AutoIP: A United Framework to Integrate Physics into Gaussian Processes

Da Long, Zheng Wang, Aditi Krishnapriyan, Robert Kirby, Shandian Zhe, Michael Mahoney
(2022)

Advanced Methods in Applied Statistics

March 6, 2025

Alexandra Haslund-Gourley and Søren Jefsen

UNIVERSITY OF COPENHAGEN

Modeling Physical Systems

Our goals are to:

- Minimizing misfit
- Require few training data points
- Ensuring our model is physically realistic
- Quantifying uncertainty

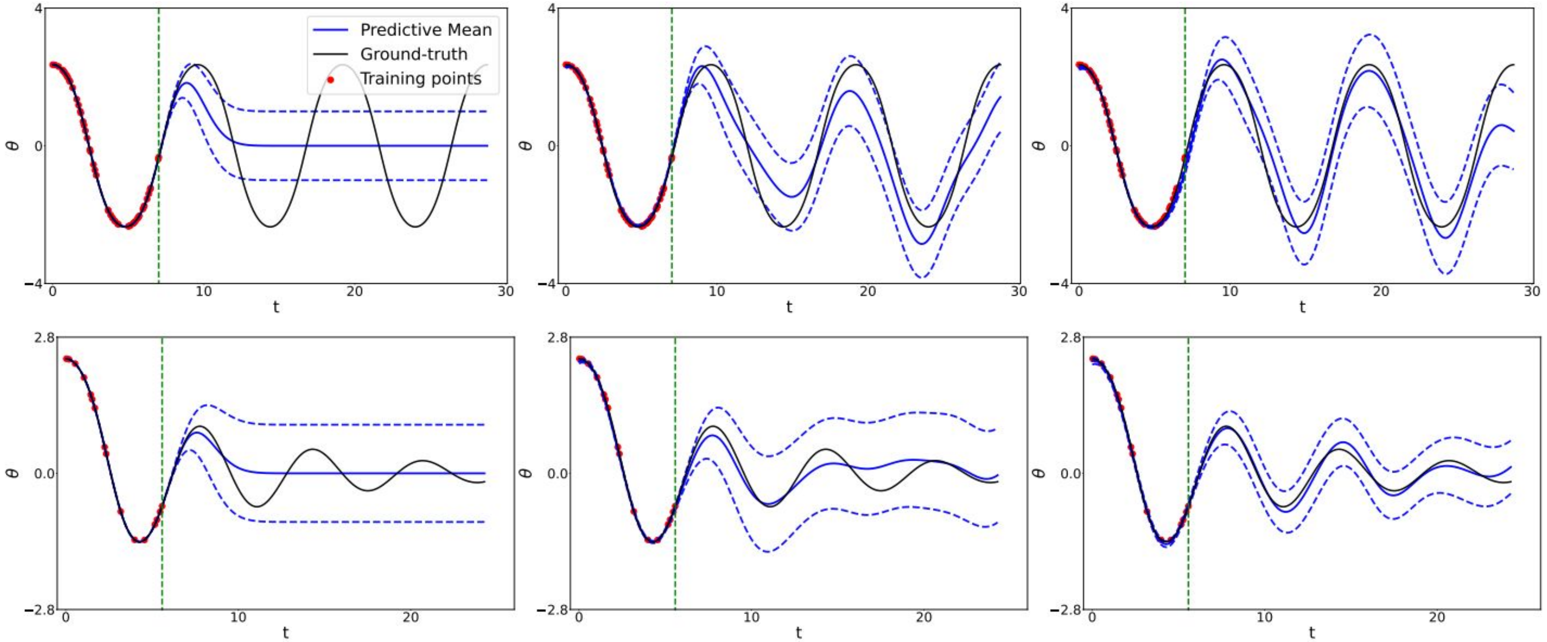
Common methods:

- Analytical Expressions
- Physics Informed Neural Networks (PINNs)

Automatically Incorporating Physics (AutoIP)

- AutoIP leverages Gaussian Processes (GPs) to predict functions that both:
  - Fits training data
  - Conforms to physically motivated partial differential equations (PDE)

