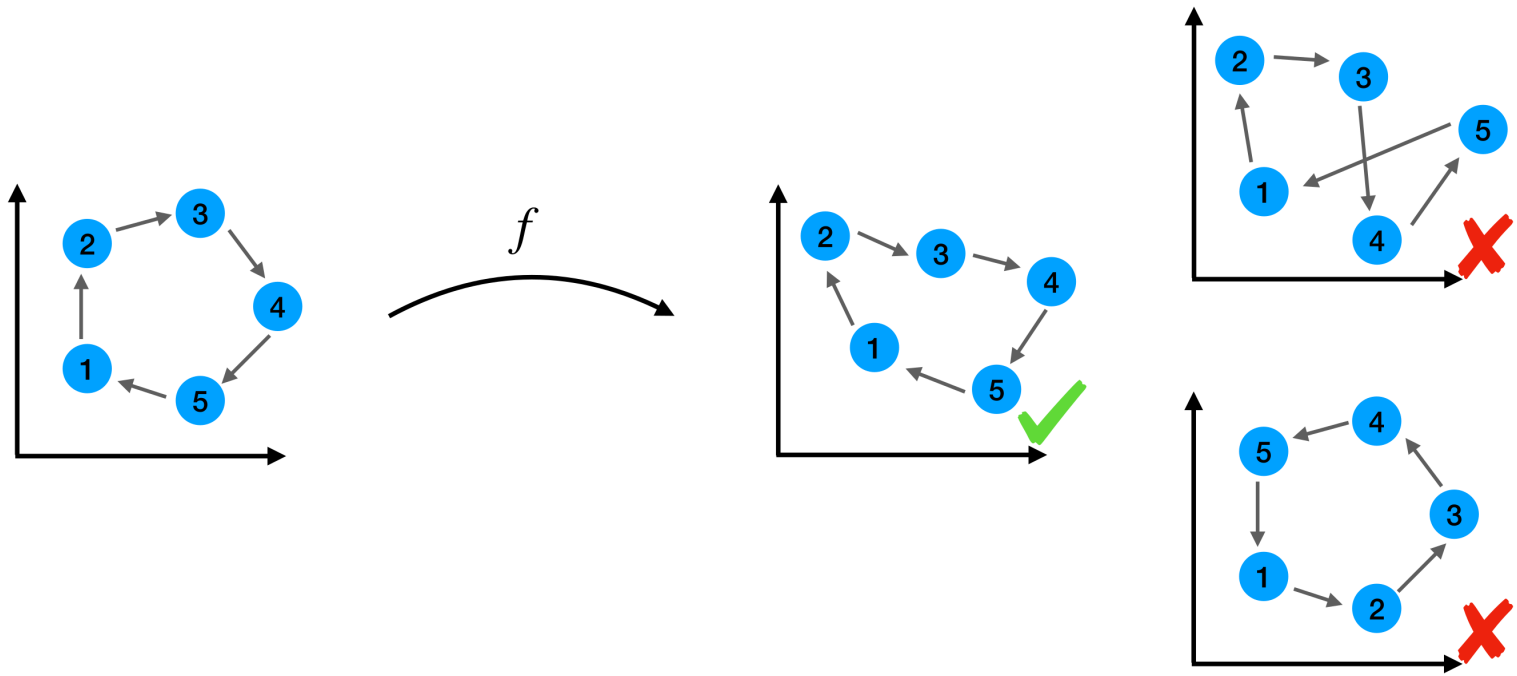
We model the transport maps as gradients of a convex Gaussian process:

$$T(\boldsymbol{x}) = \nabla\phi(\boldsymbol{x}), \quad \phi(\boldsymbol{x}) \sim CGP(\mu, K),$$

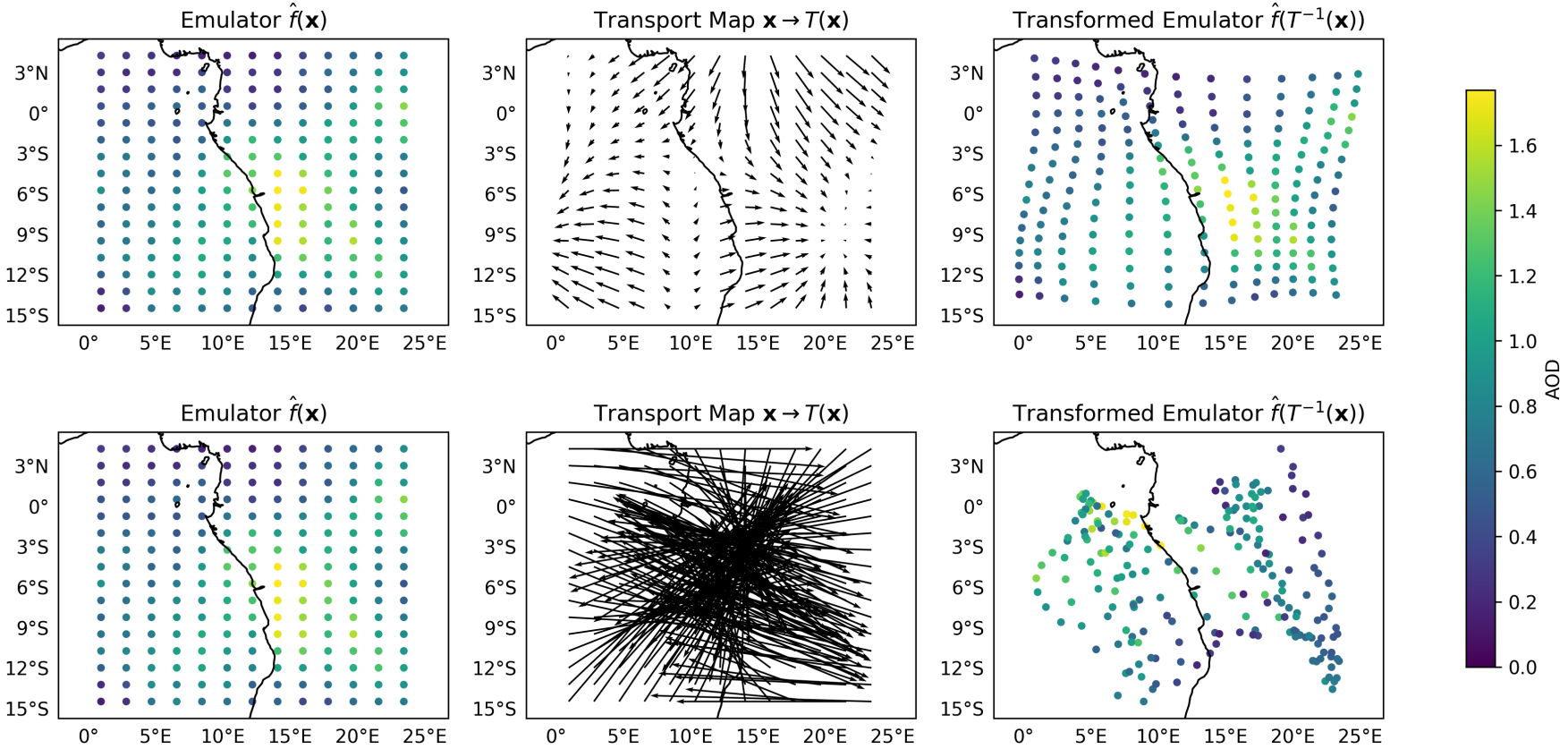
where $CGP(\mu, K)$ is the restriction of $GP(\mu, K)$ to convex functions

