Locally Adaptive Conformal Inference for Operator Models

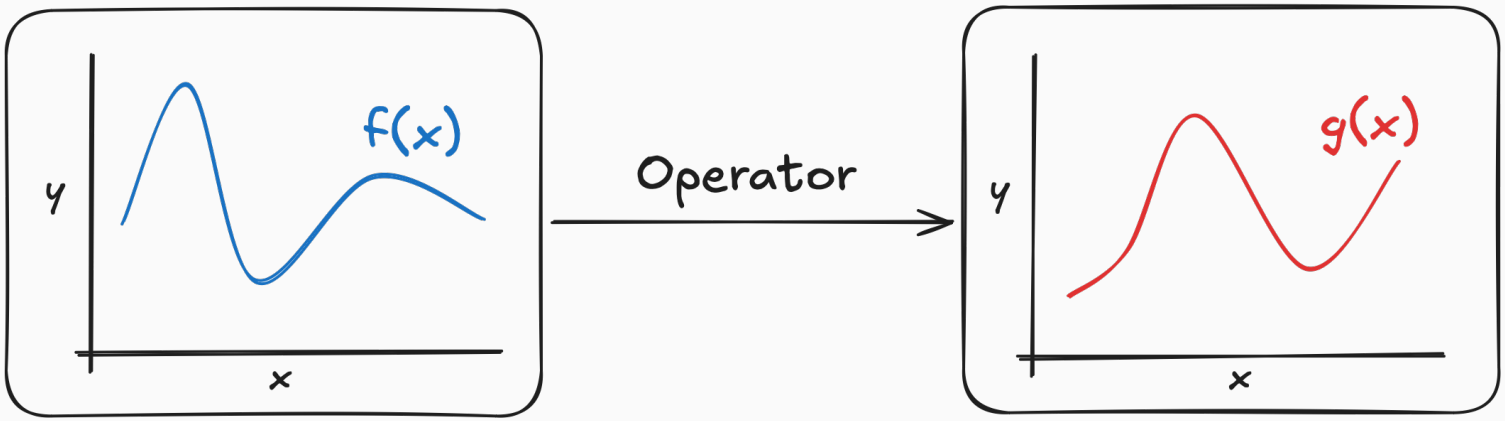
Trevor Harris

October 24, 2025

University of Connecticut
Department of Statistics

Carnegie Mellon University – STAMPS, PA

Operators and Operator Models

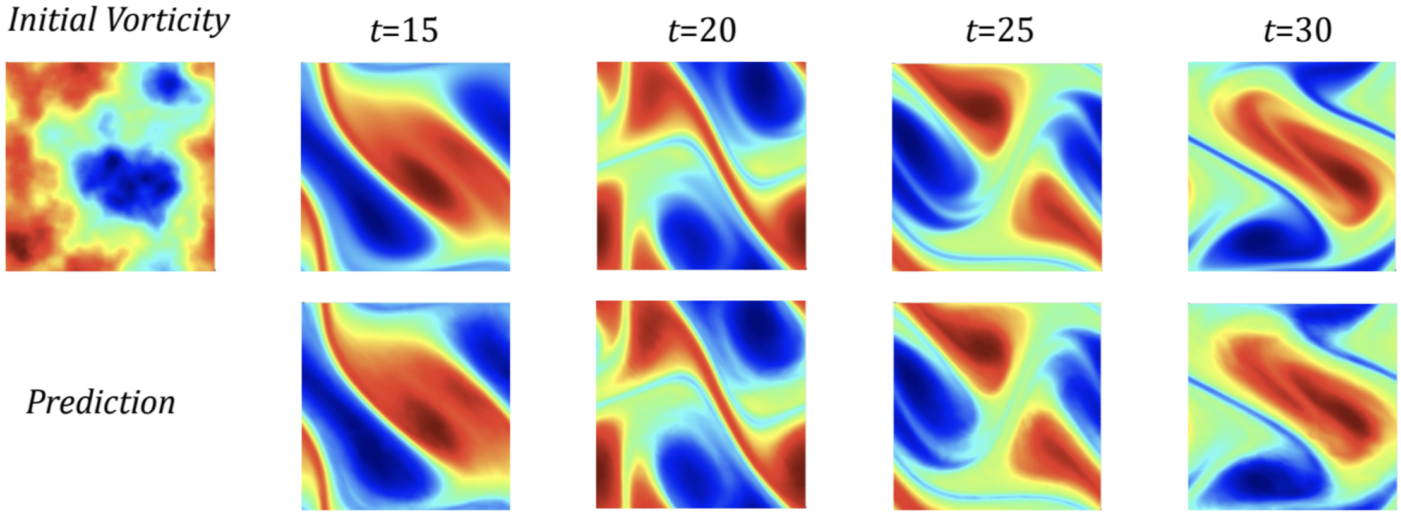


- An **operator** is a map between *functions*
  - Takes a function f and returns a function g (e.g. $\nabla[\sin(x)] = \cos(x)$)
- An **operator model** is a regression model between functions

$$\Gamma_\theta : \mathcal{F} \mapsto \mathcal{G} \tag{1}$$

  - $\mathcal{F}$ and $\mathcal{G}$ are function spaces, e.x. $\mathcal{F} = \mathcal{G} = \mathcal{L}^2([0,1])$
  - Extends regression to functional setting (inputs / outputs are functions)
  - $\theta \in \Theta$ denotes the model parameters

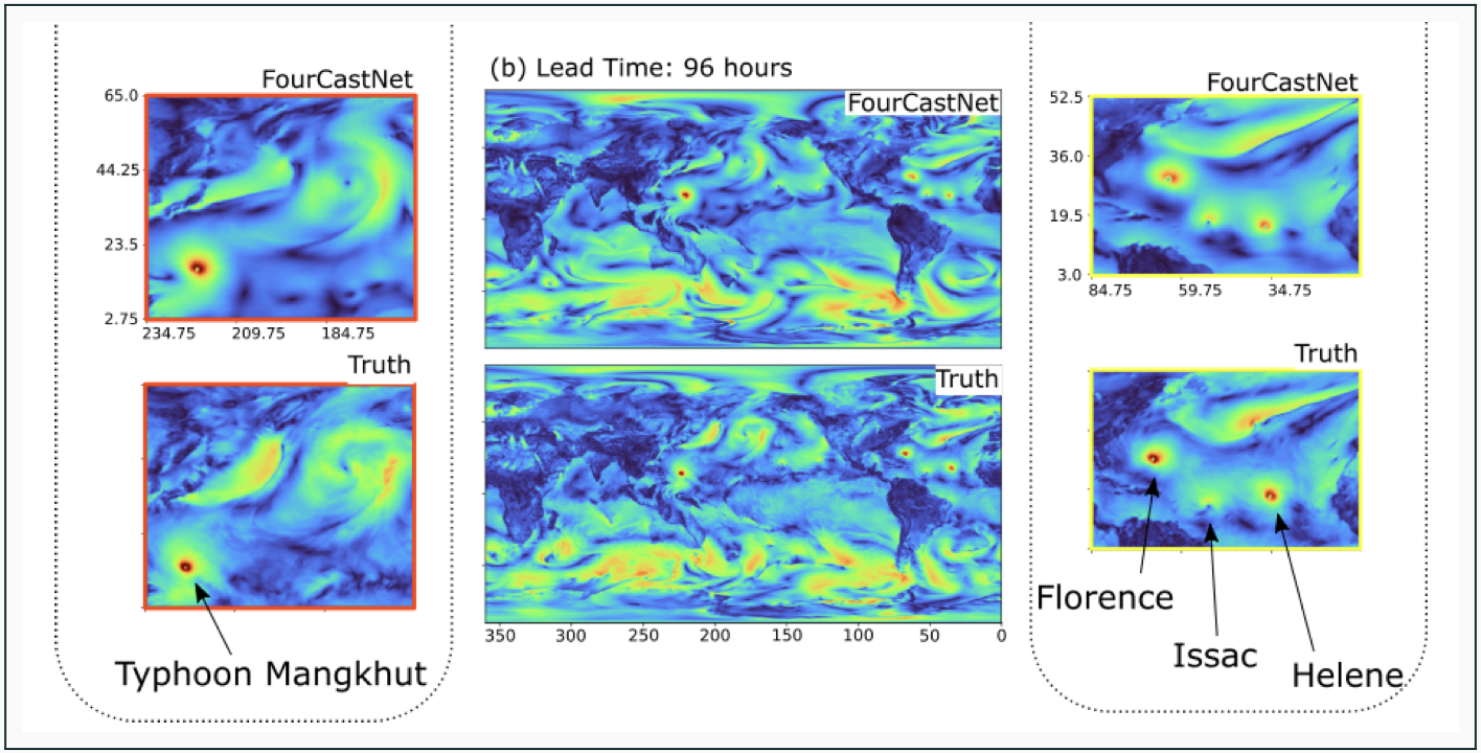
Operator learning - PDE Simulations



Zero-shot super-resolution: Navier-Stokes Equation with viscosity $\nu = 1e{-}4$; Ground truth on top and prediction on bottom; trained on $64 \times 64 \times 20$ dataset; evaluated on $256 \times 256 \times 80$ (see Section 5.4).