AutoIP- Predicting Pendulum Dynamics



(a) GPR (b) AutoIP-I (c) AutoIP-C

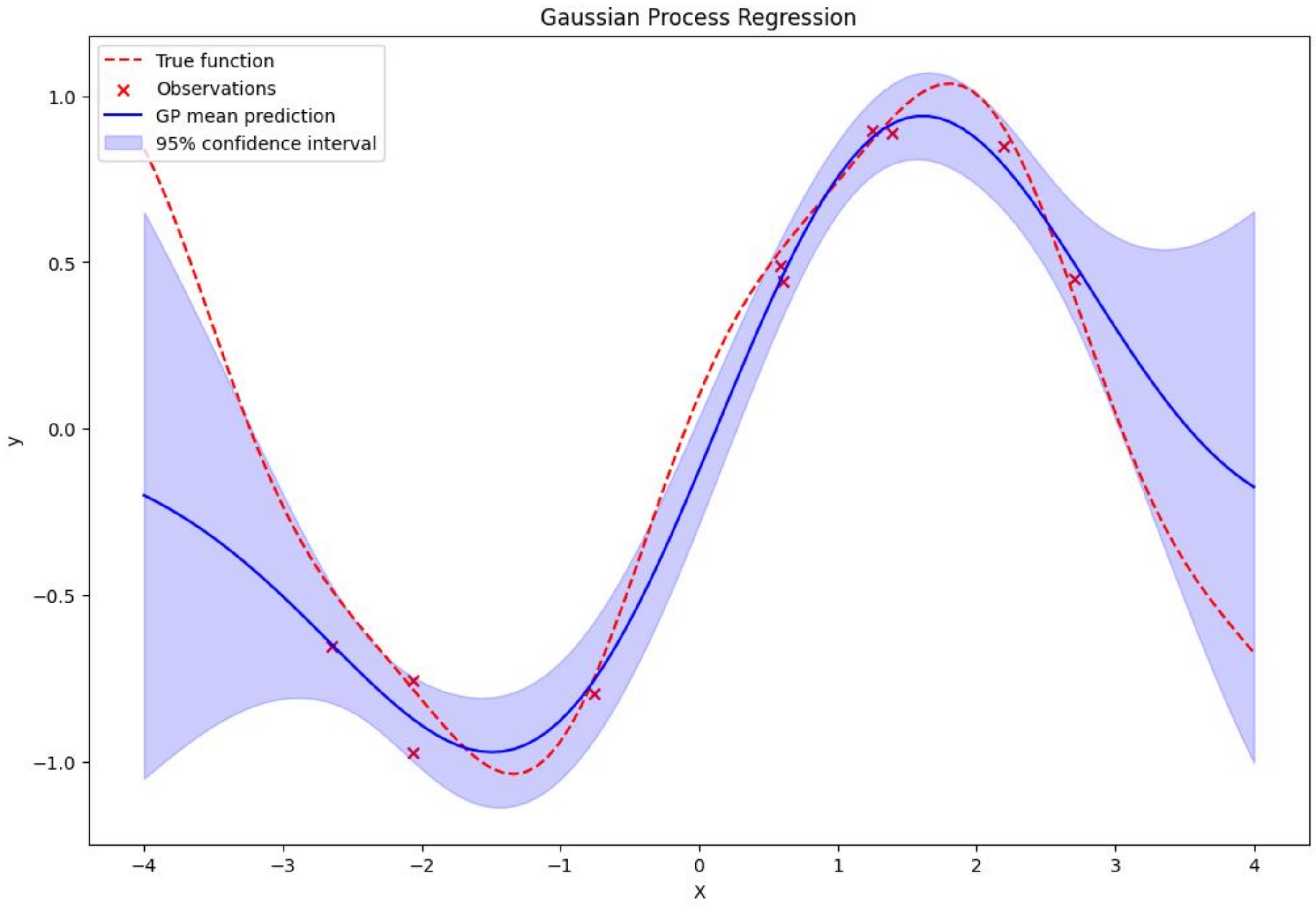
Gaussian Processes

A method that allows you to find a family of curves that fit a set of points.

Parameterized by:

- Mean Function μ(x)
  - Often set to 0
- Covariance Function
  - Kernel Function k(x, x')
  - Radial Basis Function is a

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$



Gaussian Processes – Understanding the Covariance Function

- **Key idea:** Describe f(x) values over x with covariance matrices
  - Each column of covariance matrix is a separate x value.