Why convex Gaussian process?

$\Rightarrow$ Rockafellar (1966): f is cyclically monotone if and only if $f = \nabla\phi$, where ϕ is convex



Transport maps from convex Gaussian processes



Convex ϕ vs. generic ϕ

Inference for transport error model

The likelihood is given by

$$L(\boldsymbol{u}, \boldsymbol{\nu}, \boldsymbol{\psi}) = p(\boldsymbol{z}|\boldsymbol{u}, \boldsymbol{\nu}, \boldsymbol{\psi}) = \int p(\boldsymbol{z}|T, \boldsymbol{u}, \boldsymbol{\nu}) p(T|\boldsymbol{\psi}) dT,$$

where $\boldsymbol{u}$ are the climate model parameters, $\boldsymbol{\nu} = (\delta, \ell_x^\nu, \ell_y^\nu)$ are the additive error covariance parameters in $\varepsilon_{\mathbf{x},other} \sim \mathrm{GP}(0, K_{\boldsymbol{\nu}})$ and $\boldsymbol{\psi} = (\sigma, \ell_x, \ell_y)$ are the parameters of the CGP covariance function

$$K_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{x}')\right), \quad \Sigma = \mathrm{diag}([\ell_x^2 \;\; \ell_y^2])$$

Inference for transport error model

$$L(\boldsymbol{u}, \boldsymbol{\nu}, \boldsymbol{\psi}) = p(\boldsymbol{z}|\boldsymbol{u}, \boldsymbol{\nu}, \boldsymbol{\psi}) = \int p(\boldsymbol{z}|T, \boldsymbol{u}, \boldsymbol{\nu}) p(T|\boldsymbol{\psi}) dT$$

Problem: Intractable likelihood due to truncation to convex functions

- $p(T|\boldsymbol{\psi})$ not available in closed form
- Complicated and high-dimensional integration domain

Solution: *Neural simulation-based inference* (Cranmer et al., 2016, 2020) enables estimating this kind of intractable integrals based on samples of $(\boldsymbol{z}_i, \boldsymbol{u}_i, \boldsymbol{\nu}_i, \boldsymbol{\psi}_i)$

As long as it is fast to simulate $\boldsymbol{z}_i$ for given $(\boldsymbol{u}_i, \boldsymbol{\nu}_i, \boldsymbol{\psi}_i)$, it is possible to train a neural network to learn a neural surrogate model for the intractable likelihood $L(\boldsymbol{u}, \boldsymbol{\nu}, \boldsymbol{\psi})$

Exact HMC for Sampling Transport Maps

Define an extended Gaussian process $\tilde{\phi}$ with known analytic covariance

$$\tilde{\phi}(\mathbf{x}) = [\phi_x(\mathbf{x}), \phi_y(\mathbf{x}), \phi_{xx}(\mathbf{x}), \phi_{xy}(\mathbf{x}), \phi_{yy}(\mathbf{x})]^T$$

Enforce convexity pointwise at observed locations $\mathcal{X}$:

$$\phi_{xx}(\mathbf{x}) \geq 0, \qquad \phi_{xx}(\mathbf{x})\phi_{yy}(\mathbf{x}) - \phi_{xy}(\mathbf{x})^2 \geq 0, \qquad \forall \mathbf{x} \in \mathcal{X}$$

Reduces to sampling a quadratically constrained multivariate Gaussian

$\Rightarrow$ Can use *exact Hamiltonian Monte Carlo* (Pakman and Paninski, 2014)

- Gaussian HMC trajectories are analytically computable
- Analytic solution to hit times with constraint boundaries
- Elastically reflect HMC trajectory off of boundaries
- General-purpose Python implementation available in package `tmg-hmc` from PyPI (Bensen and Kuusela, 2026)

Samples exactly what our model needs:

$$T(\mathbf{x}) = [\phi_x(\mathbf{x}), \phi_y(\mathbf{x})]^T, \qquad \det \mathbf{J}\, T(\mathbf{x}) = \phi_{xx}(\mathbf{x})\phi_{yy}(\mathbf{x}) - \phi_{xy}(\mathbf{x})^2$$

Speeding Up Sampling

Analytic hit time algebraic complexity significantly slows down sampling

$\Rightarrow$ Converted to C++ and used an LLM to remove redundant computations

- Achieved about 100x speedup

The Classifier Trick and Neural Density Ratios

Neural network classifiers are a powerful tool for simulation-based inference (Cranmer et al., 2016)

Consider two classes:

$$\mathcal{C}_1: \quad \mathbf{z}_1^1, \mathbf{z}_2^1, \ldots, \mathbf{z}_n^1 \overset{\text{iid}}{\sim} p_1, \quad \mathcal{C}_2: \quad \mathbf{z}_1^2, \mathbf{z}_2^2, \ldots, \mathbf{z}_m^2 \overset{\text{iid}}{\sim} p_2$$

Train a classifier $h(\mathbf{z})$ to separate $\mathcal{C}_1$ from $\mathcal{C}_2$ by minimizing the binary cross-entropy loss; then the classifier will asymptotically learn:

$$h(\mathbf{z}) = P(\mathcal{C}_1|\mathbf{z}) = \frac{p(\mathbf{z}|\mathcal{C}_1)P(\mathcal{C}_1)}{p(\mathbf{z}|\mathcal{C}_1)P(\mathcal{C}_1) + p(\mathbf{z}|\mathcal{C}_2)P(\mathcal{C}_2)} = \frac{\lambda \frac{p(\mathbf{z}|\mathcal{C}_1)}{p(\mathbf{z}|\mathcal{C}_2)}}{\lambda \frac{p(\mathbf{z}|\mathcal{C}_1)}{p(\mathbf{z}|\mathcal{C}_2)} + 1} = \frac{\frac{p_1(\mathbf{z})}{p_2(\mathbf{z})}}{\frac{p_1(\mathbf{z})}{p_2(\mathbf{z})} + \frac{1}{\lambda}}$$