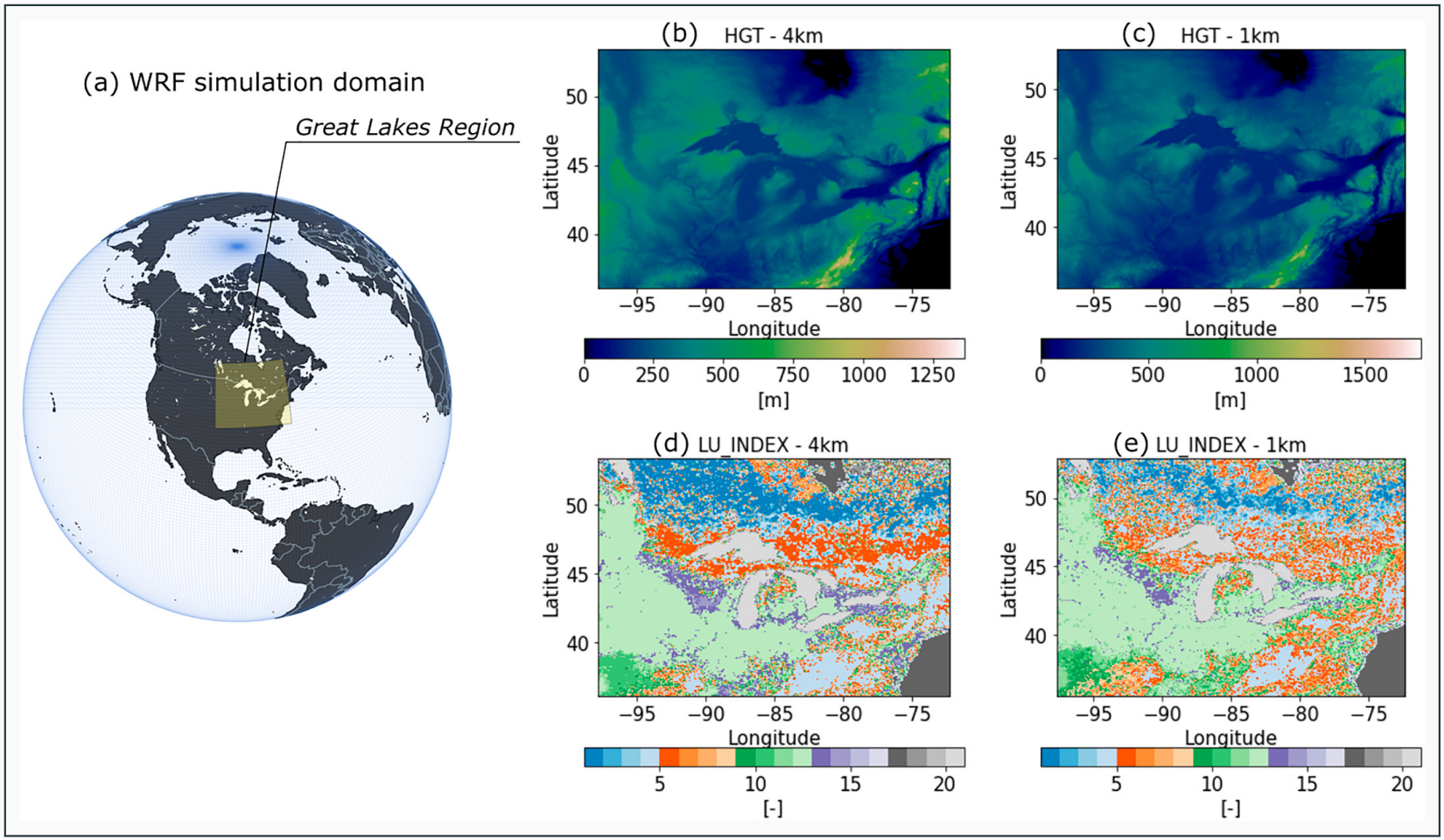
- [Li et al. (2020)] used neural operators to solve high resolution Navier-Stokes equations from low resolution training data
- Neural operator model ran 1000x times faster with nearly free zero-shot super-resolution

Operator learning - Weather Forecasting



- [Pathak et al. (2022)] multi-channel neural operator to forecast global weather patterns (temperature, humidity, wind velocity, etc.)
- Matches the skill of ECMWF Integrated Forecasting System (IFS) at a fraction of the cost

Operator learning - Super-resolution



- [Jiang et al. (2023)] used neural operators for high resolution downscaling of regional climate model output
- Matches the skill of dynamically downscaled (WRF) output

Operator Learning

- Functional data analysis – Inference
  - Linear methods: e.g. [Ramsay and Dalzell (1991); Besse and Cardot (1996)]
  - Nonlinear methods: e.g. [Yao and Müller (2010); Scheipl et al. (2015)]
- Neural operators models - Deep predictive models
  - Sequence of non-linear *kernel integrations* ala DNN layers.
  - Neural Operator Models - [Chen and Chen (1995); Lu et al. (2021); Bhattacharya et al. (2021); Nelsen and Stuart (2021)], etc.
  - Computationally efficient: Fourier Neural operators [Li et al. (2020)], Spherical Operators [Bonev et al. (2023)], Wavelet Operators [Tripura and Chakraborty (2023)], and others [Lanthaler et al. (2023); Kovachki et al. (2024)]
- Neural operators lack uncertainty quantification (UQ). UQ critical for many applications (e.g. weather forecasting, climate emulation).

  $\Rightarrow$ Develop UQ for general operator models

Conformal Inference

- General framework for quantifying predictive uncertainty [Vovk et al. (2005); Lei et al. (2018)]
  - Constructs post-hoc, finite-sample statistically valid *prediction sets*
  - No asymptotics, priors, or modification to the training procedure.
  - Requires *score function* $S(f_t, g_t)$ – **exchangeable**. E.g. $S_t = \|g_t - \Gamma_{\hat{\theta}}(f_t)\|_2$

- **Alg. Sketch** (Split) Conformal Inference

  Given: Model $\Gamma_\theta : \mathcal{F} \mapsto \mathcal{G}$, $D_{tr} = \{(f_s, g_s)\}_{s=1}^m$, $D_{cal} = \{(f_t, g_t)\}_{t=1}^n$.

  1. Train on D_{tr}: $\hat{\theta} = \arg\min \mathcal{L}(f, \Gamma_\theta(g))$
  2. Compute scores on D_{cal}: $S_t = S(f_t, g_t) \quad \forall t \in 1, ..., n.$
  3. Find $q = S_{(k)}$, the k'th largest score where $k = \lceil (1-\alpha)(n+1) \rceil$
  4. For a new f_{n+1}, define

$$C_\alpha(f_{n+1}) = \{g : S(f_{n+1}, g) \leq q\}$$

  .

- q defines a "typical" range of errors, $C_\alpha(f_{n+1})$ is all functions in range.

Adaptive Conformal inference

- Standard split conformal inference works well marginally
  - $\mathbb{P}(g \in C_\alpha(f)) \geq 1 - \alpha$ even in *finite samples* (valid)
  - $\mathbb{P}(g \in C_\alpha(f)) \leq 1 - \alpha + o(n)$ (minimal)
  - Works for any prediction algorithm $\Gamma_\theta : \mathcal{F} \mapsto \mathcal{G}$ (general)
- But it is **non-adaptive** to any heterogeneity in $g \mid f$.
  - Size and shape of prediction sets fixed at calibration time
  - E.g., L^∞-Score: $C_\alpha(f_{n+1}) = \hat{\Gamma}_\theta(f_{n+1}) \pm q$, $C_\alpha(f_{n+2}) = \hat{\Gamma}_\theta(f_{n+2}) \pm q$, etc.

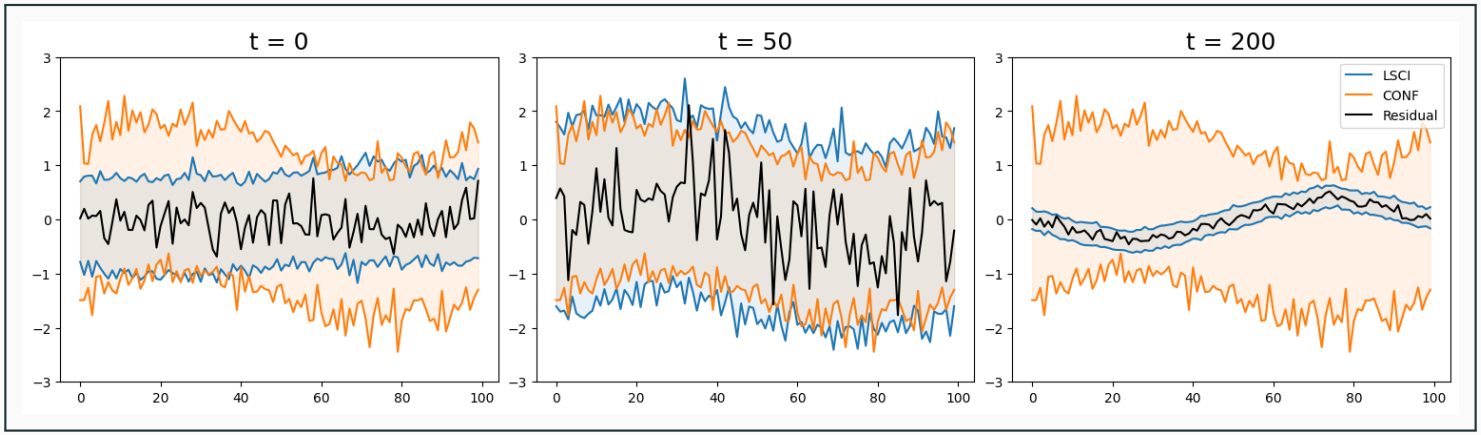


Figure 1: E.x. heterogeneous residual functions. Not captured by standard CI interval

