Multitask Gaussian Processes for Scalable Modeling of Complex Systems with Functional Inputs

Razak Christophe Sabi Gninkou[1] Andrés F. López-Lopera[2] Rodolphe Le Riche[3] Franck Massa[4]

[1] Université Polytechnique Hauts-de-France, CERAMATHS
[2] Université de Montpellier, IMAG [3] CNRS, LIMOS [4] Université Polytechnique Hauts-de-France, LAMIH

Abstract

We introduce a multitask Gaussian process framework for probabilistic modeling of complex systems with functional covariates. The proposed approach specifically targets scenarios where the input variables represent time-dependent curves but it can be generalized to multivariate functional data such us spatial, spatio-temporel or other high-dimensional signals. Considering functional data as inputs of complex computer codes been recently considered in many scientific and engineering applications, however modeling correlation between diiferent tasks remained as an open question.

Our model relies on a fully separable kernel architecture that captures dependencies along three complementary dimensions: the task, the functional input, and the scalar (temporal) domain. The latter scalar covariate is considered to take into account time-varying outputs, a parameter requised in our mechanical application where outputs represent time-varying forces. This separability naturally induces a Kronecker product formulation of the covariance operator, enabling exact and scalable inference. Closed-form expressions for the marginal likelihood and posterior predictions are derived, while structured tensor algebra ensures numerical efficiency and GPU compatibility.

The proposed framework is validated on both synthetic and real mechanical datasets, demonstrating its ability to deliver accurate predictions and well-calibrated uncertainty estimates at a reduced computational cost. The entire approach is implemented in `PyTorch/GPyTorch`, leveraging optimized tensor operations and GPU acceleration for efficient computation of the marginal likelihood and posterior predictions. This work establishes a general and efficient probabilistic modeling paradigm for high-dimensional functional inputs, applicable to a wide range of domains from computational mechanics to other data-driven complex systems.

Keywords: Gaussian processes; functional data; separable kernels; machine learning; surrogate modeling; uncertainty quantification; complex systems.

References

1. Rasmussen, C. E., Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning.* MIT Press, Cambridge, MA.
2. Ramsay, J. O., Silverman, B. W. (2005). *Functional Data Analysis*, 2nd edition. Springer.
3. Bonilla, E. V., Chai, K. M. A., Williams, C. K. I. (2008). *Multi-task Gaussian process prediction.* In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 20, pp. 153–160, Vancouver, Canada.
4. López-Lopera, A. F., Massa, F., Turpin, I., Leconte, N. (2022). *Modeling complex mechanical computer codes with functional input via Gaussian process.* In *XLIII Ibero-Latin American Congress on Computational Methods in Engineering (CILAMCE).*

Multitask Gaussian processes for scalable modeling complex systems with functional inputs

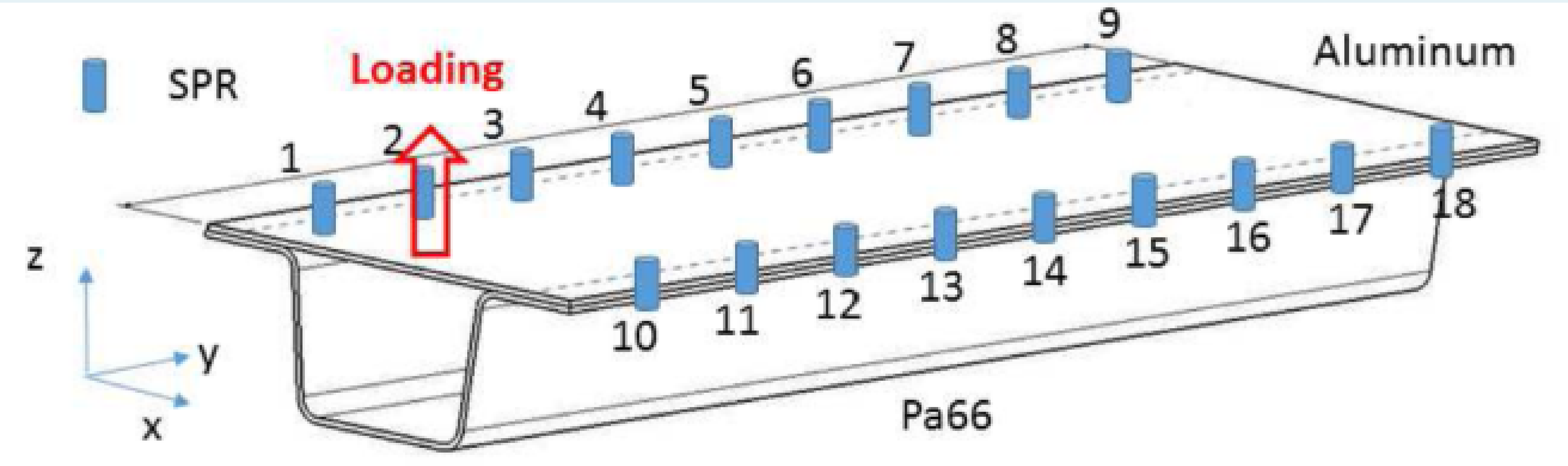
Razak Christophe Sabi Gninkou[1] Andrés F. López-Lopera[2] Rodolphe Le Riche[3] Franck Massa[4]