Solving for $\frac{p_1(\mathbf{z})}{p_2(\mathbf{z})}$ gives $\frac{p_1(\mathbf{z})}{p_2(\mathbf{z})} = \lambda^{-1} \frac{h(\mathbf{z})}{1-h(\mathbf{z})} = \lambda^{-1}\text{odds}(h(\mathbf{z}))$ where $\lambda = \frac{P(\mathcal{C}_1)}{P(\mathcal{C}_2)}$

$\Rightarrow$ The classifier $h(\mathbf{z})$ implicitly learns the density ratio $\frac{p_1(\mathbf{z})}{p_2(\mathbf{z})}$

Neural Likelihood

We can strategically pick the classes $\mathcal{C}_1$ and $\mathcal{C}_2$ to learn the likelihood (e.g., Walchessen et al., 2024)

Let $\boldsymbol{\theta}$ be our parameter vector, $\mathbf{z} \sim p_{\boldsymbol{\theta}}$ and choose classes:

$$\mathcal{C}_1: \quad (\mathbf{z}_1, \boldsymbol{\theta}_1), \ldots, (\mathbf{z}_n, \boldsymbol{\theta}_n) \overset{\text{iid}}{\sim} p(\mathbf{z}, \boldsymbol{\theta}),$$
$$\mathcal{C}_2: \quad (\mathbf{z}_1, \boldsymbol{\theta}_1), \ldots, (\mathbf{z}_n, \boldsymbol{\theta}_n) \overset{\text{iid}}{\sim} p(\mathbf{z})p(\boldsymbol{\theta})$$

Letting $P(\mathcal{C}_1) = P(\mathcal{C}_2) = 1/2$ and following the previous algebra we get:

$$\text{odds}(h(\mathbf{z}, \boldsymbol{\theta})) = \frac{p(\mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{z})} \propto \mathcal{L}(\boldsymbol{\theta}; \mathbf{z})$$

Issue: Empirically struggles to learn moderate-high dimensional $\boldsymbol{\theta}$

- Our setting: $\boldsymbol{\theta} = [u_1, u_2, \sigma, \ell_x, \ell_y, \delta, \ell_x^v, \ell_y^v]^\mathsf{T} \in \mathbb{R}^8$
- Model predominantly learns "easy" δ dependence

$\Rightarrow$ Learn separate dependence with Telescoping Ratio Estimators
Presented later this session by Dan Leonte (Leonte et al., 2025)

