DeepKriging on the Sphere: Baseline vs. Spherical Variants

Yue Yu†

†: Department of Statistics, Indiana University

Motivation & Goals

- **Goal:** Apply `DeepKriging` to processes on the sphere S^2 and compare:
    - Baseline *Euclidean* `DeepKriging` (lat–lon treated as flat 2D).
    - Option A: **Spherical** `DeepKriging` with *great-circle RBFs*.
    - Option B: **Spherical CNN** (Chebyshev graph convolutions).
- Why: Euclidean distances distort neighborhoods on the globe (poles/dateline).

Baseline Euclidean `DeepKriging` — Overview

Idea: Treat (lat, lon) as flat 2D coordinates; build Euclidean RBF features and learn a nonlinear predictor with an MLP (`DeepKriging`).
Pipeline:

1. Sample N sites uniformly on S^2; simulate $y = f + \varepsilon$ via truncated spherical harmonics with Matérn-like spectrum.
2. Normalize $(\mathrm{lat}, \mathrm{lon}) \mapsto [-1, 1]^2$; place *multi-resolution* grid centers in lat–lon.
3. Compute Euclidean distances to centers; map to compactly supported **Wendland** C^4 features:

$$\phi(r) = \begin{cases} (1-r)^6 \, \dfrac{35r^2 + 18r + 3}{3}, & 0 \le r \le 1, \\ 0, & r > 1, \end{cases} \qquad r = \frac{\|x - c\|_2}{\theta}.$$

4. Concatenate $[\,\mathrm{intercept} \mid \phi\,]$ and fit an MLP head 50-50-50 $\to$ 1 with MSE.

Baseline Euclidean `DeepKriging` — Feature Engineering

Centers (multi-res): Cartesian product grids per level (e.g., 9×18, 18×36).
Range selection (auto-θ).

- Per level ℓ: compute median nearest-neighbor distance among centers; set $\theta_\ell = \kappa \cdot \mathrm{medianNN}_\ell$.
- *Coverage guard:* If any site activates $< m_{\mathrm{min}}$ bases, inflate all $\theta_\ell \leftarrow \gamma\, \theta_\ell$ iteratively.

Known limitations: Distances distort near poles; centers non-uniform in surface area measure.

Option A — Spherical DK with Great-Circle RBFs: Concept

Goal: Respect spherical geometry by replacing flat distances with *great-circle* (haversine) distances while keeping a compactly supported basis.

Centers on S^2: Use Fibonacci (low-anisotropy) nodes per level; optionally add coarse levels to ensure global coverage.
Distances: For site x and center c with spherical angle $\alpha(x, c)$:

$$d_{\mathrm{gc}}(x, c) = R \cdot \alpha(x, c), \qquad \alpha = 2 \arcsin \sqrt{\sin^2 \frac{\Delta\varphi}{2} + \cos\varphi_x \cos\varphi_c \, \sin^2 \frac{\Delta\lambda}{2}}.$$

Kernel: Same Wendland C^4 with $r = d_{\mathrm{gc}}/\theta$.

Option A — GC-RBFs: Construction & Tuning

Per-level range (auto-θ):

- For level ℓ with centers $\{c\}$, compute center–center geodesic distances and pick $\theta_\ell = \kappa \cdot \text{medianNN}_\ell$.
- *Coverage guard:* ensure each sample activates at least m_{min} bases by multiplicative inflation.

Design matrix: $\Phi_{ij} = \phi\big(d_{\mathrm{gc}}(x_i, c_j)/\theta_{\ell(j)}\big)$; concatenate intercept/covariates if any.
Head & loss. Same 50-50-50 $\to$ 1 MLP, MSE objective.
Advantages:

- Geometry-aware neighborhoods; compact support gives sparsity and local adaptivity.
- Maintains interpretability of range/levels similar to classical kriging practice.

Option A — Practical Workflow

Workflow.

1. Pick levels (e.g., 64, 256 centers) via Fibonacci nodes; precompute pairwise center geodesic distances.
2. Auto-θ: $\theta_\ell \leftarrow \kappa \cdot \mathrm{medianNN}_\ell$; enforce coverage.
3. Build Φ on train/val/test; train MLP head (Adam, early stopping).
4. Inspect coverage stats, residual maps, and sensitivity to κ, m_{min}, and level sets.

Option B — Spherical CNN (Chebyshev) on a kNN Sphere Graph

Graph: Build geodesic k-NN graph: adjacency W from great-circle distances; symmetric normalization.
Laplacian: $L = I - D^{-1/2} W D^{-1/2}$; estimate $\lambda_{\max}(L)$ (power iteration); rescale

$$\tilde{L} = \frac{2}{\lambda_{\max}} L - I \ \in [-1, 1]$$

for Chebyshev stability.
Chebyshev conv (order K).

$$T_0 X = X, \quad T_1 X = \tilde{L} X, \quad T_k X = 2\tilde{L}(T_{k-1} X) - T_{k-2} X.$$

Feature map: $Z = \sum_{k=0}^{K-1} T_k X \, \Theta_k$.