[1] Univ. Polytechnique Hauts-de-France, CERAMATHS [2] Univ. de Montpellier, IMAG [3] CNRS, LIMOS [4] Univ. Polytechnique Hauts-de-France, LAMIH

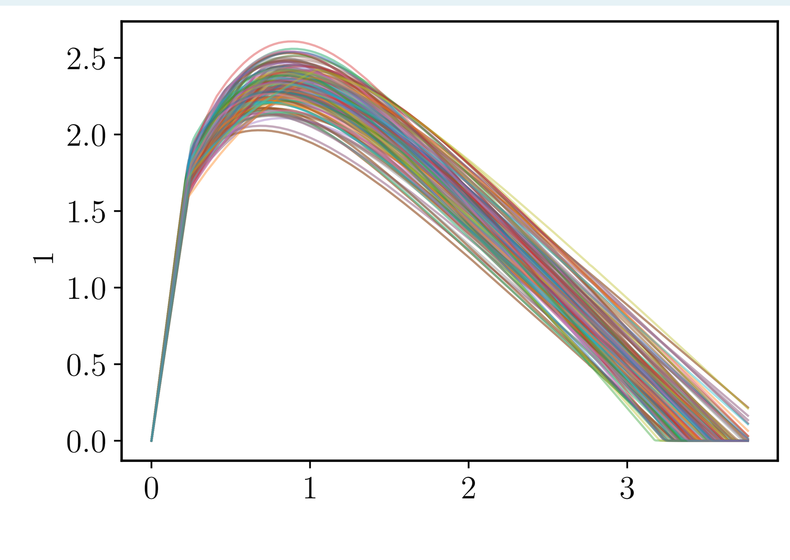
November 13-14, 2025

1. Context and Problem Statement

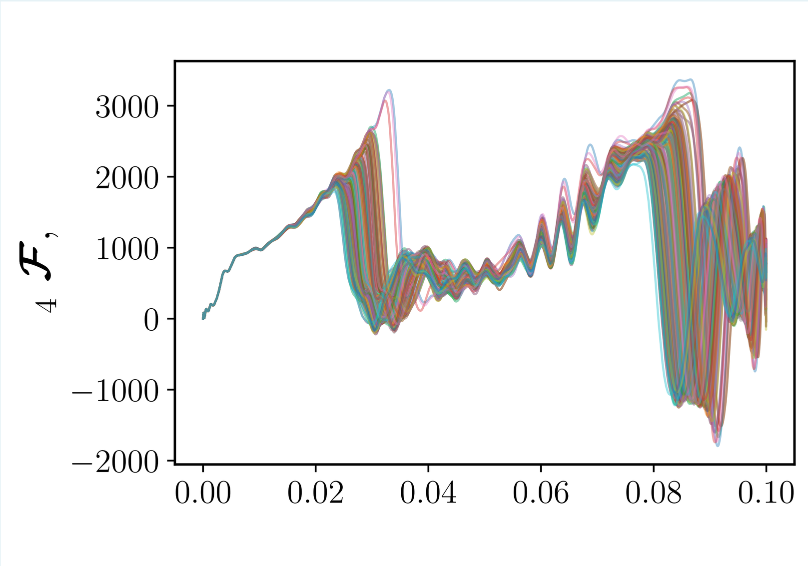
- We aim to predict correlated funtional outputs (e.g., time-series) when the shared inputs are also functional (multitask framework).
- Application and inspiration: mechanical assembly with forces and displacements at different locations (rivets, other).