Local Conformal Inference

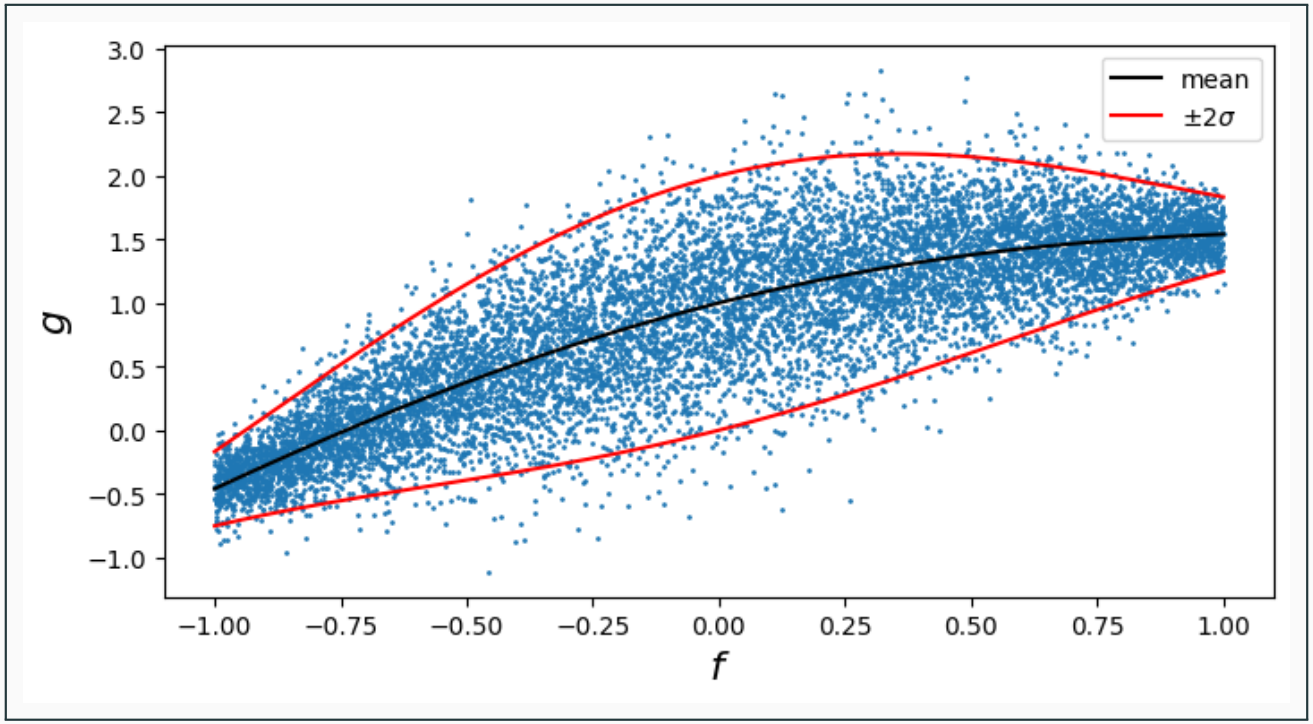
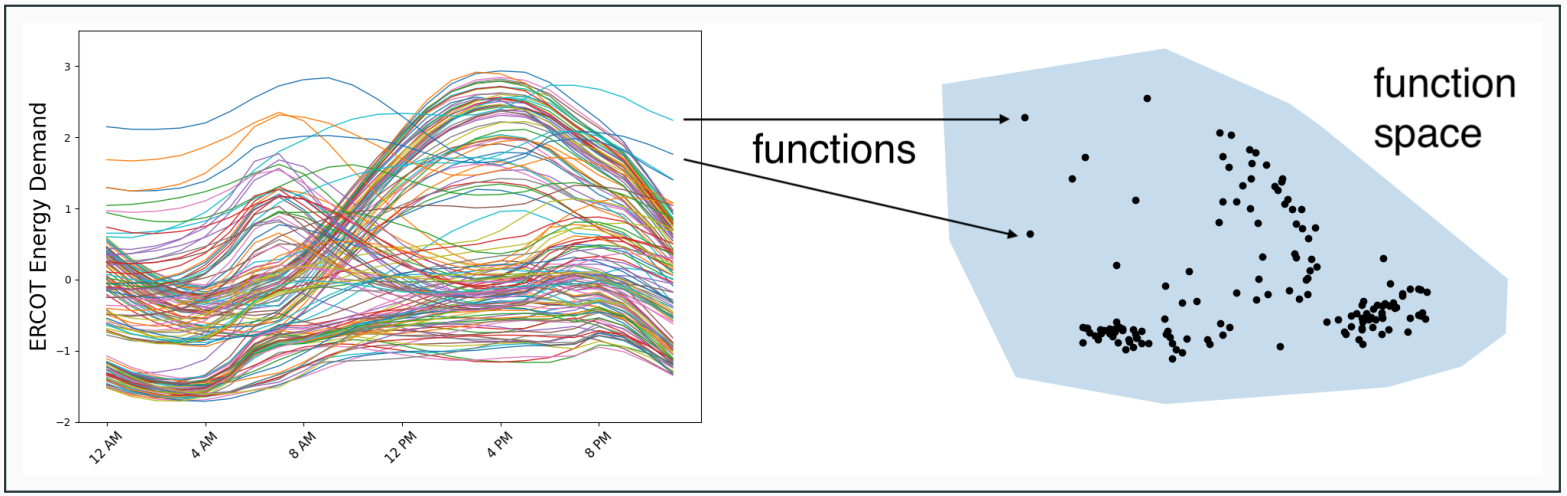


Figure 2: 1D example of local smoothness / local exchangeability

- Idea: place more weight on data near f_{n+1} when estimating $C_\alpha(f_{n+1})$
- [Guan (2023)] – strong performance of local methods (univariate).

  $\Rightarrow$ We take a local approach for *operator* problems.

Localizing Operator Models



- Operator Model: $\Gamma_\theta : \mathcal{F} \to \mathcal{G}$
  - $\mathcal{F}, \mathcal{G} \subset \mathcal{L}^2(\mathbb{R}^p)$ – square-integrable functions on $\mathbb{R}^p$ (vector space)
  - $\mathcal{D}_{\text{tr}} = \{(f_s, g_s)\}_{s=1}^m$ and $\mathcal{D}_{\text{cal}} = \{(f_t, g_t)\}_{t=1}^n$
  - New input function $f_{n+1} \in \mathcal{F}$
- Statistical model $g_t = \Gamma(f_t) + r_t$
  - $\Gamma : \mathcal{F} \to \mathcal{G}$ – unknown population operator
  - $(r_t)_{t \in \mathcal{T}}$ – error process varies smoothly as a function of f_t.
- Functions as primitives. Analysis in function space.

