Scalable Gaussian Process Hyperparameter Optimization via Coverage Regularization

Killian Wood [1] Alec M. Dunton [2] Amanda Muyskens [2] Benjamin W. Priest [2]

[1]University of Colorado Boulder [2]Lawrence Livermore National Laboratory

Abstract

Gaussian processes (GPs) are Bayesian non-parametric models popular in a variety of applications due to their accuracy and native uncertainty quantification (UQ). Tuning GP hyperparameters is critical to ensure the validity of prediction accuracy and uncertainty; uniquely estimating multiple hyperparameters in, e.g. the Matérn kernel can also be a significant challenge. Moreover, training GPs on large-scale datasets is a highly active area of research: traditional maximum likelihood hyperparameter training requires quadratic memory to form the covariance matrix and has cubic training complexity. To address the scalable hyperparameter tuning problem, we present a novel algorithm which estimates the smoothness and length-scale parameters in the Matèrn kernel in order to improve robustness of the resulting prediction uncertainties. Using novel loss functions similar to those in conformal prediction algorithms in the computational framework provided by the hyperparameter estimation algorithm MuyGPs, we achieve improved UQ over leave-one-out likelihood maximization while maintaining a high degree of scalability as demonstrated in numerical experiments.

Gaussian Processes

- Let $k_\theta(\boldsymbol{x}, \boldsymbol{x'})$ be the *kernel* function, which generates the covariance between $\boldsymbol{x}$ and $\boldsymbol{x'}$ and is controlled by hyperparameters θ [1]. Y is a Gaussian process if for every finite sample of Y, $Y(\boldsymbol{X}) \sim \mathcal{N}(\boldsymbol{m}(\boldsymbol{X}), \boldsymbol{K}_\theta(\boldsymbol{X}, \boldsymbol{X}))$ with $\boldsymbol{m}(\boldsymbol{X})$ the mean and $\boldsymbol{K}_\theta(\boldsymbol{X}, \boldsymbol{X})$ the covariance.
- Let $\Gamma(\cdot)$ be the gamma function and K_ν the modified Bessel function of the second kind. We assume $\boldsymbol{K}_\theta(\boldsymbol{X}, \boldsymbol{X})$ is induced by the Matérn kernel with $\theta = [\gamma^2, \rho, \nu, \tau]$ and $\|\boldsymbol{x} - \boldsymbol{x'}\|_2 = d$,

$$k_\theta(\boldsymbol{x}, \boldsymbol{x'}) = \gamma^2 \left[\frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{d}{\rho} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{d}{\rho} \right) + \tau^2 \mathbb{I}\left(d = 0\right) \right]. \tag{1}$$

MuyGPs

- Conventional GP training is prohibitively expensive at large-scale ($\mathcal{O}(n^3)$ for n training points).
- MuyGPs is an approximate hyperparameter training algorithm that achieves efficiency by: (1) Utilizing loss functions based on leave-one-out cross-validation (LOOCV), (2) Localizing kernel matrices and therefore predictions to nearest neighbor training data, and (3) batching.
- MuyGPs conditions a training feature vector $\mathbf{x}_i$ only on its k nearest neighbors $\boldsymbol{X}_{N_i}$,

$$\hat{Y}_\theta(\boldsymbol{x}_i | \boldsymbol{X}_{N_i}) = \boldsymbol{K}_\theta(\boldsymbol{x}_i, \boldsymbol{X}_{N_i}) \boldsymbol{K}_\theta(\boldsymbol{X}_{N_i}, \boldsymbol{X}_{N_i})^{-1} Y(\boldsymbol{X}_{N_i}), \tag{2}$$

$$\text{Var}(\hat{Y}_\theta(\boldsymbol{x}_i \mid \boldsymbol{X}_{N_i})) = \boldsymbol{K}_\theta(\boldsymbol{x}_i, \boldsymbol{x}_i) - \boldsymbol{K}_\theta(\boldsymbol{x}_i, \boldsymbol{X}_{N_i}) \boldsymbol{K}_\theta(\boldsymbol{X}_{N_i}, \boldsymbol{X}_{N_i})^{-1} \boldsymbol{K}_\theta(\boldsymbol{X}_{N_i}, \boldsymbol{x}_i). \tag{3}$$