Nuisance parameter profile likelihoods

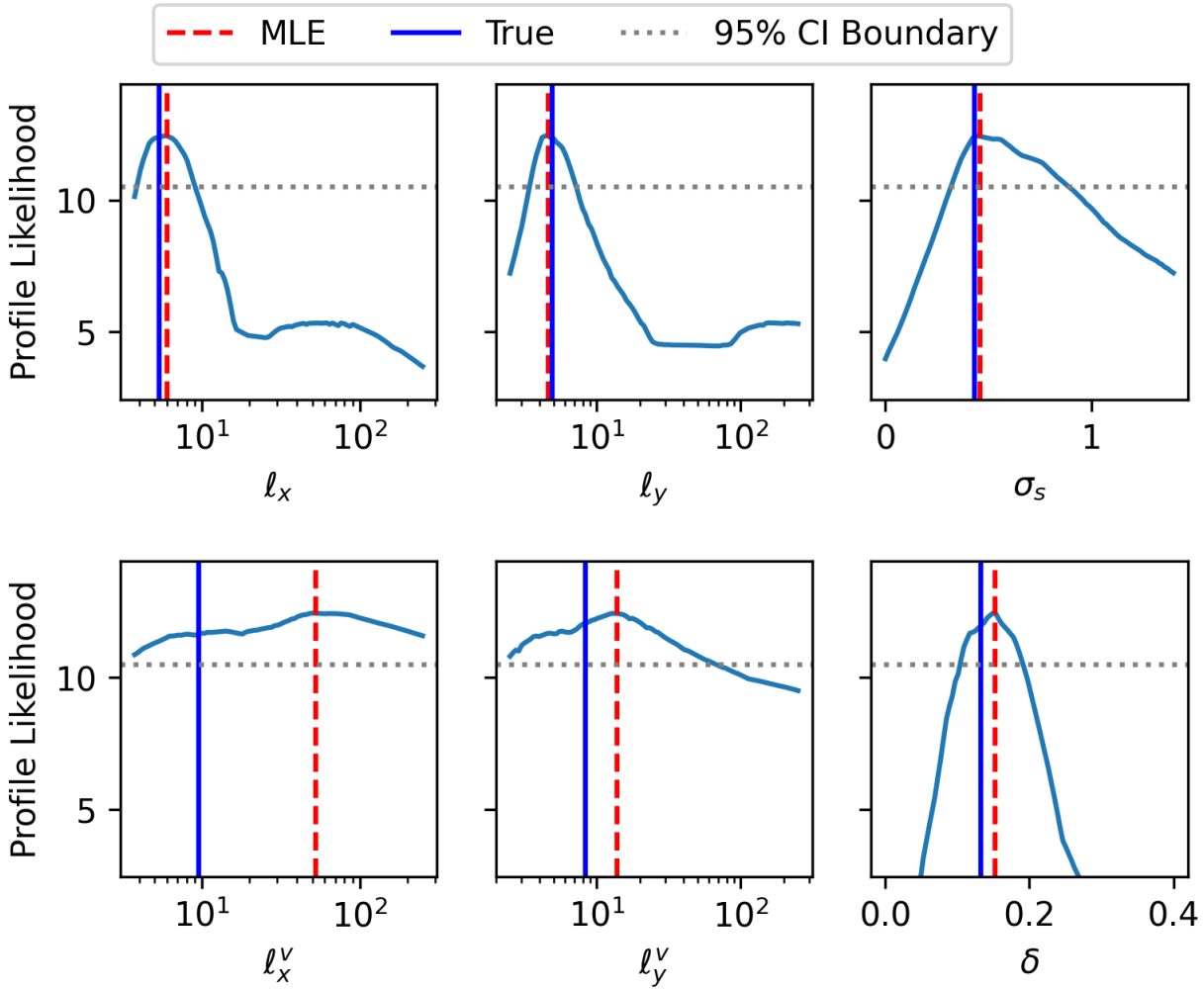
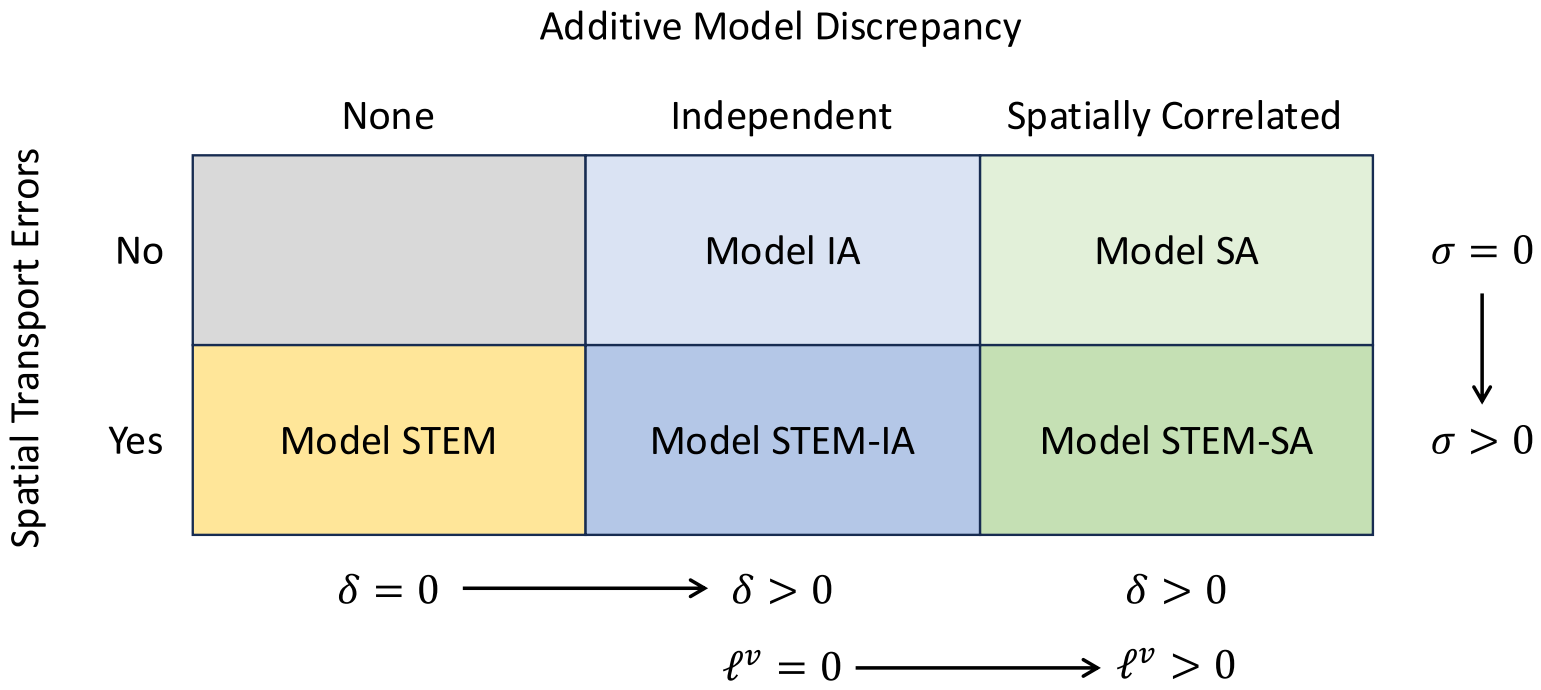


Figure: Neural profile likelihood curves for the nuisance parameters fit to simulated satellite observations.

Model selection

We can consider 5 different model discrepancy models:

- IA = independent additive
- SA = spatial additive
- STEM = spatial transport
- STEM-IA = spatial transport & independent additive
- STEM-SA = spatial transport & spatial additive



Neural LR Test for Model Selection

To assess the significance of the spatial transport error model, we can test:

$$H_0: \quad \sigma = 0 \qquad \text{vs.} \qquad H_1: \quad \sigma > 0$$

We do this using a neural likelihood ratio test

$$\lambda(\mathbf{z}) = \sup_{\boldsymbol{\theta} \in \Theta} \log \hat{\mathcal{L}}(\boldsymbol{\theta}; \mathbf{z}) - \sup_{\boldsymbol{\theta} \in \Theta_0} \log \hat{\mathcal{L}}(\boldsymbol{\theta}; \mathbf{z}),$$

where $\hat{\mathcal{L}}$ is the neural surrogate likelihood

We use a bootstrap sample $\mathbf{z}_1^*, \ldots, \mathbf{z}_B^* \overset{\text{iid}}{\sim} p_{\hat{\boldsymbol{\theta}}_0}$, where $\hat{\boldsymbol{\theta}}_0 = \arg\max_{\boldsymbol{\theta} \in \Theta_0} \mathcal{L}(\boldsymbol{\theta}; \mathbf{z})$, to obtain the null distribution of λ ($\mathcal{L}$ is known analytically for H_0)