Local Adaptive Scoring

- Prediction set $C_\alpha(f_{n+1}) \subset \mathcal{G}$ such that

$$\mathbb{P}(g_{n+1} \in C_\alpha(f_{n+1})) \geq 1 - \alpha$$

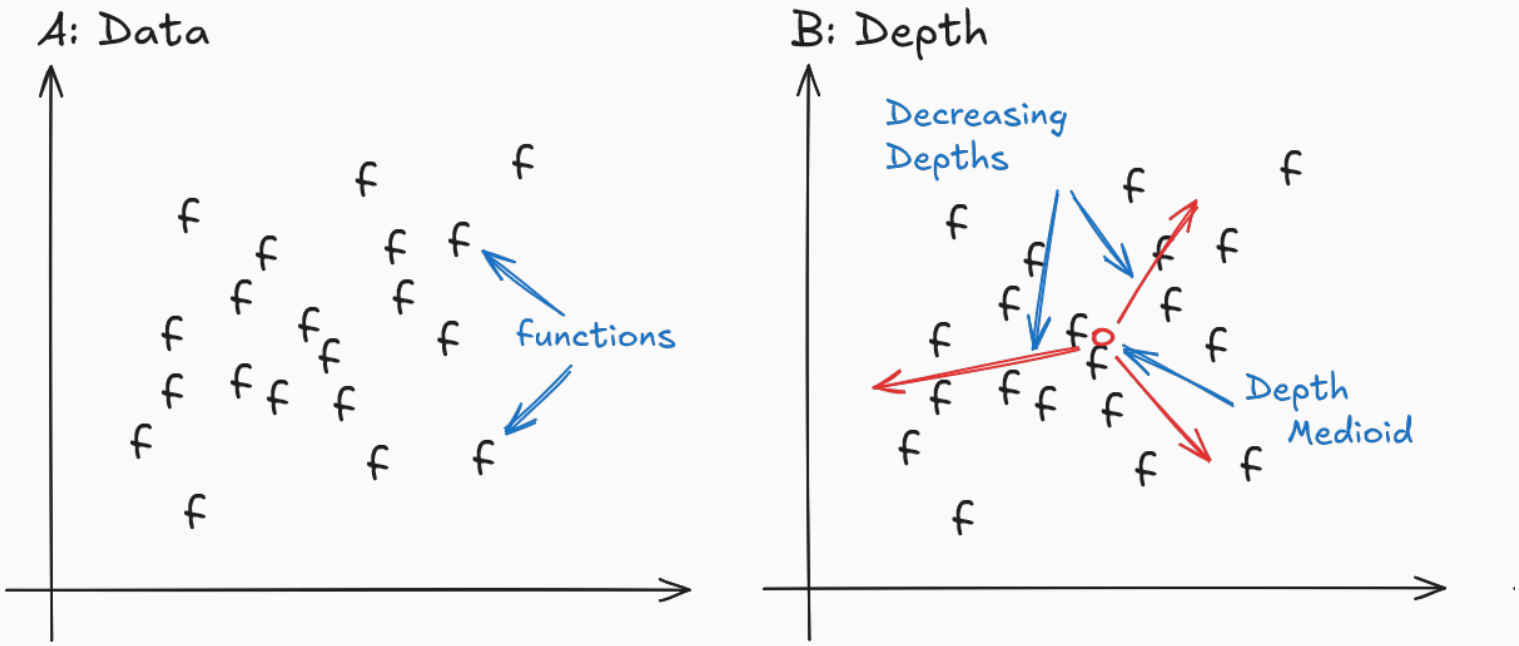
  - The set $C_\alpha(f_{n+1})$ should reflect local heterogeneity of $(r_t)_{t \in \mathcal{T}}$
  - If f_s and f_t are similar, then $C_\alpha(f_s)$ and $C_\alpha(f_t)$ are similar (shape, volume, location, etc.)
  - Closely circumscribe variability of $g_{n+1} = \Gamma_{\hat{\theta}}(f_{n+1}) + r_{n+1}$, at level $1 - \alpha$.

- Define test-specific score $S(\cdot)$ that scores residuals with respect to the *distribution* of r_{n+1}, denoted P_{n+1}.

$$S(r_1 \mid P_{n+1}), S(r_2 \mid P_{n+1}), ..., S(r_n \mid P_{n+1}) \tag{2}$$

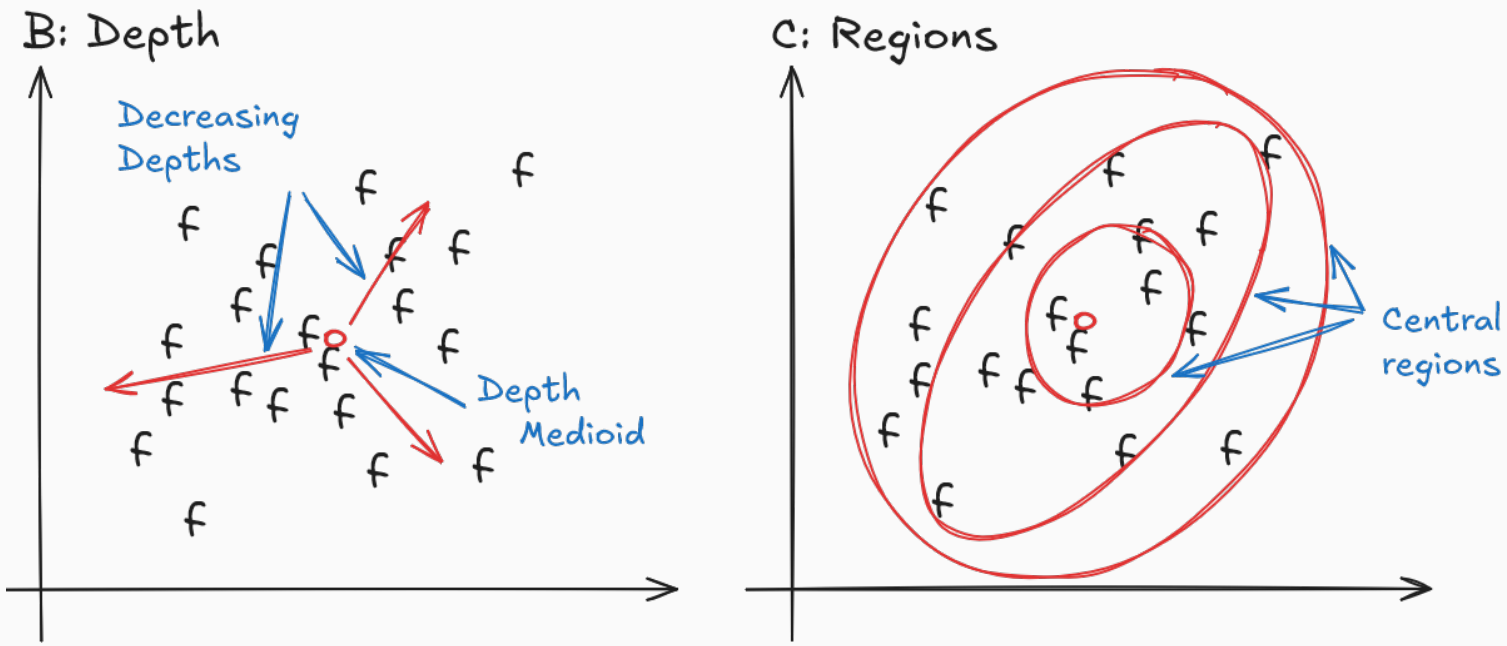
- Induce a prediction set $C_\alpha(f_{n+1})$ that broadly reflects the scale, location, shape of P_{n+1}

Measuring Outlyingness - Data Depth



- Depth function $d : \mathcal{F} \times \mathcal{P}(\mathcal{F}) \to [0, 1]$ quantifies centrality of f with respect to P ($0 =$ most outlying; $1 =$ most central).
- E.g. integrated/infimum depths [Mosler and Polyakova (2012); Mosler (2013)], norm depths [Zuo and Serfling (2000)], band depths [López-Pintado and Romo (2009)], and shape depths [Harris et al. (2021)].

Measuring Outlyingness - Central Regions



- For any $\gamma \in (0, 1)$, we define the γ-level central region of P as

$$D_\gamma(P) \;=\; \{\, f \in \mathcal{F} :\; D(f \mid P) \geq \gamma \,\}.$$

- Central regions are nested and expand monotonically as $\gamma \to 0$. Reflect the location, scale, and shape of P. $\Rightarrow$ Characterize P.

Φ-depth

- Let Φ denote a projection class $\phi : \mathcal{F} \to \mathbb{R}$, define the Φ-depth:

$$D^{\Phi}(f \mid P) = \inf_{\phi \in \Phi} D\big(\phi(f) \,\big|\, \phi(P)\big).$$

  - $\phi(P)$ is the pushforward of P through ϕ
  - D(x, F) – univariate depth function

$$D(x \mid F) = 2 \min\{\hat{F}(x), 1 - \hat{F}(x)\}$$

  with $\hat{F} \approx \phi(P)$ (empirical CDF)

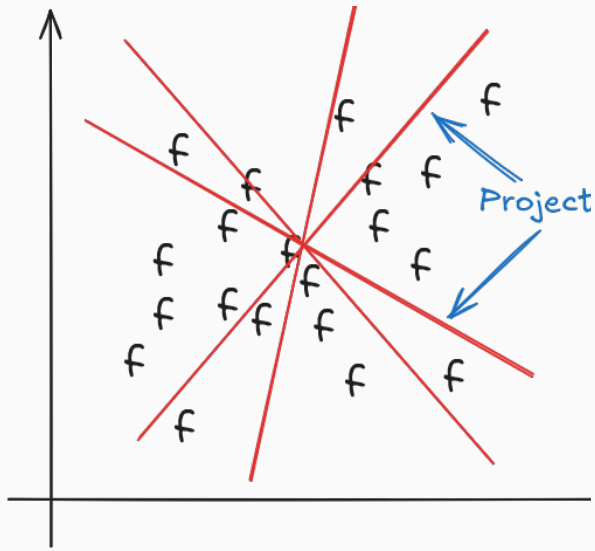


Figure 3: Slicing functional data

- Φ-depth measures worst case "outlyingness" along all projections $\phi \in \Phi$.