- We minimize a loss function $Q(\theta)$ over a randomly sampled batch B of training points.

$$\hat{\theta} = \arg\min_\theta Q_B(\theta). \tag{4}$$

- Using loss functions such as leave-one-out-likelihood (LOOL), evaluating the loss function requires $\mathcal{O}(bk^3) \ll \mathcal{O}(n^3)$ FLOPS, cheaper than log-likelihood maximization.
- MuyGPs predicts the response distribution for a novel point $\boldsymbol{z}$ with neighbors $\boldsymbol{X}_{N^*}$ via

$$\hat{Y}_{\hat{\theta}}(\boldsymbol{z} | \boldsymbol{X}) = \boldsymbol{K}_{\hat{\theta}}(\boldsymbol{z}, \boldsymbol{X}_{N^*}) \boldsymbol{K}_{\hat{\theta}}(\boldsymbol{X}_{N^*}, \boldsymbol{X}_{N^*})^{-1} Y(\boldsymbol{X}_{N^*}), \tag{5}$$

$$\text{Var}(\hat{Y}_{\hat{\theta}}(\boldsymbol{z} \mid \boldsymbol{X})) = \boldsymbol{K}_{\hat{\theta}}(\boldsymbol{z}, \boldsymbol{z}) - \boldsymbol{K}_{\hat{\theta}}(\boldsymbol{z}, \boldsymbol{X}_{N^*}) \boldsymbol{K}_{\hat{\theta}}(\boldsymbol{X}_{N^*}, \boldsymbol{X}_{N^*})^{-1} \boldsymbol{K}_{\hat{\theta}}(\boldsymbol{X}_{N^*}, \boldsymbol{z}). \tag{6}$$

Coverage Penalized Leave-One-Out-Likelihood for Calibration

- We formulate the LOOL loss function computed using LOOCV and local Kriging.

$$Q_B(\theta) = \sum_{i \in B} \log(\sigma_i^2(\theta)) + \frac{(Y(\boldsymbol{x}_i) - \mu_i(\theta))^2}{\sigma_i^2(\theta)}. \tag{7}$$

- $\mu_i(\theta)$ (Eqn. 2) and $\sigma_i^2(\theta)$ (Eqn. 3) are the posterior mean and variance of the ith batch point.
- Let z_α be a z-score corresponding to a given confidence level α, e.g., $z_{0.95} = 1.96$. The coverage function $c_\alpha(\theta)$ is given by the fraction of ground truth response values for $i \in B$ which lie with a confidence interval of width $z_\alpha \sigma_i(\theta)$ around $\mu_i(\theta)$.

$$c_\alpha(\theta) = \frac{1}{b} \sum_{i \in B} \mathbf{1}_{(\mu_i(\theta) - z_\alpha \sigma_i(\theta),\ \mu_i(\theta) + z_\alpha \sigma_i(\theta))}(Y(\boldsymbol{x}_i)). \tag{8}$$

- We introduce a sequence of m confidence levels $\{\alpha_j\}_{j=1}^m$ as a penalty on the LOOL.
- Let $C_\alpha(\theta) = [c_{\alpha_j}(\theta)]_{j=1}^m$ and $\alpha = [\alpha_j]_{j=1}^m$ respectively.

$$\begin{aligned} \min_\theta \quad & Q(\theta) + \frac{\beta}{2} \|C_\alpha(\theta) - \alpha\|_2^2, \\ \text{s.t} \quad & C_\alpha(\theta) = \alpha. \end{aligned} \tag{9}$$

- We formulate the augmented Lagrangian.

$$\mathcal{L}(\theta, \lambda; \beta) = Q(\theta) + \langle \lambda, C_\alpha(\theta) - \alpha \rangle + \frac{\beta}{2} \|C_\alpha(\theta) - \alpha\|_2^2. \tag{10}$$

- We use the method of multipliers (MM) to train hyperparameters θ.

Numerical Experiments

- We sample data from a univariate Gaussian process on the unit interval $[0, 1]$ and vary ν and ρ for $(\nu, \rho) = (0.135, 0.95), (0.425, 0.625), (0.635, 0.475)$, and $(0.965, 0.125)$.