STEM Model Significance Results

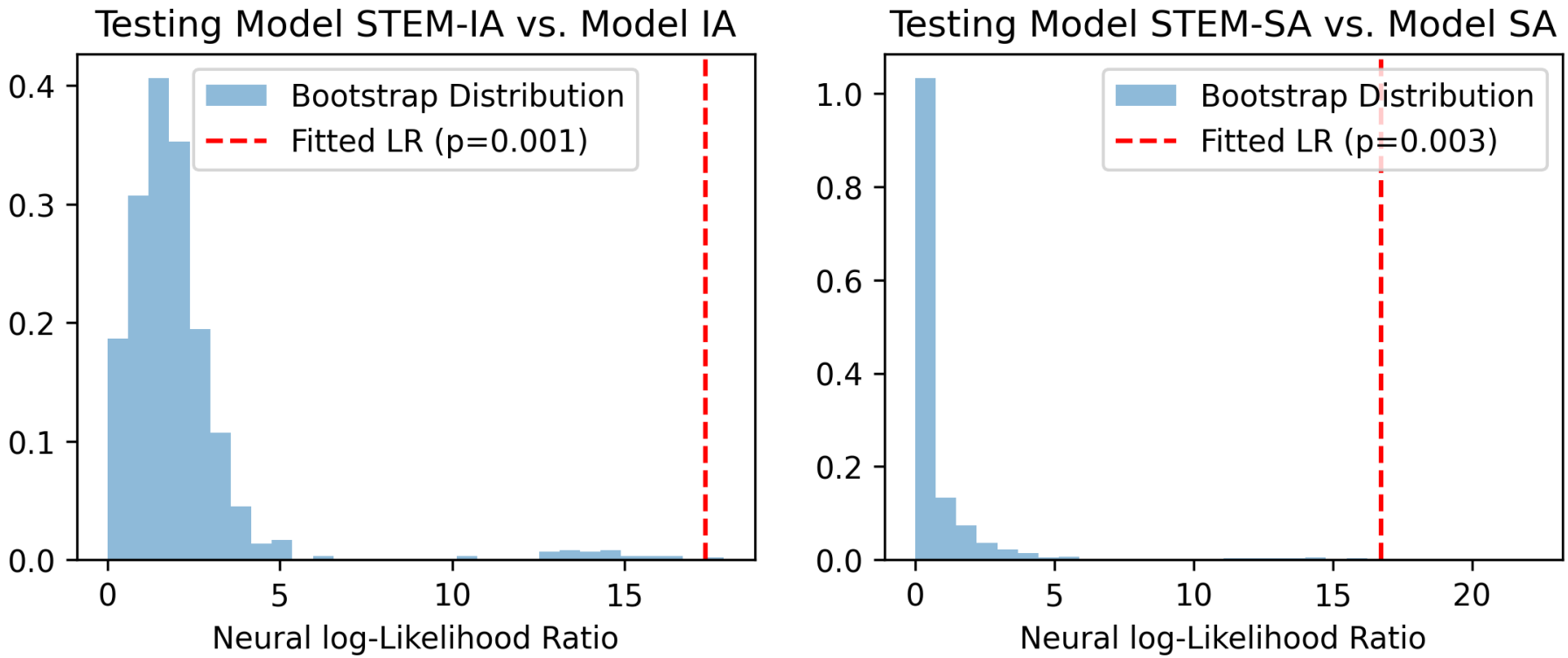
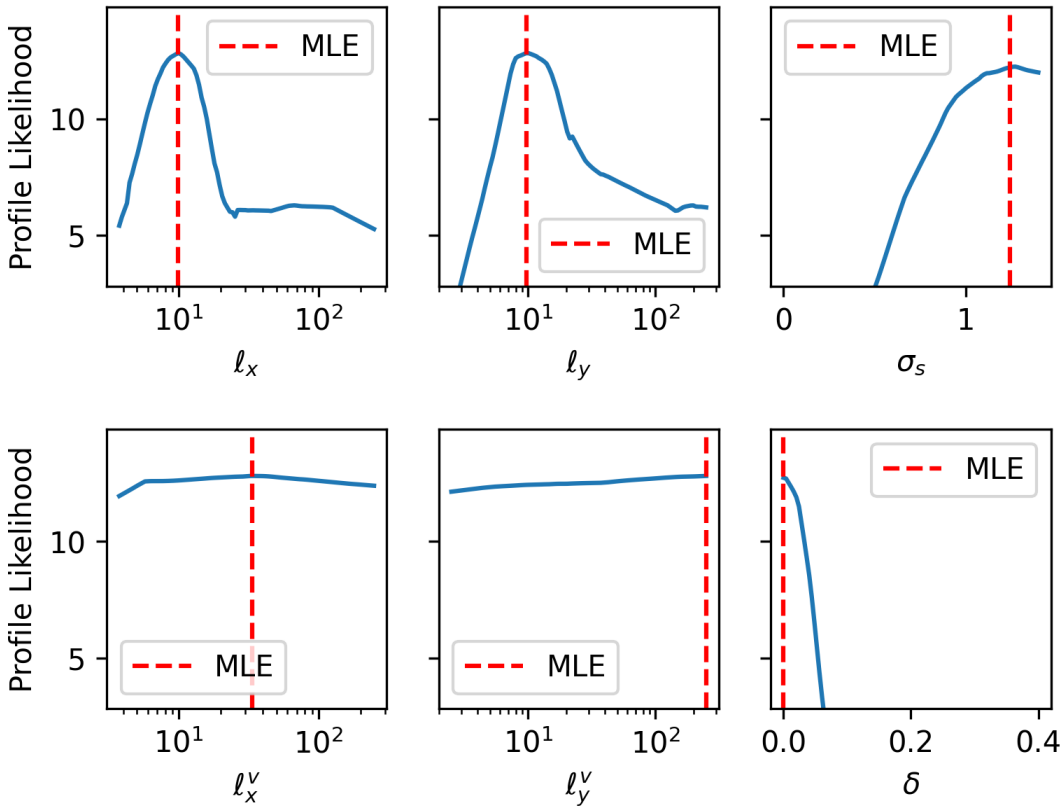


Figure: Histograms of the bootstrapped neural likelihood ratio statistics for testing $H_0 : \sigma = 0$ vs. $H_1 : \sigma > 0$. (left) STEM-IA vs. IA (right) STEM-SA vs. SA

We reject IA in favor of STEM-IA and SA in favor of STEM-SA

Additive Discrepancy Significance Results



In addition, in both STEM-IA and STEM-SA, the neural MLE of the additive error standard deviation is $\hat{\delta} = 0$