- Non-degenerate in function spaces, affine-equivariant, robust to outliers, and decrease continuously from the center outwards

Local Φ-Scoring

- Introduce a new scoring function – Local Φ-Score.

$$S^{\Phi}(r_t \mid P_{n+1}) = \inf_{\phi \in \Phi} D(\phi(r_t) \mid \phi(P_{n+1}))$$

  - Φ-depth of r_t with respect to P_{n+1}
  - $\phi : \mathcal{G} \mapsto \mathbb{R}$ is a projection function.
  - $D : (\mathbb{R}, P) \mapsto [0, 1]$ is a 1D "outlyingness" score relative to P_{n+1}

- Automatically localizes scores to the test distribution P_{n+1}
- Does r_t "conform" to P_{n+1}?
- Prediction sets are residual central regions shifted by the prediction
- Problem? We don't know P_{n+1}. Thats okay! Only need $\widehat{\phi}(P_{n+1})$.

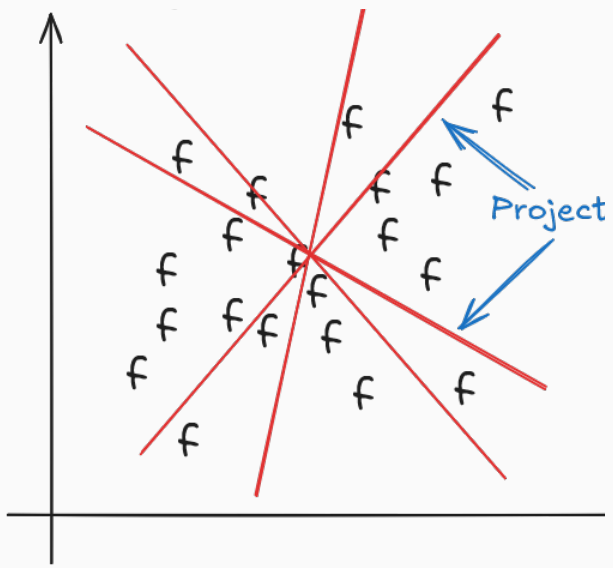


Figure 4: Slicing functional data

Estimating the Local Φ-Score

- Approximate $\hat{F} = \widehat{\phi}(P_{n+1})$ with *locally weighted CDFs* [Guan (2023)]

$$\hat{F}_{n+1} = \widehat{\phi}(P_{n+1}) = \sum_{t=1}^{n} w_t \delta(\hat{\phi}(r_t)) + w_{n+1}\delta(\infty)$$

  - Weights $w_1, w_2, .., w_{n+1}$ positive and sum to one
  - Defined by similarity kernel $H : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$

$$w_t \propto \exp(-\lambda H(f_t, f_{n+1})) \qquad (\text{e.g. } H(f_t, f_{n+1}) = \|f_t - f_{n+1}\|_2)$$

  - Computed in parallel across all $\phi \in \Phi$, reuse similarity weights.

- Use $\hat{F}_{n+1}(\cdot)$ to estimate 1D depths $\Rightarrow$ scores

$$\Rightarrow D(\phi(r_t) \mid \phi(P_{n+1})) = 2\min\{\hat{F}_{n+1}(x_t), 1 - \hat{F}_{n+1}(x_t)) \qquad x_t = \phi(r_t)$$
$$\Rightarrow S^{\Phi}(r_t \mid P_{n+1}) = \inf_{\phi \in \Phi} D(\phi(r_t) \mid \phi(P_{n+1}))$$

- Modifications: perturb f_{n+1} with noise, normalize slice scales, feature map localization $\|\varphi(f_t) - \varphi(f_{n+1})\|$

Prediction Sets – LSCI

Alg. Local Sliced Conformal Inference (LSCI)

1. Fit model on training data

$$\hat{\theta} = \arg\min \mathcal{L}(g, \Gamma_\theta(f)) \qquad D_{tr} = \{(f_s, g_s)\}_{s=1}^{m} \tag{3}$$

2. Locally score prediction residuals on held out calibration data

$$S^\Phi(r_1 \mid P_{n+1}), S^\Phi(r_2 \mid P_{n+1}), ..., S^\Phi(r_n \mid P_{n+1}) \qquad D_{cal} = \{(f_t, g_t)\}_{t=1}^{n} \tag{4}$$

3. Rank residuals by score (lower is worse)

$$\text{Find } q_{n+1} = S_{(k)}, \text{ the } k = \lfloor \alpha(n+1) \rfloor \text{ smallest score} \tag{5}$$

4. Prediction set for g_{n+1} is defined as

$$C_\alpha(f_{n+1}) = \{\Gamma_{\hat{\theta}}(f_{n+1}) + r : S(r \mid P_{n+1}) \geq q_{n+1})\} \tag{6}$$

- Repeat 2-4 for every new covariate $f_{n+1}, f_{n+2}, f_{n+3}...$

Theory – Coverage

- Localization breaks exchangeability. We can still bound the coverage gap

$$\mathbb{P}(g \in C_\alpha(f)) \geq 1 - \alpha - \sum_{t=1}^{n} w_t \, d_{\mathrm{TV}}(R, R^t) \tag{7}$$

- Bound the gap by assuming residuals are *locally exchangeable* [Campbell et al. (2019)]. Technical condition: if f_s, f_t are "close", then r_s, r_t are "more exchangeable".
- Gap is small in practice

Proposition

Let $d_t \propto H(f_t, \tilde{f}_{n+1})$ *and suppose* $w_t \propto \exp(-\lambda d_t)$, *then*

$$\sum_{t=1}^{n} w_t \, d_{\mathrm{TV}}(R, R^t) \;\leq\; \frac{\sum_{t=1}^{n} \exp(-\lambda d_t)\, d_t}{\sum_{t=1}^{n+1} \exp(-\lambda d_t)}. \tag{8}$$

Theory – Bandwidth

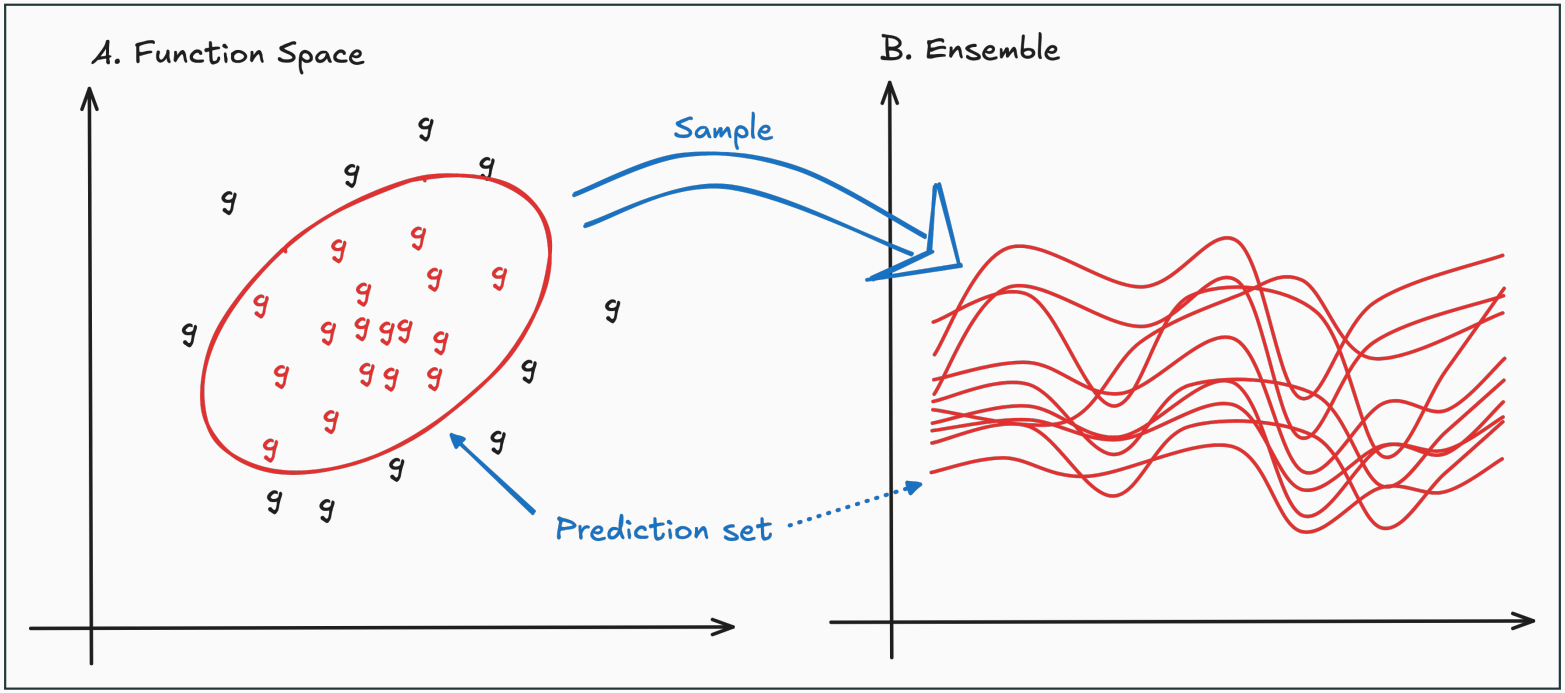
- Coverage gap controlled by the bandwidth $\lambda \geq 0$ (localization strength)

$$\sum_{t=1}^{n} w_t \, d_{\mathrm{TV}}(R, R^t) \;\leq\; \frac{\sum_{t=1}^{n} \exp(-\lambda d_t)\, d_t}{\sum_{t=1}^{n+1} \exp(-\lambda d_t)}.$$

- λ trades off localization and effective sample size.
  - $\lambda \to 0$, then weights become *uniform*. No localization, high effective sample size.
  - $\lambda \to \infty$, then weights *concentrate* at the test point. Extreme localization, low effective sample size.
  - Neither extreme minimizes bound in practice.