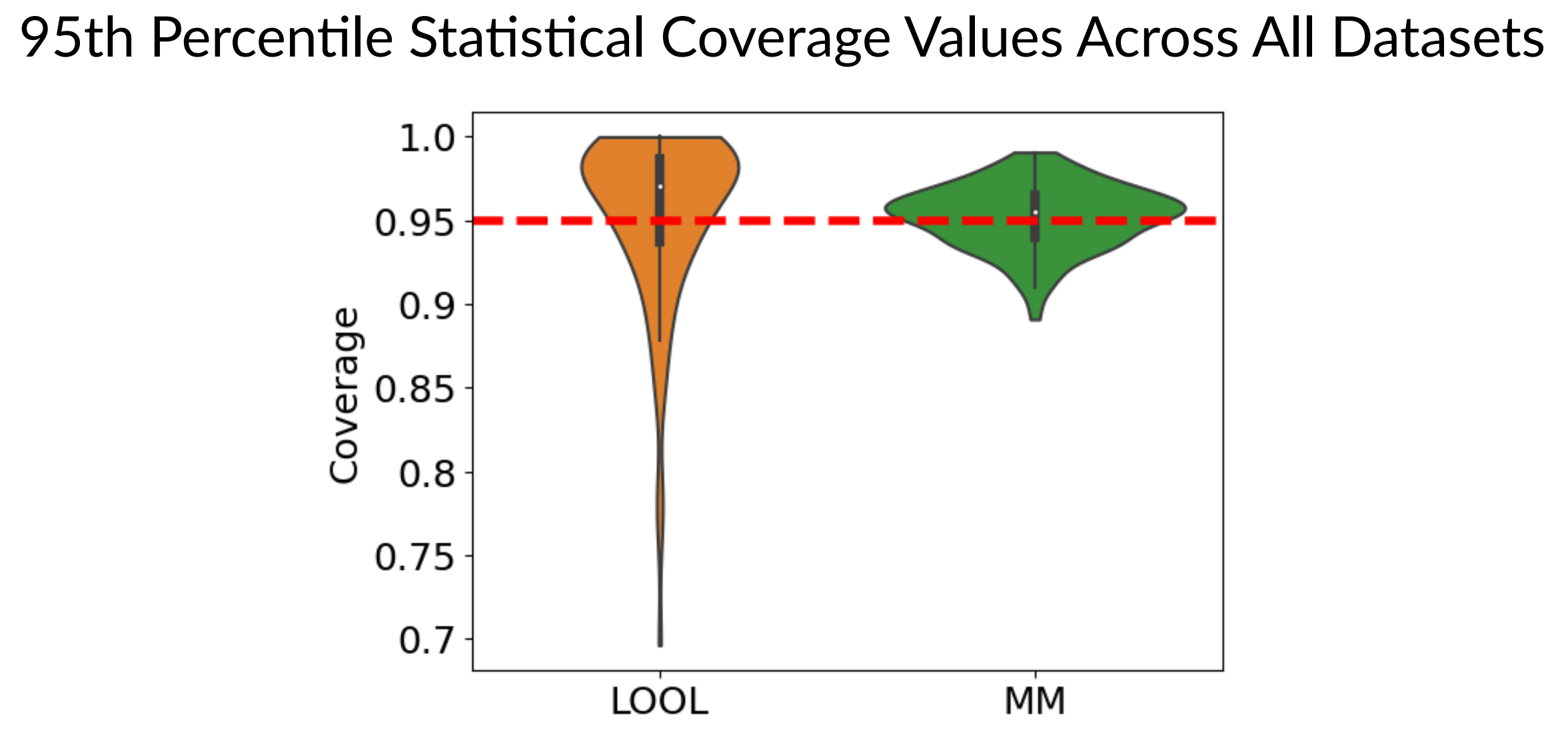

Figure 1. Violin plots of 95th percentile statistical coverage for MSE (left panel), LOOL and MM (right panel). The red dashed line indicates the target coverage value of 95 percent.

- MM achieves coverage closer to 95 percent with much lower variance than LOOL.