SPR joint : aluminum–PA66 assembly with 18 rivets. Vertical load near rivet 2; each task s corresponds to a force series.



Functional input f_1 (load/material profiles).



Simulator outputs $g_s(\mathcal{F}, t)$ (one curve per rivet/task).

Setup :

- Let $\mathcal{F} = (f_1, \dots, f_{d_f}) \in \mathbf{F}(\mathcal{T}, \mathbb{R})^{d_f}$, with $\mathcal{T} \subset \mathbb{R}$, be a vector of functional inputs.
- For a task index $s \in \mathcal{S} = \{1, \dots, S\}$ and scalar covariate $t \in \mathbb{R}$ (e.g., time or load level), the simulator produces :

$$g_s(\mathcal{F}, t) = \text{complex mechanical response for task } s.$$

Goal :

- To **replace this costly simulator by a Gaussian process (GP)** surrogate that captures both *functional dependencies* and *task correlations*.

2. Multitask Gaussian Processes with functional inputs

Model : We place a zero-mean GP prior on $g = \{g_s(\mathcal{F}, t)\}$:

$$Y \sim \mathcal{GP}(0, k), \qquad \mathrm{Cov}(Y_s(\mathcal{F}, t), Y_{s'}(\mathcal{F}', t')) = k\big((s, \mathcal{F}, t), (s', \mathcal{F}', t')\big), \tag{1}$$

defined on $\mathcal{S} \times \mathbf{F}(\mathcal{T}, \mathbb{R})^{d_f} \times \mathbb{R}$. Thus k jointly captures task, functional, and scalar-covariate dependencies.

Separable kernel structure :

$$\mathrm{Cov}(Y_s(\mathcal{F}, t),\ Y_{s'}(\mathcal{F}', t')) = [K_\mathcal{S}]_{s,s'}\, k_f(\mathcal{F}, \mathcal{F}') k_t(t, t') \tag{2}$$

- $K_\mathcal{S}$: inter-task correlation matrix, k_f: functional kernel, k_t: scalar kernel

Construction of the kernel k_f : For $\mathcal{F} = (f_1, \dots, f_{d_f}) \in \mathbf{F}(\mathcal{T}, \mathbb{R})^{d_f}$,

$$k_f(\mathcal{F}, \mathcal{F}') = \psi(\|\mathcal{F} - \mathcal{F}'\|_\ell), \qquad \|\mathcal{F} - \mathcal{F}'\|_\ell^2 = \sum_{d=1}^{d_f} \frac{\|f_d - f_d'\|^2_{L^2(\mathcal{T})}}{\ell_d^2}. \tag{3}$$

On any Hilbert space ($L^2(\mathcal{T})$), k_f is PSD if $\psi(\sqrt{\cdot})$ is completely monotone (Schoenberg, 1938), equivalently

$$\psi(r) = \int_0^\infty e^{-\omega r^2}\, d\mu(\omega), \quad \mu \geq 0. \tag{4}$$

Examples : Squared exponential $e^{-r^2/2\ell^2}$; Exponential $e^{-r/\ell}$.

3. Practical implementation with only observations of f_d

Project each f_d on a finite basis $\{\Upsilon_{d,r}\}_{r=1}^{p_d}$:

$$f_d(t) \approx \sum_{r=1}^{p_d} \alpha_{d,r} \Upsilon_{d,r}(t), \quad \|f_d - f_d'\|^2_{L^2} \approx (\alpha_d - \alpha_d')^\top \Phi_d (\alpha_d - \alpha_d'), \tag{5}$$

with $[\Phi_d]_{r,r'} = \int_\mathcal{T} \Upsilon_{d,r} \Upsilon_{d,r'}\, dt$ (and $\Phi_d = I$ if the basis is orthonormal).