$\Rightarrow$ We choose STEM as the best-fitting model

Best-fitting transport map

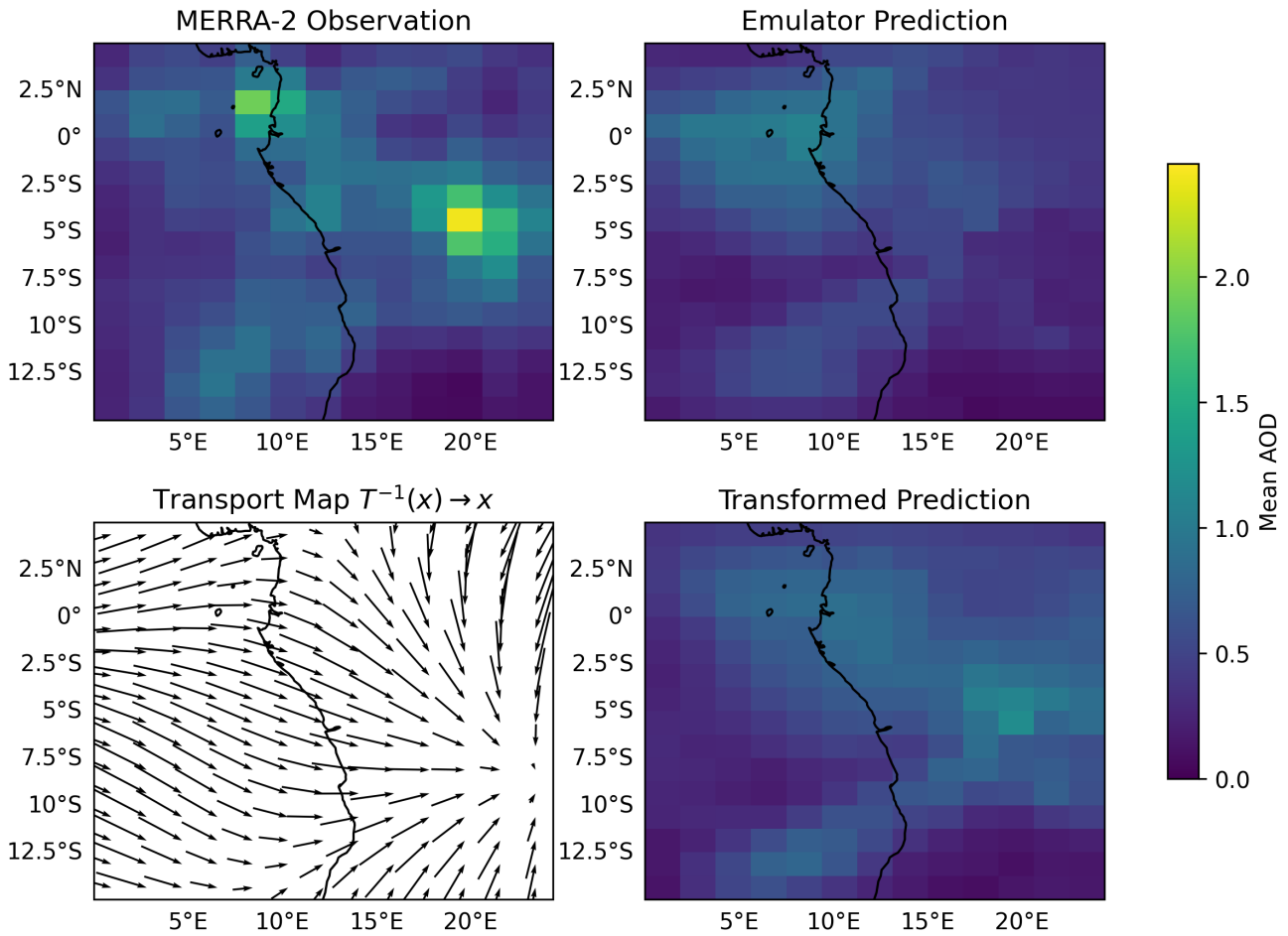


Figure: Visualization of the best-fitting emulator and transport map for model SW

Standardized residuals

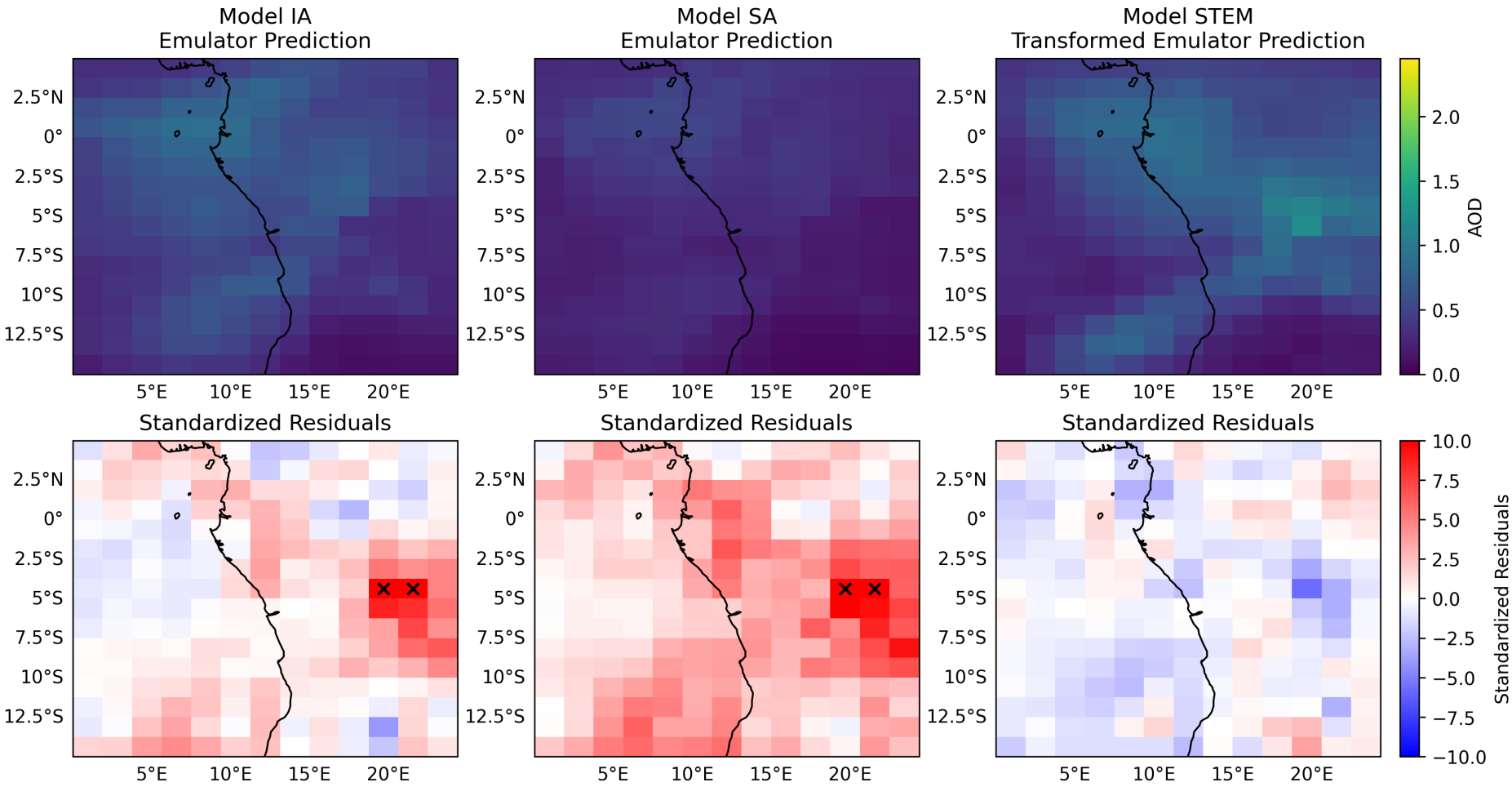


Figure: Best-fitting emulator predictions (top row) for models IA (left), SA (middle) and SW (right) and the associated standardized residuals (bottom row)