Climate Science Application

- Dataset: land surface temperatures measured on August 4, 2016 on a 500 × 300 grid between longitudes -95.91153 and -91.28381 and latitudes 34.29519 to 37.06811, with 105,569 training observations and 42740 testing observations (see [3]).

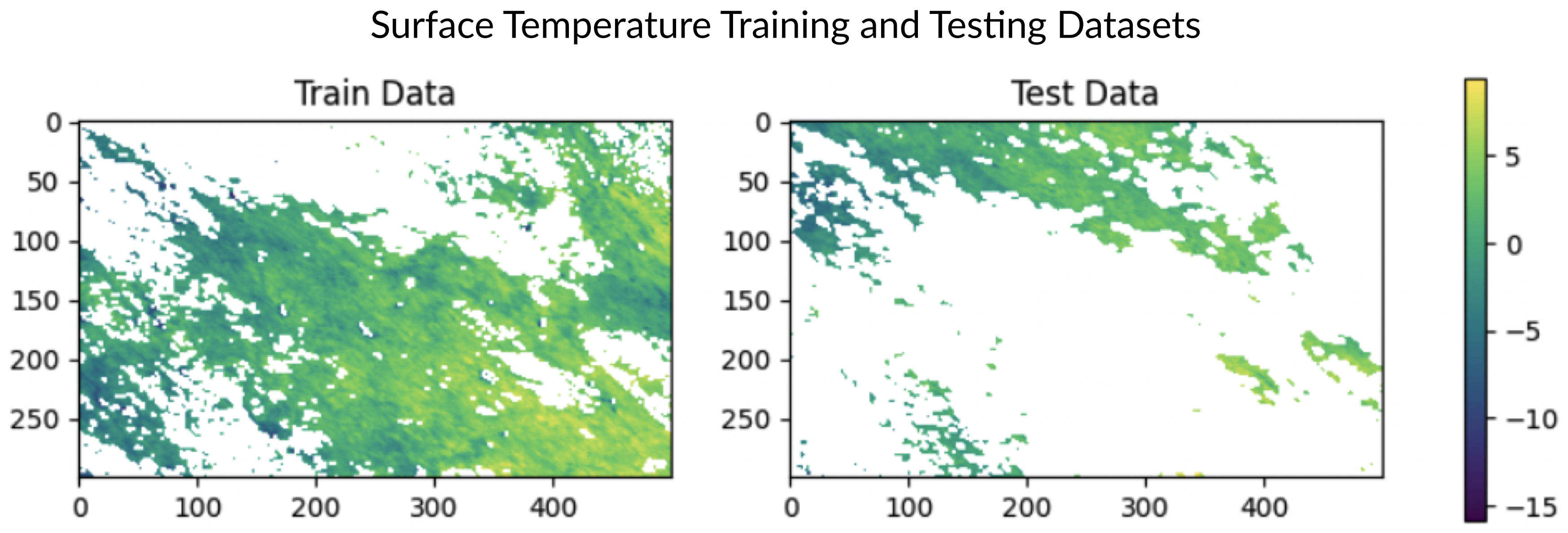

Figure 2. Zero-mean rescaled training and testing data for the surface temperature prediction problem.