- To illustrate this idea, we show sampling in five dimensional joint Gaussian distribution in the GIF below.

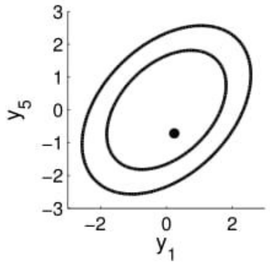
$$\boxed{f_{\mathbf{X}}(x_1,\ldots,x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}}$$

Gaussian Processes – Understanding the Covariance Function

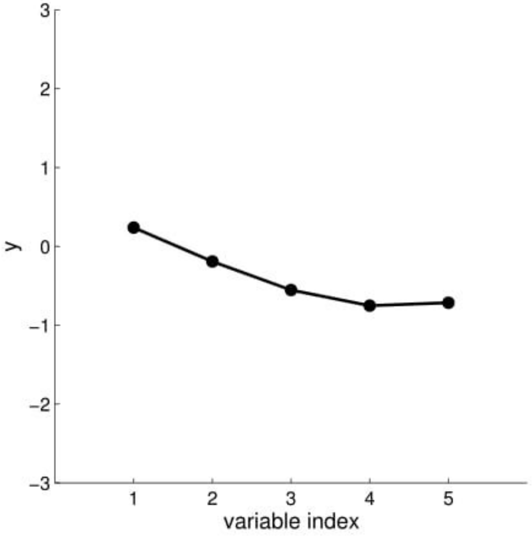
- **Key idea:** Describe f(x) values over x with covariance matrices
  - Each column of covariance matrix is a separate x value.

- To illustrate this idea, we show sampling in five dimensional joint Gaussian distribution in the GIF below.

$$f_{\mathbf{X}}(x_1, \ldots, x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}}$$



$$\Sigma = \begin{bmatrix} 1 & .9 & .8 & .6 & .4 \\ .9 & 1 & .9 & .8 & .6 \\ .8 & .9 & 1 & .9 & .8 \\ .6 & .8 & .9 & 1 & .9 \\ .4 & .6 & .8 & .9 & 1 \end{bmatrix}$$



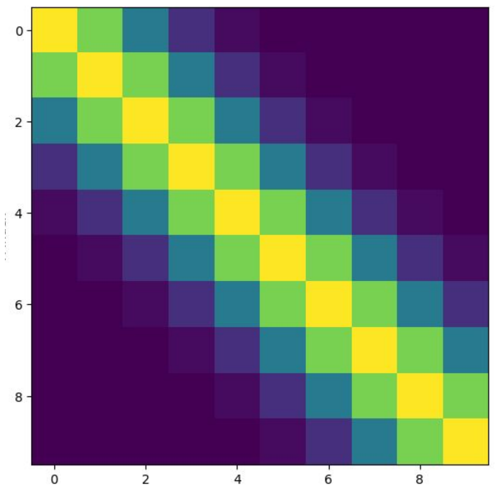
[Gaussian Processes, Not Quite for Dummies]

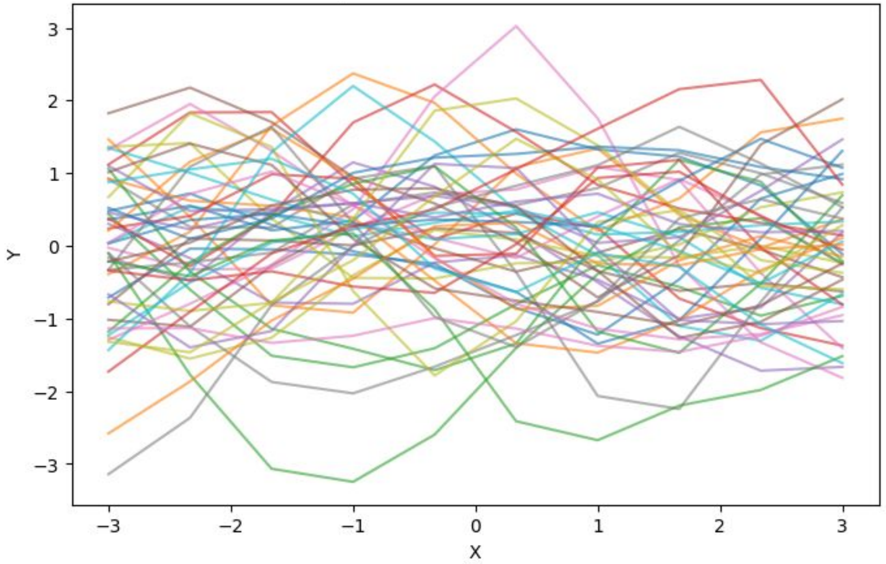
Gaussian Processes – Understanding the Covariance Function

