ICML 2026 · SEOUL, SOUTH KOREA

gp2Scale: Exact Gaussian Processes on 10 Million Data Points via Compactly Supported Non-Stationary Kernels & Distributed Computing

Marcus M. Noack[1], Mark D. Risser[2], Hengrui Luo[3], Vardaan Tekriwal[1,4], Ronald J. Pandolfi[1]

[1]Applied Mathematics & Computational Research Division, Lawrence Berkeley National Laboratory [2]Climate & Ecosystem Sciences Division, LBNL

[3]Department of Statistics, Rice University [4]University of California, Berkeley

Lawrence Berkeley National Laboratory

U.S. Department of Energy

gpCAM

gpcam.lbl.gov

R&D 100 Award 2024

Motivation & Core Insight

Gaussian process regression is the gold standard for probabilistic function approximation — but standard GPs are limited to ∼10,000 points due to $\mathcal{O}(N^3)$ time and $\mathcal{O}(N^2)$ memory complexity.

The core claim: GPs are not *inherently* dense. Standard kernel designs *impose* density. With the right kernel, the covariance matrix reveals natural sparsity.

The dominant paradigm sacrifices accuracy or customizability:

INDUCING POINTS
SVGP, SGPR — approximate likelihood; restrict kernel choices

LOCAL APPROXIMATION
NNGP, Vecchia — approximate posterior; sparse precision matrix Q^{-1}

GRID INTERPOLATION
SKI/KISS-GP — restricted to low-dimensional Euclidean inputs

SPARSE UQ
All approximate methods risk overconfident or poorly calibrated posteriors

gp2Scale achieves scalability *without* these approximations by discovering sparsity *through kernel design*, not imposing it as an algorithmic constraint.

Three Pillars of gp2Scale

1 Flexible Non-Stationary Compact Kernels
A new class of kernels that return *exactly zero* for independent pairs — allowing natural sparsity to emerge from the data structure, not be imposed by approximation.

2 Distributed HPC Computing (DASK)
Covariance matrix computed block-wise across GPU workers. Blocks transmitted in sparse COO format, assembled in CSR for fast linear solves. Trivially parallelizable.

3 Block Metropolis–Hastings MCMC
Hyperparameters sampled in independent blocks for fast convergence. MCMC provides full posterior uncertainty over hyperparameters — no local optima problem.

Compute Pipeline & Training

Split Data → DASK Workers → COO Blocks → CSR Assembly → Sparse Solve

Each DASK worker computes a dense covariance block and returns it in sparse COO format, minimizing communication load. The host assembles a global sparse CSR matrix and runs sparse linear algebra.

Training objective (log marginal likelihood)

$$\log p(y \mid \varphi) \propto -\tfrac{1}{2} y^\top \big(K(\varphi)+V(\varphi)\big)^{-1} y \;-\; \tfrac{1}{2}\log\big|K(\varphi)+V(\varphi)\big|$$

Both the linear solve $(K+V)^{-1}y$ and the log-determinant $\log|K+V|$ are computed via sparse linear algebra. Hyperparameters φ are sampled via block-MH MCMC, propagating full posterior uncertainty and avoiding local optima in the non-convex landscape.

Kernel Family

Wendland-style (Stationary Baseline) Eq. 3

Compactly supported stationary kernel with radius r_0, where $d_{ij} = \|x_i - x_j\|$. Returns *exactly zero* when $d_{ij} \geq r_0$. $p(t) = 32t^3 + 25t^2 + 8t + 1$ is the degree-3 polynomial factor ensuring positive definiteness and C^6 smoothness *(Wendland, 1995)*.

$$k_W(x_i, x_j;\, r_0) = \left(1 - \frac{d_{ij}}{r_0}\right)^8 p\left(\frac{d_{ij}}{r_0}\right), \quad d_{ij} < r_0$$

Non-Stationary via Convolution Eq. 6

Paciorek–Schervish convolution with spatially varying signal variance $\sigma_s(x)$ and anisotropic length scale matrix $\Sigma(x)$. Positive semi-definiteness guaranteed. Ideal for smooth, strongly non-stationary functions (topography, climate).

$$k(x_i, x_j) = \frac{\sigma_s(x_i)\,\sigma_s(x_j)\,|\Sigma(x_i)|^{1/4}|\Sigma(x_j)|^{1/4}}{\left|\frac{\Sigma(x_i)+\Sigma(x_j)}{2}\right|^{1/2}}\, k_W\Big(\sqrt{Q(x_i, x_j)}\Big)$$

Bump-Function Kernel (Far-Field) Eq. 9

Far-field covariances modeled via bump functions $b_u(x, x_p)$ acting as learnable masks. Each $g_u(x) = \sum_p b_u(x, x_p)$ aggregates bumps at locations x_p; the product $g_u(x_i)g_u(x_j)$ activates covariances between spatially separated subsets. Far-field rank controlled by U.

$$k(x_i, x_j) = k_{\text{core}}\Big[\sum_u^U g_u(x_i)\, g_u(x_j) + k_W(x_i, x_j)\Big]$$

Collapsed Bumps → Delta Mask Eq. 10

For discrete or high-dimensional inputs (e.g., MNIST images), bump radii collapse to Dirac deltas $\delta(x, x_q)$. Maximally sparse, differentiability-free mask. Combined with k_W for diagonal stability.