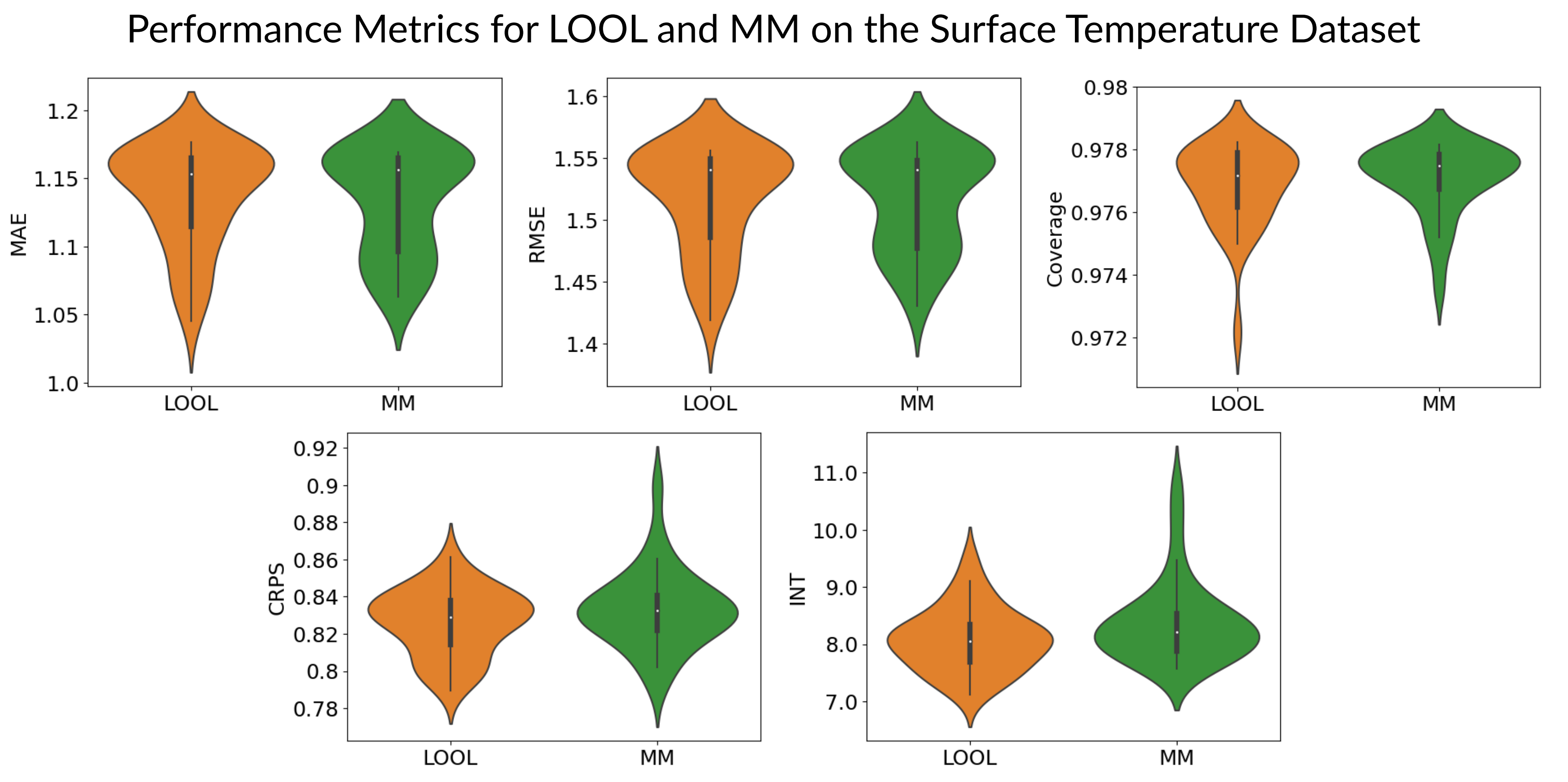

Figure 3. From top left to bottom right: the mean absolute error (MAE), root MSE, 95th percentile statistical coverage, continuous rank probability score (CRPS) [4], and interval score (INT) [4] for surface temperature dataset from the GP competition paper [3]. LOOL is shown in the orange violin on the left in each panel. MM is shown in green on the right in each panel.

- MM and LOOL outperform all methods in the competition paper [3] (best MAE and RMSE of 1.10 and 1.53) and original MuyGPs algorithm in [2] (best MAE and RMSE of 1.07 and 1.53).
- Because the optimal value of ν in this case is closer to 1, the identification of the smoothness parameter is more difficult.

References

[1] Rasmussen, Carl Edward. "Gaussian processes in machine learning." Summer school on machine learning. 2003.
[2] Muyskens, Amanda, et al. "MuyGPs: Scalable Gaussian Process Hyperparameter Estimation Using Local Cross-Validation." (2021).
[3] Heaton, Matthew J., et al. "A case study competition among methods for analyzing large spatial data." (2019).
[4] Gneiting, Tilmann, and Adrian E. Raftery. "Strictly proper scoring rules, prediction, and estimation." (2007).