4. Hyperparameter Estimation

- **Covariance (shared time grid) :** Stack $\mathbf{y} \in \mathbb{R}^n$ with $n = S n_f n_t$ and

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_\theta), \qquad [\mathbf{K}_\theta]_{(s,i,j),(s',i',j')} = K_\mathcal{S}(s, s')\, k_f(\mathcal{F}_i, \mathcal{F}_{i'})\, k_t(t_j, t_{j'}).$$

- **Kronecker structure :** $\mathbf{K}_\theta = \mathbf{K}_\mathcal{S} \otimes \mathbf{K}_f \otimes \mathbf{K}_t$

$$[\mathbf{K}_\mathcal{S}]_{s,s'} = K_\mathcal{S}(s, s'),\ [\mathbf{K}_f]_{i,i'} = k_f(\mathcal{F}_i, \mathcal{F}_{i'}),\ [\mathbf{K}_t]_{j,j'} = k_t(t_j, t_{j'}).$$

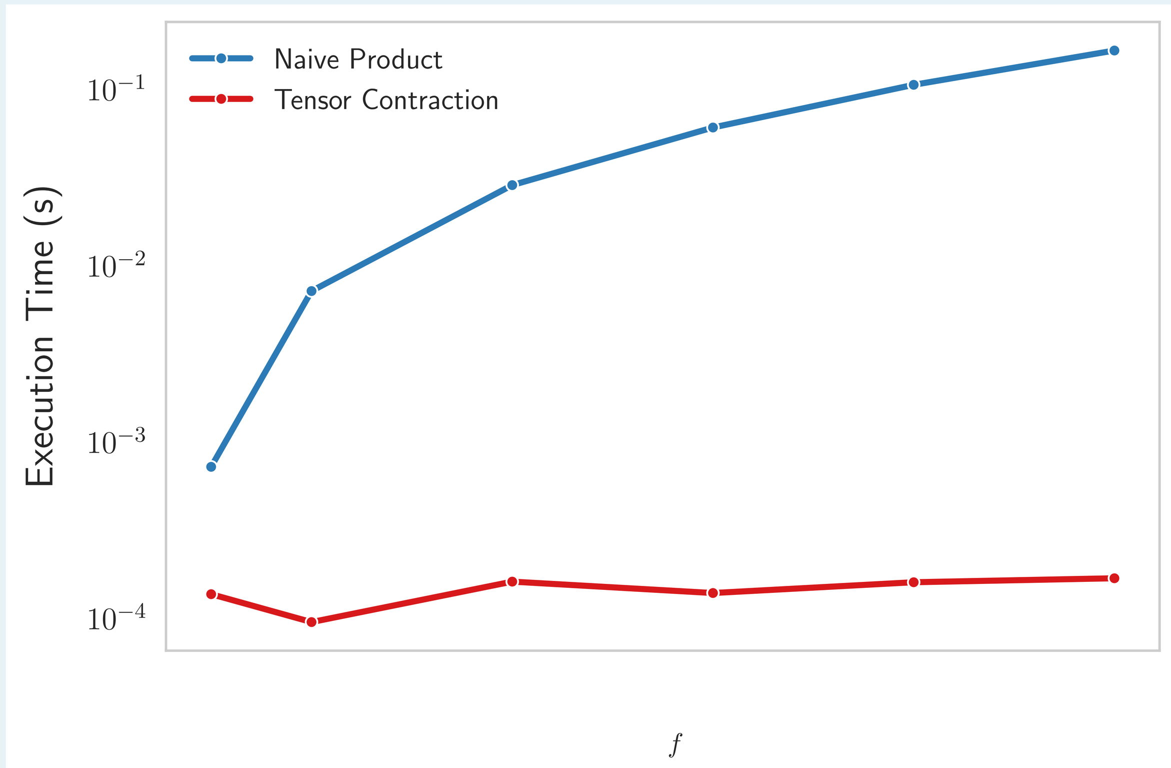
- **Parameters :** $\theta = (\mathbf{K}_\mathcal{S},\ \sigma_f^2,\ \ell_f,\ \sigma_t^2,\ \ell_t)$
- **NLML :** $\mathcal{L}(\theta) = \frac{1}{2}\mathbf{y}^\top \mathbf{K}_\theta^{-1} \mathbf{y} + \frac{1}{2}\log|\mathbf{K}_\theta| + \frac{n}{2}\log(2\pi)$.