- Balance localization with effective sample size.
  - Tune λ to minimize bound.
  - Cross validation on calibration set

Representing Uncertainty



- How to convert a function space prediction set into something useful?
  - C_α is a "blob" in function space. Doesn't help with observations.
- Randomly sample the prediction set $\Rightarrow$ **Ensemble approximation**
  - No way(?) to directly compute the boundaries
  - Ensembles used to represent complex uncertainty (e.g. climate projections, hurricane trajectories, deep ensembles, etc.).

Local Sampler

Algorithm 1 Inverse Transform Residual Sampling

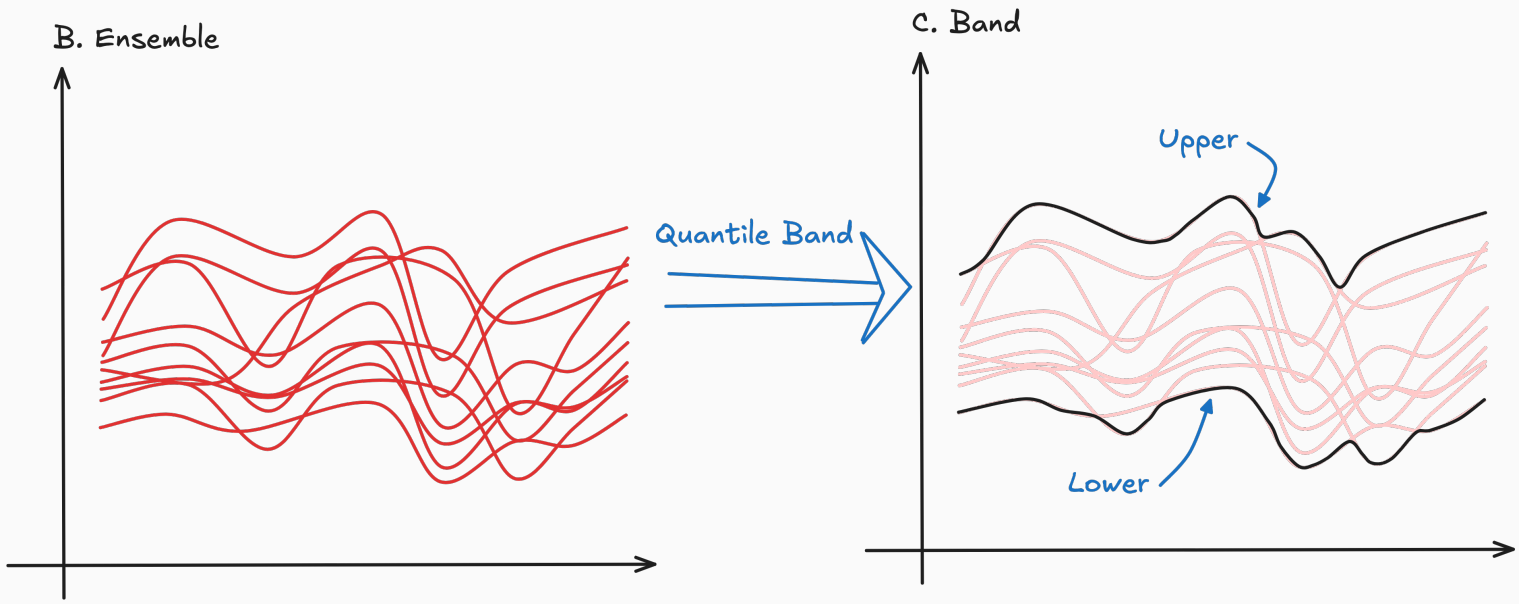
Input: $C_\alpha(f_{n+1})$, Φ, H

1. Sample $\tilde{r} \sim \Phi(G_{n+1})$
2. Accept if $\Gamma_\theta(f_{n+1}) + \tilde{r} \in C_\alpha(f_{n+1})$
3. Repeat until n_s samples accepted.

Return: $\{\tilde{r}^1_{n+1}, ..., \tilde{r}^{n_s}_{n+1}\}$.

- Sampler works with local functional principal components (FPCA)
    1. Estimate a local FPCA basis $\phi_1, ..., \phi_K$ around the test feature r_{n+1}
    2. Sample FPC scores by inverse transform from the weighted empirical pushforwards $\phi_k(P_{n+1})$
    3. Reconstruct candidate residuals $\tilde{r}$ and accept if $\Gamma_\theta(f_{n+1}) + \tilde{r} \in C_\alpha(f_{n+1})$
- Inverse-transform samples $\phi_k(r)$ independently for proposal generation
- Rejection step ensures samples lie inside the local central region.

Band representation



- Convert ensembles into bands with min and max envelopes

$$[Q_0(t), Q_1(t)]$$

- Convert ensembles into bands with pointwise quantiles

$$[Q_{\alpha/2}(t), Q_{1-\alpha/2}(t)]$$

- Pointwise quantiles can trim the band to approximate other types of coverage. E.g. probability of covering the target in observation space.

Simulations – Marginal Coverage

- Base model: Fourier Neural Operator (FNO)
  - Four FNO layers with 16 or 16x32 Fourier modes
  - Train with Adam (lr = 1e-3) for 50 epochs on NVIDIA V100.
- LSCI localization settings
  - N random projections (1, 10, 100, 200)
  - Localizers L^2, L^∞, and k-NN ($k = 0.1n$)
  - Feature maps – Identity (none), FPCA (p = 16), Latent Embedding, Fourier coefficients (p = 16)
- Data gen settings
  - $f_t \sim \mathcal{GP}(0, K)$, $g_t = f_t + \varepsilon_t$
  - Homoskedastic error process
  - $n = 1000$ train, cal., test each
  - 100 replicates

Marginal Coverage

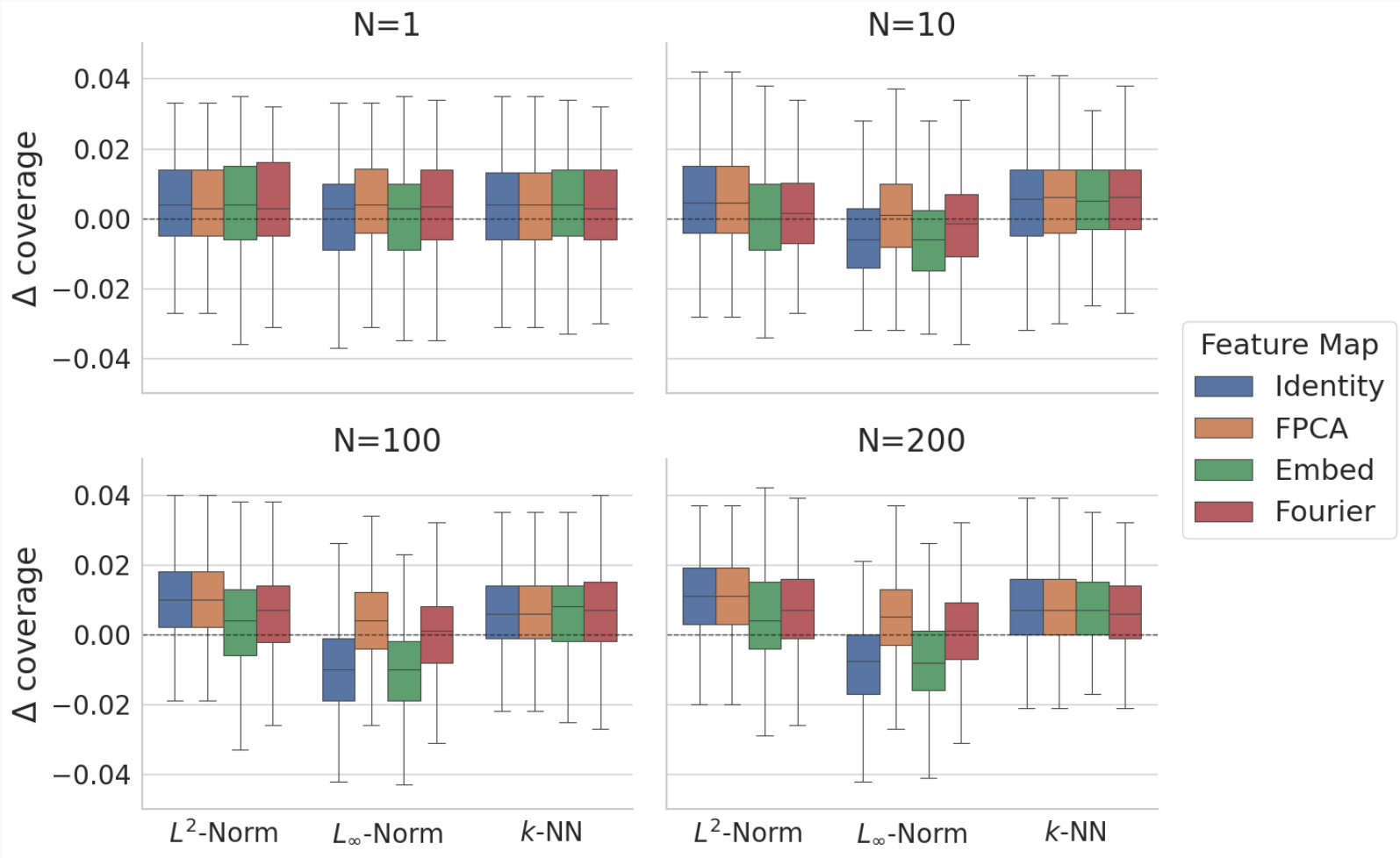


Figure 5: LSCI empirical coverage ($\alpha = 0.1$) across H–φ and projection N

Empirical coverage gap is small. Instabilities due to ties (try: softmin).