Aerosol parameter profile likelihoods

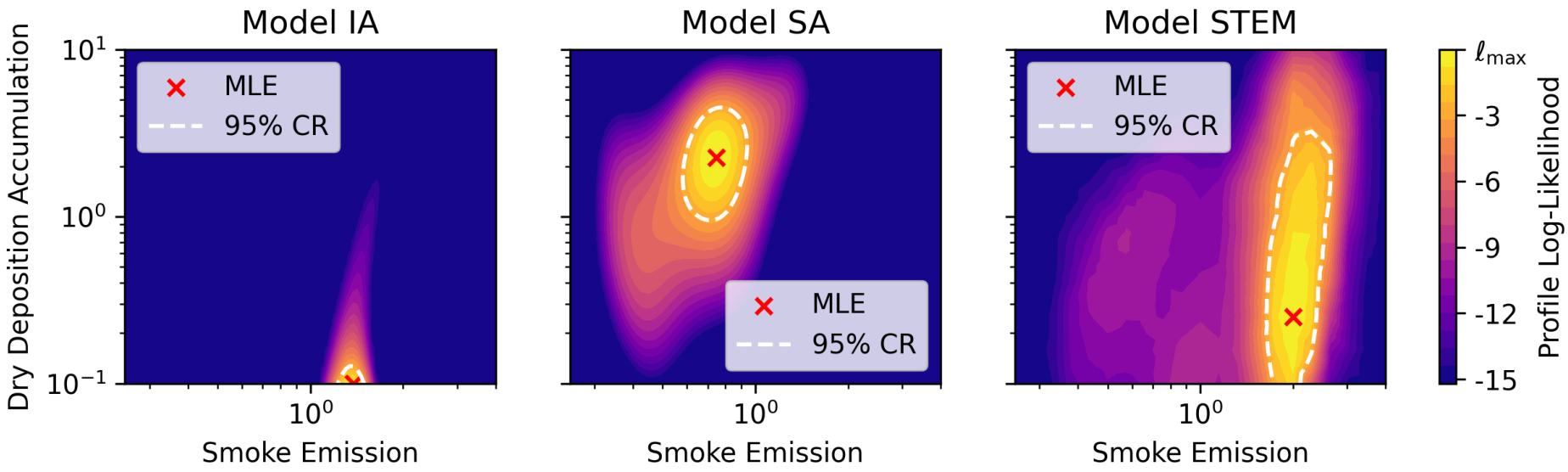


Figure: Profile likelihood surfaces for models IA (left), SA (middle) and SW (right) as a function of the climate model parameters $\boldsymbol{u}$ and the associated approximate 95% confidence regions

Discussion and conclusions

- PPEs can be used to observationally constrain aerosol uncertainties but model discrepancy is a major challenge
- We proposed a spatial transport error model to complement additive model discrepancy terms
- Exact inference is intractable due to an intractable likelihood function
- Exact Hamiltonian Monte Carlo enables fast and accurate simulation from the model so we can use SBI to train a neural surrogate model for the likelihood
- TRE allows us to handle higher parameter dimensions
- Spatial transport error model provides a better fit to the data and plausible UQ
- Different ways of handling the model discrepancy can lead to drastically different UQ results

Questions?

References I

E. A. Bensen and M. Kuusela. tmg_hmc: A python package for exact hmc sampling for truncated multivariate gaussians with linear and quadratic constraints. *Journal of Open Source Software*, 2026. In Review.

J. Brynjarsdòttir and A. O'Hagan. Learning about physical parameters: The importance of model discrepancy. *Inverse problems*, 30(11):114007, 2014.

J. Carzon, B. Abreu, L. Regayre, K. Carslaw, L. Deaconu, P. Stier, H. Gordon, and M. Kuusela. Statistical constraints on climate model parameters using a scalable cloud-based inference framework. *Environmental Data Science*, 2:e24, 2023.

J. Carzon, B. Abreu, L. Regayre, K. Carslaw, L. Deaconu, P. Stier, H. Gordon, and M. Kuusela. Statistical constraints on climate model parameters using a scalable cloud-based inference framework – Corrigendum. *Environmental Data Science*, 4:e14, 2025.

K. Cranmer, J. Pavez, and G. Louppe. Approximating likelihood ratios with calibrated discriminative classifiers, 2016. URL `https://arxiv.org/abs/1506.02169`.

K. Cranmer, J. Brehmer, and G. Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

References II

M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

D. Leonte, R. Huser, and A. E. D. Veraart. Simulation-based inference via telescoping ratio estimation for trawl processes, Oct. 2025. URL `http://arxiv.org/abs/2510.04042`. arXiv:2510.04042 [stat].

A. Pakman and L. Paninski. Exact Hamiltonian Monte Carlo for Truncated Multivariate Gaussians. *Journal of Computational and Graphical Statistics*, 23(2):518–542, Apr. 2014. ISSN 1061-8600. Taylor & Francis.

R. Rockafellar. Characterization of the subdifferentials of convex functions. *Pacific Journal of Mathematics*, 17(3):497–510, June 1966. ISSN 0030-8730. Mathematical Sciences Publishers.

P. D. Sampson and P. Guttorp. Nonparametric Estimation of Nonstationary Spatial Covariance Structure. *Journal of the American Statistical Association*, 87(417): 108–119, Mar. 1992. ISSN 0162-1459. Publisher: ASA Website _eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1992.10475181.

J. Walchessen, A. Lenzi, and M. Kuusela. Neural likelihood surfaces for spatial processes with computationally intensive or intractable likelihoods. *Spatial Statistics*, 62: 100848, 2024.