Tensorized Cholesky & fast NLML : If $\mathbf{K}_\mathcal{S} = L_\mathcal{S} L_\mathcal{S}^\top$, $\mathbf{K}_f = L_f L_f^\top$, $\mathbf{K}_t = L_t L_t^\top$, then $\mathbf{K}_\theta = LL^\top$ with $L = L_\mathcal{S} \otimes L_f \otimes L_t$. Let $\alpha = L^{-1}\mathbf{y}$ (no explicit inverse); then

$$\mathcal{L}(\theta) = \frac{1}{2}\|\alpha\|^2 + \sum_{k=1}^n \log L_{kk} + \frac{n}{2}\log(2\pi), \tag{6}$$

where L_{kk} are the diagonal entries of L. To solve $\alpha = L^{-1}\mathbf{y}$, apply L^{-1} by mode-wise triangular solves via the Kronecker–vec identity:

$$\Big(\bigotimes_{d=1}^{D} A_d\Big) \mathrm{vec}(\mathcal{Y}) = \mathrm{vec}\big(A_D \cdots A_2(A_1 \mathcal{Y})\big). \tag{7}$$



Comparison of $\alpha = L^{-1}\mathbf{y}$ computation strategies

- **Complexity :** from $\mathcal{O}((S n_f n_t)^3)$ to $\mathcal{O}(S n_f n_t^2 + S n_t n_f^2 + n_f n_t S^2)$.

5. Posterior prediction

- **Given from estimation :** $\alpha = L^{-1}\mathbf{y}$ and the Cholesky factors L_{task}, L_f, L_t.
- **Factorized cross-covariance at test** $(s, \mathcal{F}_\star, t_\star)$:

$$u_\star = k_\mathcal{S}(\cdot, s) \otimes k_f(\cdot, \mathcal{F}_\star) \otimes k_t(\cdot, t_\star).$$

- **Mode-wise triangular solves :**

$$\zeta_\mathcal{S} = L_\mathcal{S}^{-1} k_\mathcal{S}(\cdot, s), \quad \zeta_f = L_f^{-1} k_f(\cdot, \mathcal{F}_\star), \quad \zeta_t = L_t^{-1} k_t(\cdot, t_\star),$$

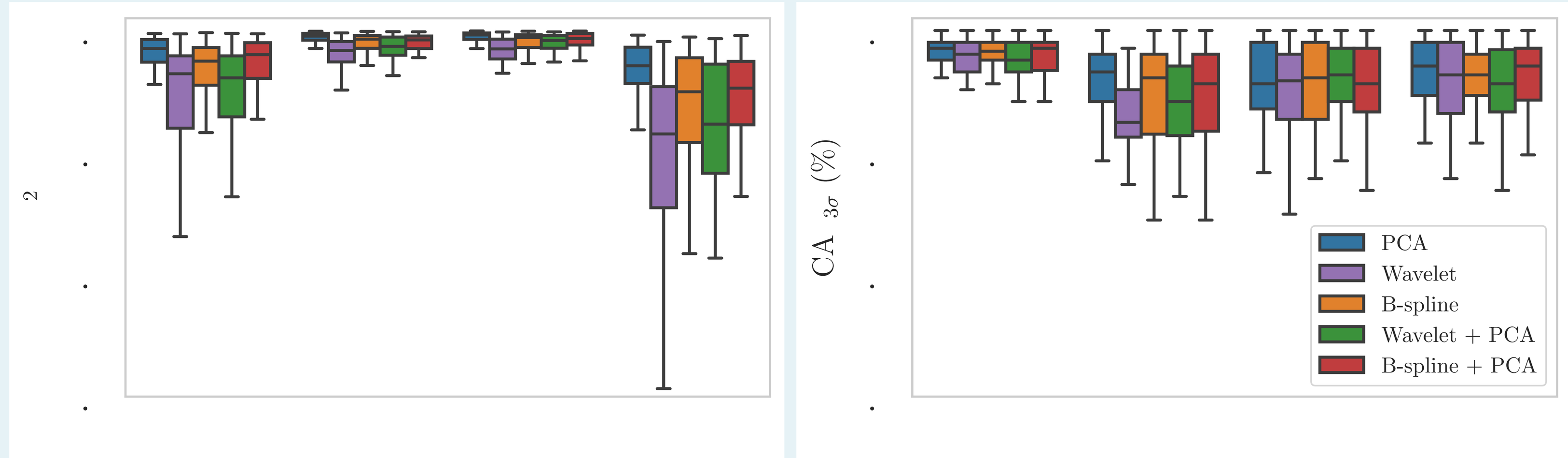
$$\widehat{u}_\star = \zeta_\mathcal{S} \otimes \zeta_f \otimes \zeta_t.$$