Baseline Comparisons

- Base model: Fourier Neural Operator (FNO)
  - Four FNO layers with 16 or 16x32 Fourier modes
  - Train with Adam (lr = 1e-3) for 50 epochs on NVIDIA V100.
- Baseline conformal UQ methods
  1. Conf. - Marginal conformal with FPCA [Lei et al. (2015)]
  2. Supr. - Marginal conformal with data depth [Diquigiovanni et al. (2022)]
  3. UQNO - Operator variant of CQR [Ma et al. (2024)]
  4. PONet - DeepONet variant of Adaptive Scoring [Moya et al. (2025)]
  5. QONet - DeepONet variant of CQR [Mollaali et al. (2024)]
- Four LSCI variants (different localization and sampling)
  1. LSCI1 - L_∞ loc., Fourier feat., Ens. band cover like UQNO
  2. LSCI2 - kNN loc., No feat., Ens. band cover like UQNO
  3. LSCI3 - L_∞ loc., Fourier feat., Ens. band cover like PONet, QONet
  4. LSCI4 - kNN loc., No feat., Ens. band cover like PONet, QONet

Metrics

- **Coverage**: Three notions of coverage for functional data. Different methods control different coverage notions.
  - FC – "Functional Coverage" – fraction of target functions 100% contained in the prediction set
  - CR – "Coverage Risk" – fraction of target functions (1 - γ)100% contained in the prediction set. FC with γ slippage.
  - EC – "Expected Coverage" – Average fraction of target function contained in the prediction set.

- **Precision**: Width and Interval Score measure the precision of the prediction set
  - BW – "Prediction Band Width" – Average width of the prediction set
  - IS – "Interval Score" – Proper scoring rule for interval forecasts (prediction bands). Lower is better if coverage is controlled.

Data Generation

- **Reg-GP1D** – Univariate GP regression. **Global Het.**

$$\sigma_t^f(u) \equiv \sigma_f \, g_f(t), \qquad \sigma_t^g(u) \equiv \sigma_g \, g_g(t),$$

  with $g_f(t)$ and $g_g(t)$ smooth sinusoidal ramps over time.
- **AR-GP1D** – AR(1) Univariate GP forecasting. **Spectral Het.**

$$\sigma_t^g(u) \;=\; \sigma_g \Big[1 + \sum_{k=1}^{2} a_{k,t} \, \phi_k(u) \Big],$$

  where $\{\phi_k\}$ are sinusoidal basis functions on $[0, 1]$.
- **AR-GP2D** – AR(1) bivariate GP forecasting. **Local Het.**

$$\sigma_t^f(u) \;=\; \sigma_f \Big[1 + \alpha_f \, \kappa\Big(\tfrac{u - c(t)}{w} \Big) \Big], \qquad \sigma_t^g(u) \;=\; \sigma_g \Big[1 + \alpha_g \, \kappa\Big(\tfrac{u - c(t)}{w} \Big) \Big],$$

  where κ is a smooth, nonnegative bump function (e.g., Gaussian).
- Train, calibrate, and test on 1000 samples per split. $p = 128$ or $32x64$. 25 simulation replicates for 1D, 5 for 2D.

Results

Table 1: Coverage and interval metrics on GPsimulations. Coverage (either FC, EC, or CR) should be high (up to 0.9), while interval score (IS) should be low.

	Reg-GP1D – Global Het.				**AR-GP1D – Spectral Het.**				**AR-GP2D – Local Het.**			
Method	FC ↑	EC ↑	CR ↑	IS ↓	FC ↑	EC ↑	CR ↑	IS ↓	FC ↑	EC ↑	CR ↑	IS ↓
Baselines												
Conf.	0.900	0.999	0.999	3.779	0.888	0.998	0.996	2.380	0.914	0.942	0.976	1.900
Supr.	0.902	0.993	0.980	2.706	0.891	0.995	0.991	2.152	0.890	1.000	1.000	3.020
UQNO	0.776	0.973	0.903	1.691	0.561	0.969	0.892	1.512	0.000	0.940	0.912	1.734
PONet	0.527	0.901	0.683	1.363	0.206	0.897	0.587	1.496	0.000	0.901	0.542	1.839
QONet	0.516	0.917	0.689	1.360	0.134	0.898	0.567	1.467	0.000	0.906	0.582	1.852
Proposed												
LSCI1	0.909	0.975	0.901	1.935	0.904	0.966	0.885	1.430	0.972	0.979	0.976	0.892
LSCI2	0.912	0.973	0.893	1.609	0.906	0.976	0.933	1.442	0.916	0.996	0.998	1.444
LSCI3	0.909	0.904	0.655	1.200	0.904	0.899	0.586	0.997	0.972	0.948	0.862	0.786
LSCI4	0.912	0.900	0.629	1.026	0.906	0.909	0.605	0.984	0.916	0.983	0.976	1.160

How big of an ensemble?

- AR-GP1D settings
  - 1000 Train, 1000 Calibration and 1000 Test functions
  - $p = 128$ sample points.
- Increase the ensemble size n_s from 50 to 5000.
- IS (skill) doesn't change if EC (coverage) doesnt.
- Compute time and space requirements increase linearly.

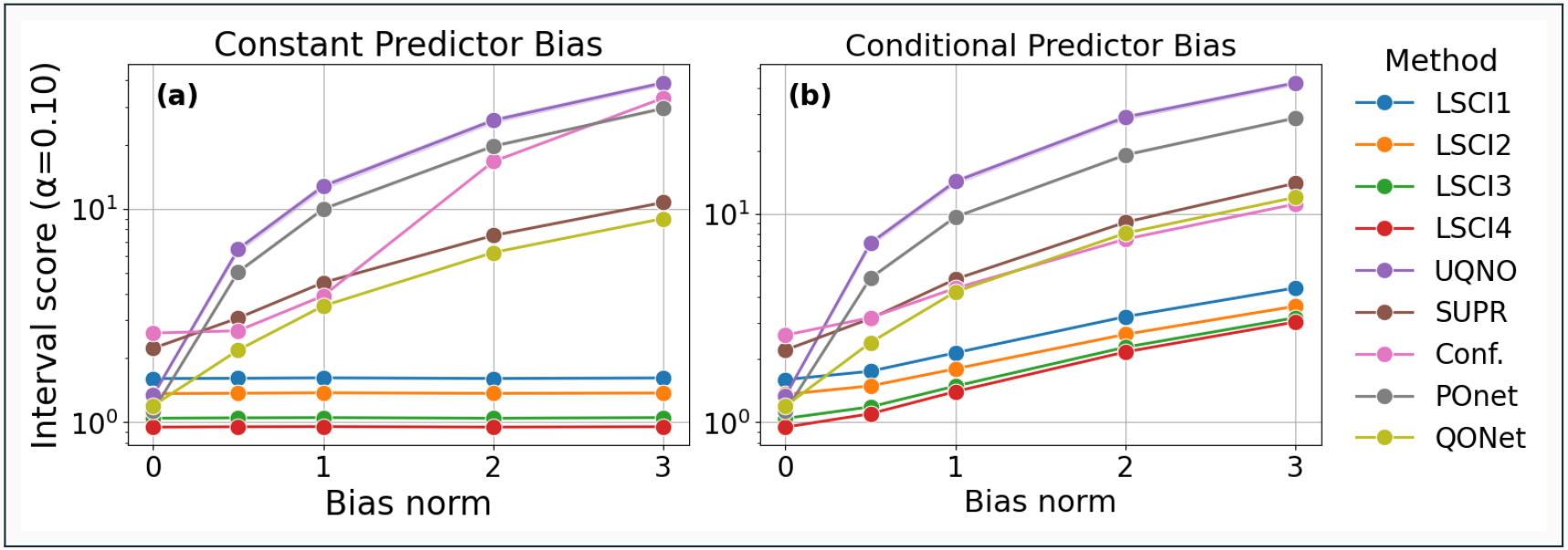
Table 2: IS doesn't vary with increasing sample sizes n_s as long as EC is controlled.