Backup

Random Transport Map Distribution

Define $CGP(\mu, K_\nu)$ as the restriction of $GP(\mu, K_\nu)$ to convex functions

$$\phi \sim CPG(\mu, K_\nu) \qquad T(\mathbf{x}) = \nabla\phi(\mathbf{x}) \qquad \mathbf{J}T(\mathbf{x}) = \mathbf{H}\phi(\mathbf{x})$$

Mean Function

$$\mu(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$$

Covariance Kernel

$$\mathbf{K}_\psi(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{(x - x')^2}{\ell_x^2} - \frac{(y - y')^2}{\ell_y^2}\right)$$

- Identity most likely
- Analogous to zero mean additive error

- Smooth transport maps
- $\sigma = 0$ reduces to additive error only model

Contrast with Sampson and Guttorp (1992)

Sampson and Guttorp (1992) details a nonparametric method of estimating the covariance kernel for nonstationary spatial processes.

- Involves deforming the plane G where the kernel is nonstationary into a plane D where the kernel is stationary

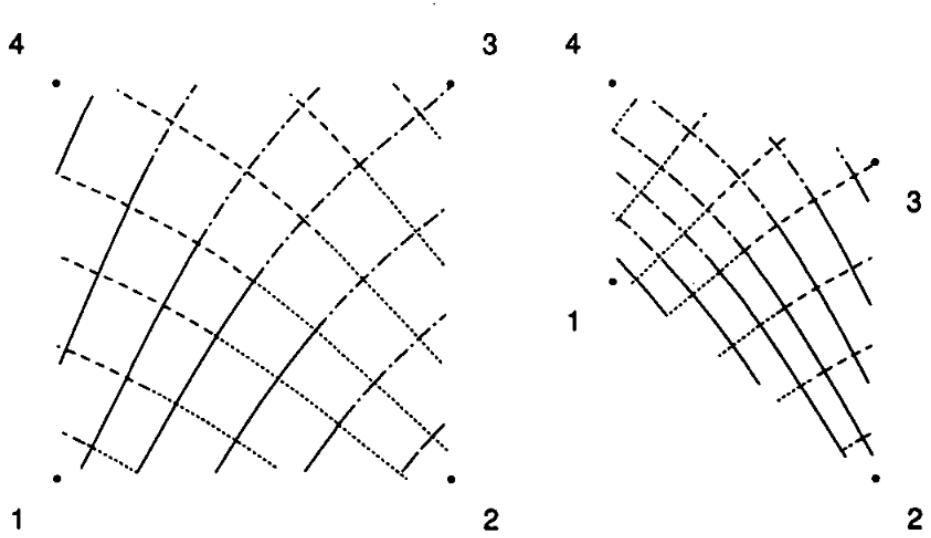


Figure: Nonlinear deformation from the G plane (left) to the D plane (right). Sampson and Guttorp (1992) Figure 2.

Contrast with Sampson and Guttorp (1992)

Key Distinctions

Goal:

- Our work: model a systematic error in climate simulators that is not captured by additive error models
- Sampson and Guttorp (1992): Estimate nonstationary covariance kernel of a spatially correlated additive error model

Distortion Behavior

- Our work: Transport map is *random* and has a different realization from the same distribution at every time
- Sampson and Guttorp (1992): Deformation is deterministic and is constant across time

Telescoping Ratio Estimation

Telescoping ratio estimators (TREs) allow us to separately learn the dependence on each parameter with fewer samples (Leonte et al., 2025)

Key Idea 1: $p(\boldsymbol{\theta})$ is known and typically a product distribution

$$\frac{p(\mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{z})} = \frac{p(\mathbf{z},\boldsymbol{\theta})}{p(\mathbf{z})p(\boldsymbol{\theta})} = \frac{p(\boldsymbol{\theta}|\mathbf{z})}{p(\boldsymbol{\theta})} = \frac{p(\ell,\sigma,\ell^{\nu},\delta,\boldsymbol{u}|\mathbf{z})}{p(\ell)p(\sigma)p(\ell^{\nu})p(\delta)p(\boldsymbol{u})}$$

Key Idea 2: decompose the joint distribution into its conditionals

$$= \frac{p(\ell|\sigma,\ell^{\nu},\delta,\boldsymbol{u},\mathbf{z})}{p(\ell)} \frac{p(\sigma|\ell^{\nu},\delta,\boldsymbol{u},\mathbf{z})}{p(\sigma)} \frac{p(\ell^{\nu}|\delta,\boldsymbol{u},\mathbf{z})}{p(\ell^{\nu})} \frac{p(\delta|\boldsymbol{u},\mathbf{z})}{p(\delta)} \frac{p(\boldsymbol{u}|\mathbf{z})}{p(\boldsymbol{u})}$$

Using Bayes rule, we can write this as:

$$= \frac{p(\sigma,\ell^{\nu},\delta,\boldsymbol{u},\mathbf{z}|\ell)}{p(\sigma,\ell^{\nu},\delta,\boldsymbol{u},\mathbf{z})} \frac{p(\ell^{\nu},\delta,\boldsymbol{u},\mathbf{z}|\sigma)}{p(\ell^{\nu},\delta,\boldsymbol{u},\mathbf{z})} \frac{p(\delta,\boldsymbol{u},\mathbf{z}|\ell^{\nu})}{p(\delta,\boldsymbol{u},\mathbf{z})} \frac{p(\boldsymbol{u},\mathbf{z}|\delta)}{p(\boldsymbol{u},\mathbf{z})} \frac{p(\mathbf{z}|\boldsymbol{u})}{p(\mathbf{z})}$$

Telescoping Ratio Estimation

The TRE decomposition allows us to train 5 separate classifiers:

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{z}) \propto \overbrace{\frac{p(\sigma, \ell^v, \delta, \boldsymbol{u}, \mathbf{z}|\ell)}{p(\sigma, \ell^v, \delta, \boldsymbol{u}, \mathbf{z})}}^{\ell \text{ Classifier}} \underbrace{\frac{p(\ell^v, \delta, \boldsymbol{u}, \mathbf{z}|\sigma)}{p(\ell^v, \delta, \boldsymbol{u}, \mathbf{z})}}_{\sigma \text{ Classifier}} \overbrace{\frac{p(\delta, \boldsymbol{u}, \mathbf{z}|\ell^v)}{p(\delta, \boldsymbol{u}, \mathbf{z})}}^{\ell^v \text{ Classifier}} \underbrace{\frac{p(\boldsymbol{u}, \mathbf{z}|\delta)}{p(\boldsymbol{u}, \mathbf{z})}}_{\delta \text{ Classifier}} \overbrace{\frac{p(\mathbf{z}|\boldsymbol{u})}{p(\mathbf{z})}}^{\boldsymbol{u} \text{ Classifier}}$$

- Each only learns dependence on 1-2 parameters
- Model forced to learn dependence on all parameters