- **Prediction :**

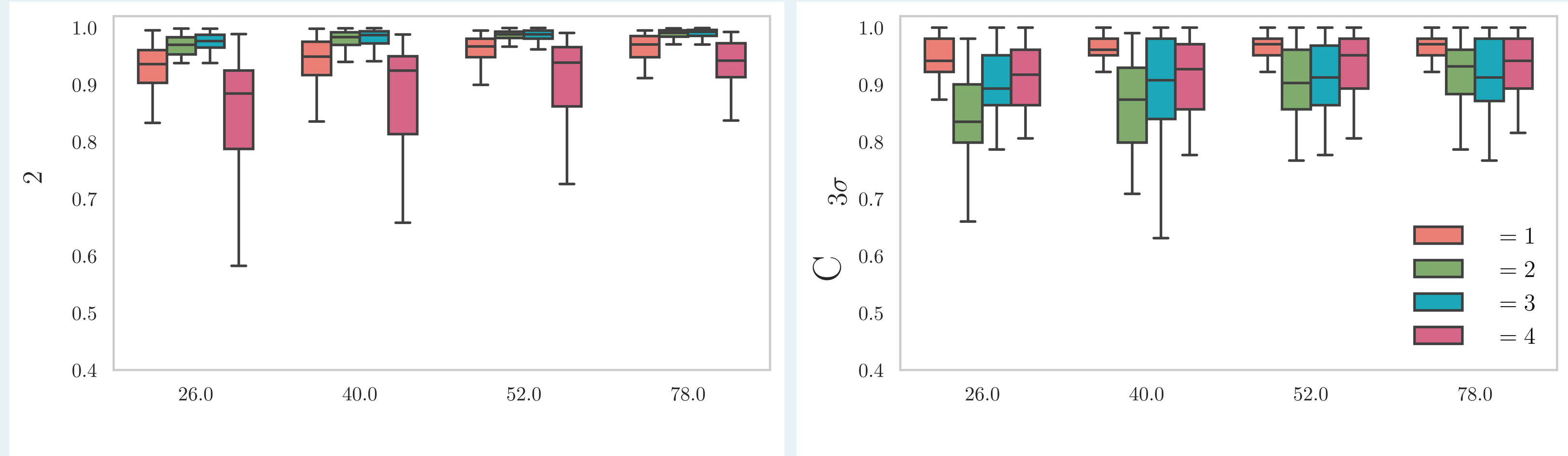
$$m_s = \widehat{u}_\star^\top \alpha, \qquad v_s = \underbrace{k_\mathcal{S}(s, s)\, k_f(\mathcal{F}_\star, \mathcal{F}_\star)\, k_t(t_\star, t_\star)}_{k_s(\mathcal{F}_\star, t_\star)} - \|\widehat{u}_\star\|^2.$$