Option B — Inputs, Head, and Stabilization

Inputs X: Positional encodings $[1, \sin\varphi, \cos\varphi, \sin\lambda, \cos\lambda]$ and (optionally) low-order harmonics to inject global structure.

Backbone: Two ChebConv layers (orders K), each with LayerNorm + LeakyReLU.

Head: Concatenate skip ($[Z \mid X]$) $\rightarrow$ MLP 50-50-50 $\rightarrow$ 1; optionally add linear skip XW to output. Train on z-scored y.

Why this helps:

- Rescaling by λ_{max} prevents spectral shrinkage/explosion.
- Skip connections and LayerNorm mitigate dead filters; inputs carry large-scale modes.

Option B — Training Recipe & Diagnostics

Recipe:

1. Choose $k \in [12, 20]$, $K \in [3, 6]$, channels per layer $\in [16, 64]$.
2. Build graph/Laplacian, compute $\lambda_{\max}$, construct $\tilde{L}$.
3. Full-batch training (batch$= N$) for graph ops; Adam with EarlyStopping & ReduceLROnPlateau.

Diagnostics:

- Conv1 feature std (channels) $\Rightarrow$ check for collapse.
- Ablate k, K, channels; visualize residuals vs. latitude/longitude.

Data & Simulation Setup

- Sample N locations uniformly on $S^2 \Rightarrow (\text{lat}, \text{lon})$.
- Simulate a smooth isotropic field using a truncated spherical-harmonic expansion:

$$f(\theta, \phi) = \sum_{\ell=0}^{L_{\max}} \sum_{m=-\ell}^{\ell} w_{\ell m} Y_{\ell m}(\theta, \phi),$$

$$\mathbb{E}[w_{\ell m}^2] = C_\ell, \quad C_\ell = \sigma_f^2 \big(1 + \alpha\, \ell(\ell+1)\big)^{-\nu}.$$

- Add noise: $y = f + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma_n^2)$.
- **Split**: train/val/test via `train_test_split`;

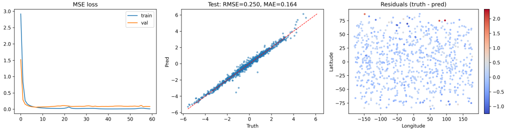
  Evaluation Metrics: RMSE/MAE (same as used in Chen et al. (2020)).

Quantitative Snapshot (current runs)

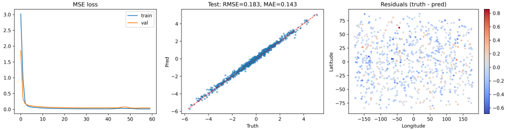
Model	RMSE (test)	MAE (test)
EuDK-RBF (Baseline)	≈ 0.250	≈ 0.164
SphDK-GC-RBF (Option A)	≈ 0.183	≈ 0.143
SphDK-SCNN (Option B)	≈ 1.497	≈ 1.165

- EuDK baseline performs well despite geometry mismatch;
- Option A and B are *work-in-progress*: Option A expected to improve with auto-θ tuning and coarser levels;
- Option B improved with Laplacian rescaling, richer inputs, and normalization, but still under baseline.

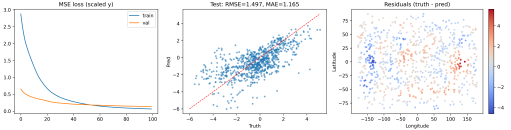
Representative Plots



Baseline: Euclidean `DeepKriging`



Option A: (Spherical DK with Great-Circle RBFs)



Option B: (Spherical CNN—Chebyshev Graph Convolutions)

Takeaways & Next Steps

Takeaways

- Baseline EuDK is a strong yardstick; local structure dominates in this simulation.
- Geometry-aware methods need careful *feature coverage* (A) and *spectral scaling* (B).

Next Steps

- Option A: add coarse level(s) (e.g., $1, 16, 64, 256$), increase θ multiplier, target min-activation ≥ 12.
- Option B: larger k, K, channels; include harmonics up to $L_{\text{feat}} = 6$–8; mutual-kNN; mild dropout/L2.
- Benchmark against real geospatial datasets; ablate hyperparameters and report CIs.

Questions?

Thank you!

References I

W. Chen, Y. Li, B. J. Reich, and Y. Sun. Deepkriging: Spatially dependent deep neural networks for spatial prediction. *arXiv preprint arXiv:2007.11972*, 2020.