$$k_d(x_i, x_j) = \sum_p g_p(x_i)\, g_p(x_j), \qquad g_p(x) = \sum_q \delta(x, x_q)$$

Results

gp2Scale compared against four state-of-the-art scalable GP methods across five datasets spanning diverse input spaces:

SVGP (inducing pts) VNNGP (hybrid) SKI (grid interp.) Vecchia (local approx.) gp2Scale (exact)

1D Synthetic

Highly non-stationary, 2,000 pts

Method	RMSE↓	CRPS↓
SVGP	0.190	0.110
VNNGP	0.200	0.210
SKI	0.190	0.110
Vecchia	0.200	0.110
Base GP	0.109	0.070
gp2Scale (Eq. 6)	**0.107**	**0.060**

Results Cont.

US Topography

20K pts, 2D, high non-stationarity

Method	RMSE↓	CRPS↓
SVGP	266.5	148.0
VNNGP	236.0	175.5
SKI	206.3	117.3
Vecchia	150.0	78.4
gp2Scale (Eq. 6)	**136.3**	**63.8**

MNIST

60K pts, 28×28 images, Brier score ↓

Method	Brier↓	Notes
SVGP	0.033	500 inducing pts
VNNGP	0.052	reverts to mean
SKI	N/A	dim. too high
gp2Scale ($k_W k_d$)	**0.018**	δ-mask, ℓ_1 norm

CA Housing

20K pts, 8D, high-dimensional

Method	RMSE↓	CRPS↓
SVGP	0.600	0.350
VNNGP	0.660	0.410
SKI	0.710	0.500
Vecchia	0.600	0.310
gp2Scale (Eq. 3)	**0.490**	**0.270**

US Temperatures

10 Million pts, 3D, Perlmutter HPC

Method	RMSE↓	Hardware
SVGP	5.90	1 node
VNNGP	5.21	1 node
SKI	OOM	–
Vecchia	2.860	1 node, 1
gp2Scale (Eq. 3, training unfinished)	**2.851**	1024 A100 GPUs

Practitioner Guide

✓ Use gp2Scale when
- Dense sampling, strong non-stationarity
- Calibrated UQ is non-negotiable
- Non-Euclidean or abstract input spaces
- Custom noise / mean functions needed
- HPC resources available

× Use approximations
- Stationary, smooth functions
- High-dimensional sparse data
- Low-latency inference required
- Limited compute budget
- Vecchia for spatial w/o non-stationarity

Bottom line: gp2Scale is not a blanket solution — it is a high-fidelity instrument for settings where exactness and customizability are scientifically required, analogous to the role of high-fidelity simulation in computational science.

Access, Reproducibility & Acknowledgments

Software
Open-source; part of gpCAM
`pip install gpcam`

Reproducibility
All scripts & datasets public
`github.com/MarcusMNoack/gp2Scale`

ACKNOWLEDGMENTS

Supported by the **Center for Advanced Mathematics for Energy Research Applications (CAMERA)** (DE-AC02-05CH11231). Supported by the **U.S. Department of Energy, Office of Science, ASCR's Applied Mathematics Competitive Portfolios program** (DE-AC02-05CH11231). Supported in part by **DOE ASCR's Applied Mathematics program** under Contract No. DE-AC02-05CH11231 at LBNL, and by the **U.S. National Science Foundation** award NSF-DMS 2412403 at Rice University. Supported by the **DOE Office of Biological and Environmental Research** under the Regional and Global Model Analysis program and the **CASCADE Scientific Focus Area**, (DE-AC02-05CH11231). Research used resources of the **National Energy Research Scientific Computing Center (NERSC)**, a DOE Office of Science User Facility at LBNL, operated under Contract No. DE-AC02-05CH11231, using NERSC awards ERCAP0031656, ERCAP0032615, and ERCAP0032229.

Key Advantages

- ✓ **Exact GP:** no inducing points, no approximations — posterior exact within the kernel's RKHS; linear solves and log-det computed via sparse linear algebra.
- ✓ **Calibrated UQ:** sparsity discovered, not imposed. Posterior variance $\sigma^2(x^*)$ reflects true epistemic uncertainty
- ✓ **Input-space agnostic:** works on Euclidean, discrete, and abstract spaces (demonstrated on image pixels with ℓ_1 metric)
- ✓ **Custom noise & mean:** heteroscedastic noise models and arbitrary mean functions fully supported
- ✓ **MCMC training:** full hyperparameter posterior — avoids local optima in non-convex log-likelihoods
- ! **Trade-off:** requires substantial HPC resources for >1M points; best when exact UQ is scientifically necessary

When to use gp2Scale: Densely sampled, non-stationary functions where calibrated uncertainty quantification is critical — autonomous experimentation, materials discovery, geoscience, climate modeling.

Scaling: 10M Points on Perlmutter

10M	1024	477 s	∼100
training pts	A100 GPUs	per MCMC iter.	iters completed

$\mathcal{O}(N^2)$ covariance computation dominates; linear solve, and log-det are cheap once sparsity is identified. Full run (1000 iterations)∼1 week — comparable to large neural network training. gp2Scale is positioned as a high-fidelity tool for settings where exact UQ is scientifically non-negotiable, not a blanket replacement for efficient approximate methods.