Covariance matrices

Sampled Functions

10 Indices

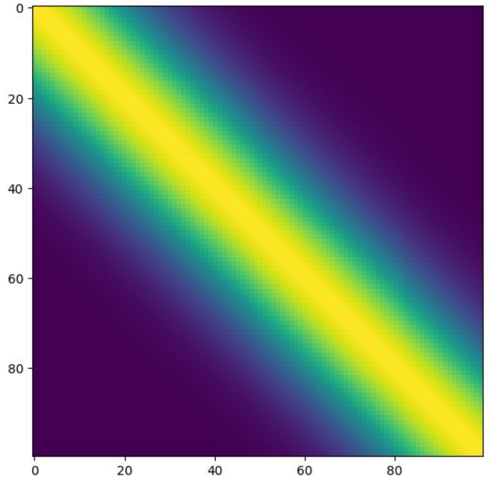


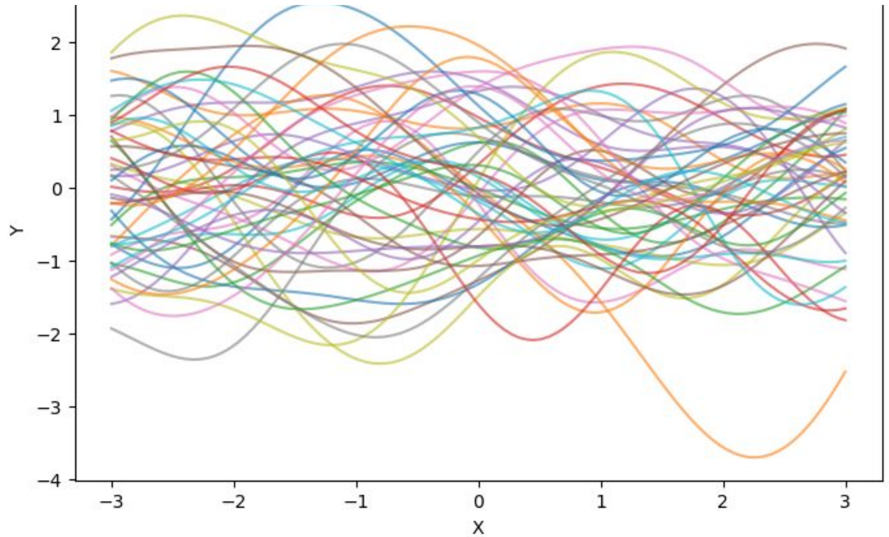


Low N prior samples

Rugged functions

100 Indices





High N prior samples

Smooth functions

Gaussian Processes Regression

What if we wish to infer function values on unseen points X*?

We create a joint distribution function p(f,f*|X*,X) and then marginalize over p(f)

Joint gaussian distribution

- Adding unknown observations

$\mathbf{K} = K(\mathbf{X},\mathbf{X}),\ \ \mathbf{K}_* = K(\mathbf{X},\mathbf{X}_*)\ \text{ and }\ \mathbf{K}_{**} = K(\mathbf{X}_*,\mathbf{X}_*)$

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^\top & \mathbf{K}_{**} \end{bmatrix} \right)$$