$n_s =$	50	500	1000	2000	5000
EC	0.922	0.932	0.933	0.932	0.929
IS	1.024	1.038	1.042	1.040	1.024

Table 3: Compute time and space scale linearly with n_s (ensemble size)

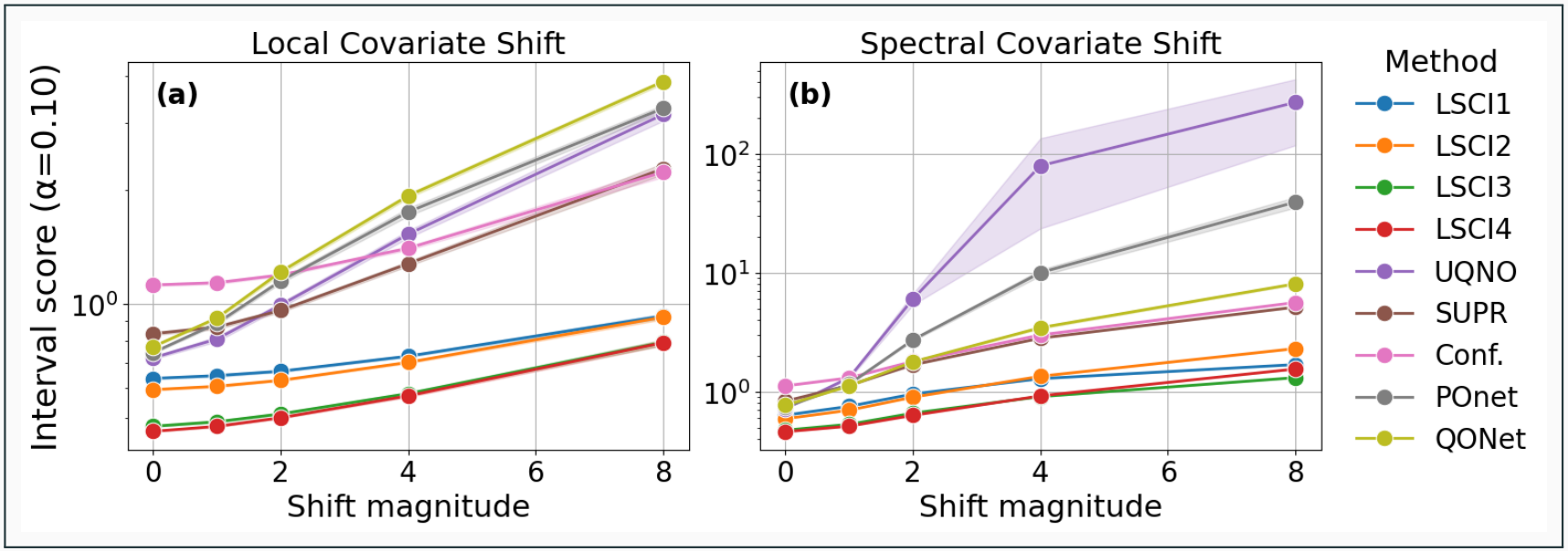
$n_s =$	50	500	1000	2000	5000
Time (s)	0.106	0.143	0.178	0.266,	0.513
Space (mb)	25.6	256.0	512.0	1024.0	2560.0

Robustness – Biased Predictors



- **(a)** – constant bias function added to each prediction
- **(b)** – Bias function that depends on f added to each prediction
- Immune to constant predictor bias (unrealistic), decays much slower than baselines with conditional predictor bias (realistic)

Robustness – Distribution Drift



- **(a)** – local σ "bump" function changes location and magnitude from train to calibration to test,
- **(b)** – Fourier coefficients of global σ function "rotate" from train to calibration to test.
- Decays much slower than baselines under local and spectral out-of-distribution variance changes.

Real Data Baselines

1. **Energy Demand Forecasting:** Daily 24-hour energy demand curves from the Electric Reliability Council of Texas (ERCOT)
   - Daily profiles from hourly measurements (eia.gov/electricity)
   - $n_{tr} = 1500$, $n_{cal} = 2500$, $p = 24$. One step ahead forecast.

2. **Air Quality Estimation:** Daily PM2.5 profiles from Beijing, China
   - Daily profiles from hourly measurements (UCI dataset 501).
   - Estimate profiles from temperature, pressure, dew point profiles.
   - $n_{tr} = 600$, $n_{cal} = 1200$, $p = 24$.

3. **Weather-ERA5 Forecasting**: Daily global 2m surface temperature fields from ERA5 reanalysis data.
   - 32×64 latitude–longitude grid, aggregated daily.
   - $n_{tr} = 3650$ (1959-1969), $n_{cal} = 1825$ (1969-1973), $p = (32, 64)$.
   - One step ahead forecast. Test (1973-1978).

Results

Table 4: Uncertainty metrics for all conformal methods applied to energy forecasting, weather forecasting, and air quality prediction.

	Energy Demand				Air Quality				Weather-ERA5			
Method	FC ↑	EC ↑	BW ↓	IS ↓	FC ↑	EC ↑	BW ↓	IS ↓	FC ↑	EC ↑	BW ↓	IS ↓
Baselines												
Conf.	0.582	0.981	2.135	2.217	0.883	0.989	1.845	2.851	0.950	0.876	6.681	8.327
Supr.	0.633	0.939	1.396	1.646	0.000	0.879	0.479	2.096	0.876	1.000	18.08	18.09
UQNO	0.513	0.913	1.353	1.690	0.000	0.161	0.091	3.851	0.000	0.916	4.572	5.654
PONet	0.496	0.841	0.895	1.466	0.565	0.894	203.7	232.5	0.000	0.889	15.63	21.23
QONet	0.482	0.802	1.016	1.759	0.000	0.296	15.68	217.3	0.000	0.890	13.293	16.65
Proposed												
LSCI1	0.892	0.935	1.518	1.546	0.887	0.676	0.243	0.433	0.919	0.990	5.362	5.418
LSCI2	0.909	0.934	1.513	1.540	0.937	0.967	0.731	0.839	0.957	0.994	5.608	5.631
LSCI3	0.892	0.897	1.227	1.257	0.887	0.659	0.229	0.424	0.919	0.985	4.836	4.916
LSCI4	0.909	0.897	1.216	1.257	0.937	0.917	0.479	0.599	0.957	0.991	5.152	5.187

Spatial adaptivity

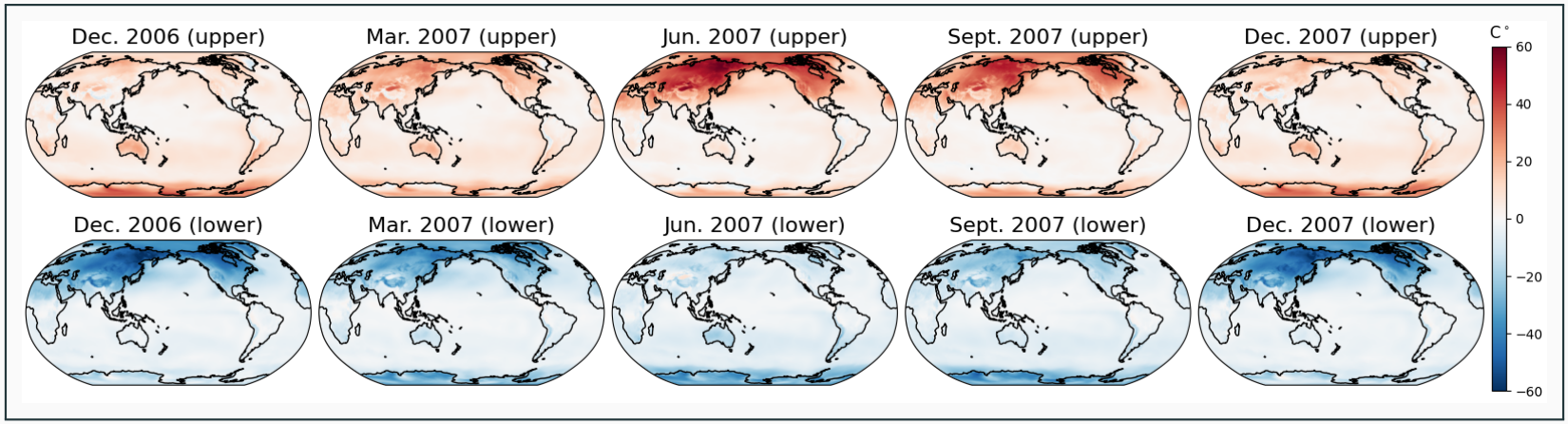


Figure 6: LSCI bands capture seasonal variations not captured by the FNO

- Sampled lower and upper envelope bands show spatially-varying seasonal variation in Weather-ERA5 forecasting
- LSCI prediction sets capture variation the underlying prediction model (FNO) missed during training

Discussion

- Motivation
  - Operator models are an important development in ML / physical modeling
  - Neural operators lack inherent UQ
  - UQ is critical to many applications
- Method
  - Local Sliced Conformal inference
  - Φ-Scores characterize local distribution
  - Define local prediction sets in function space
  - Sample to get approximate prediction sets in obs. space
- Results
  - Good coverage, low interval scores.
  - More adaptive (size and shape) than baselines
  - Robust to mild out-of-distribution behavior
- Future work
  - Improve computational efficiency! 2D is expensive. Avoid sampling?