6. Numerical experiments

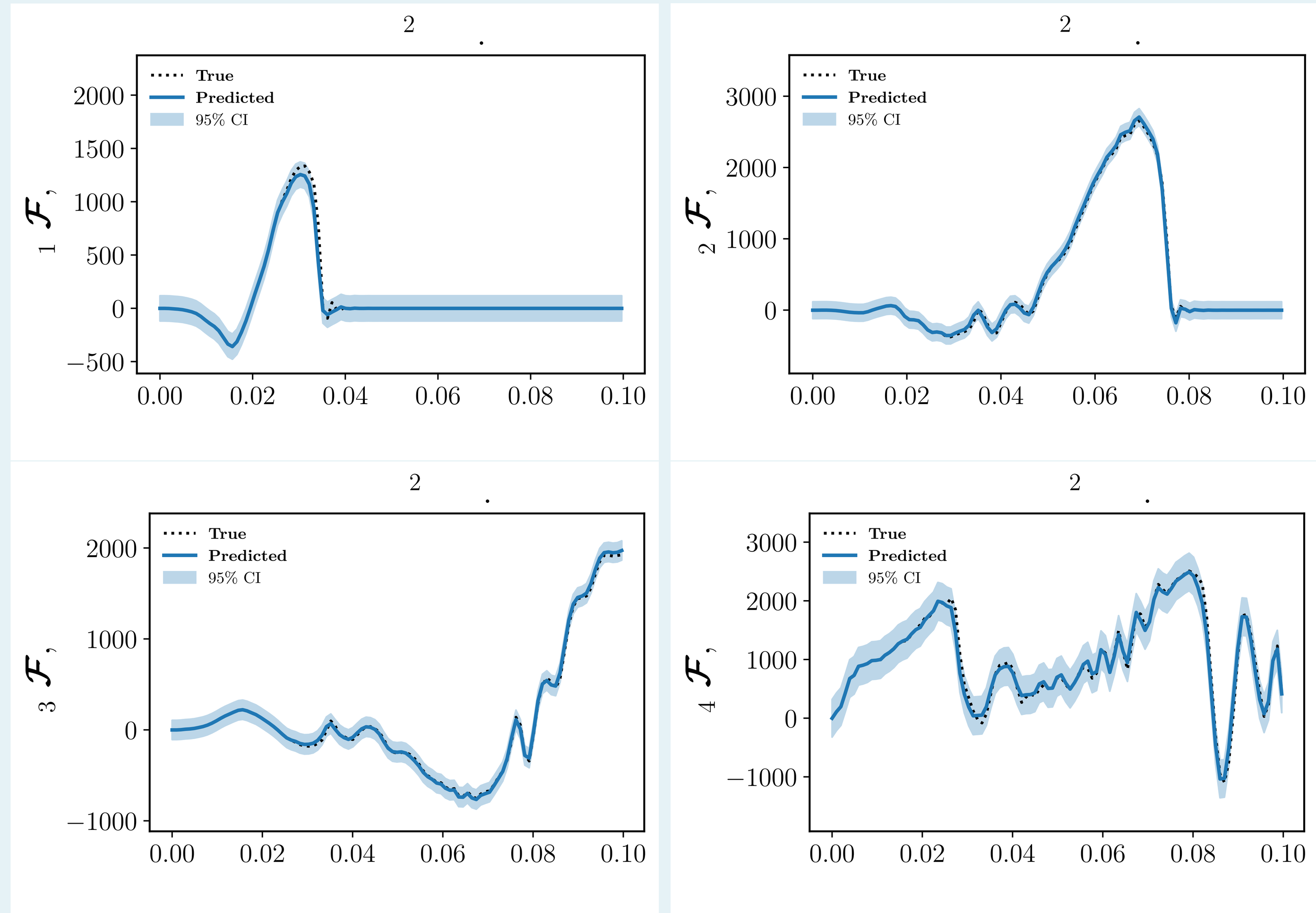
- Mechanical dataset: $d_f = 3$, $S = 4$, $n_{\text{train}} = 166$, $n_{\text{test}} = 60$.
- **Multitask GP (MTGP)** (GPyTorch) with Kronecker-based scalable inference.
- Model trained via multi-start Adam optimization with early stopping and adaptive learning rate
- Functional inputs encoded by PCA, B-splines, Wavelets, or hybrids.
- Performance criteria: Q^2 (accuracy) and CA_{95} (uncertainty coverage).



Functional encodings impact on predictions



Impact of training size



Examples of MTGP predictions of the output force-displacement curves

7. Contact & References

[1] Rasmussen, C., Williams, C. (2006). *Gaussian Processes for Machine Learning.* MIT Press.

[2] Ramsay, J., Silverman, B. (2005). *Functional Data Analysis.* Springer.

[3] Bonilla, E. V., Chai, K. M. A., Williams, C. K. I. (2008). *Multi-task Gaussian process prediction. NeurIPS.*

[4] López-Lopera, A. F., Massa, F., Turpin, I., Leconte, N. (2022). *Modeling complex mechanical computer codes with functional input via Gaussian process.* Ibero-Latin American Cong. on Comp. Methods in Eng.