Apply Bayes' Rule

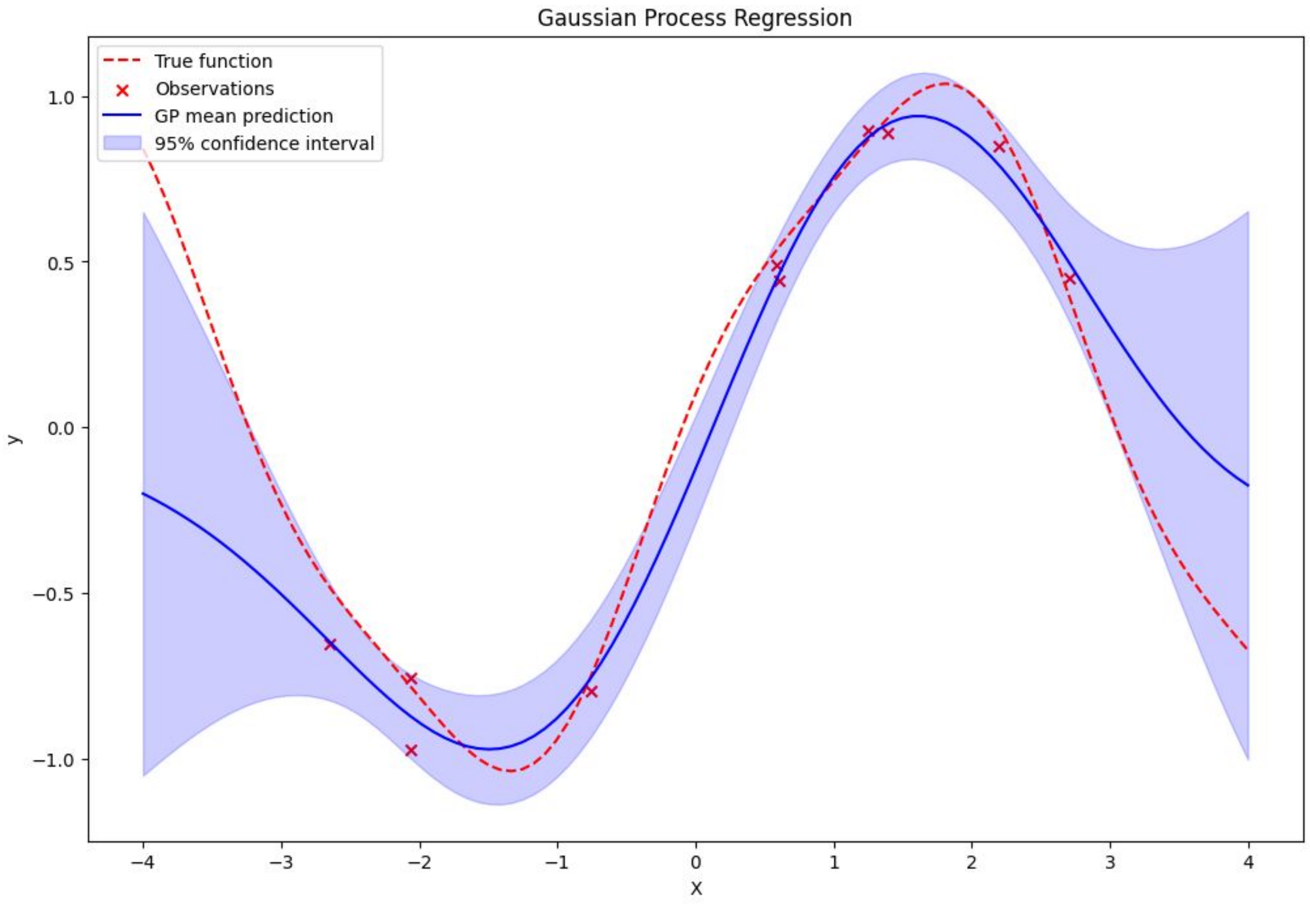
$$p(f^*|f) = \frac{p(f^*, f)}{p(f)}$$

Solve for Posterior

$$\mathbf{f}_* \mid \mathbf{f}, \mathbf{X}, \mathbf{X}_* \sim \mathcal{N}\left(\mathbf{K}_*^\top \mathbf{K}^{-1}\mathbf{f},\ \mathbf{K}_{**} - \mathbf{K}_*^\top \mathbf{K}^{-1}\mathbf{K}_*\right)$$

Gaussian Processes Regression

$$\mathbf{f}_* \mid \mathbf{f}, \mathbf{X}, \mathbf{X}_* \sim \mathcal{N}\Big(\underbrace{\mathbf{K}_*^\top \mathbf{K}^{-1}\mathbf{f}}_{\text{Mean}},\ \underbrace{\mathbf{K}_{**} - \mathbf{K}_*^\top \mathbf{K}^{-1}\mathbf{K}_*}_{\text{Variance}}\Big)$$



AutoIP

- Restricting Gaussian Process to a PDE

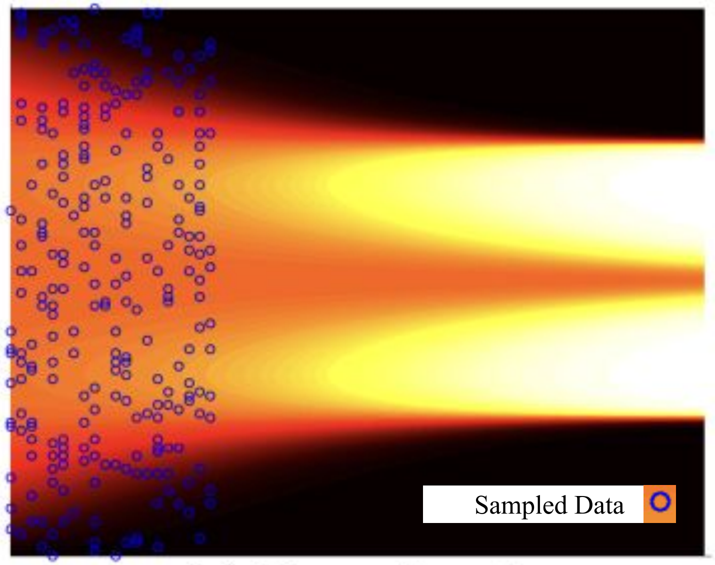
Example: Reaction-Diffusion System

Data Points Collected $(\mathbf{z}_1, y_1) \longrightarrow \mathbf{z}_n = (x_n, t_n)$

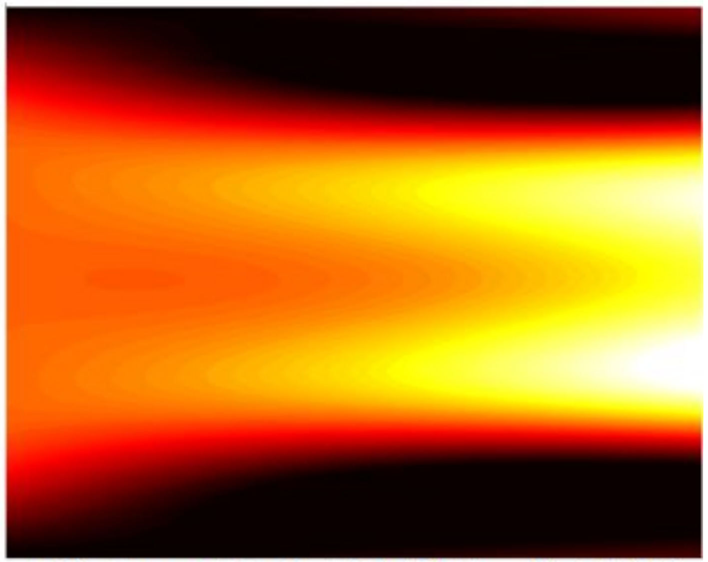
Collocation points selected $\widehat{\mathcal{Z}} = \{\widehat{\mathbf{z}}_1, \ldots, \widehat{\mathbf{z}}_M\}$

Incomplete Allen-Cahn Equation

$$\partial_t u - \nu \cdot \partial_x^2 u + \gamma \cdot u(u^2 - 1) + g(x,t) = 0$$



(a) Ground-truth


(d) AutoIP-C: RMSE = 0.1865

AutoIP

How do we incorporate derivatives into GP's?

Construct GP prior over u and g

$$u \sim \mathcal{GP}\left(0, \kappa_u(\cdot,\cdot)\right) \quad g \sim \mathcal{GP}\left(0, \kappa_g(\cdot,\cdot)\right)$$

Getting derivatives from kernel differentiation

$$\text{cov}(\hat{A}u(z_1), \hat{B}u(z_2)) = \hat{A}\hat{B}\kappa(z_1, z_2)$$

Examples of Kernel Derived Derivatives

$$\text{cov}\left(u(\mathbf{z}_1), u(\mathbf{z}_2)\right) = \kappa_u(\mathbf{z}_1, \mathbf{z}_2),$$

$$\text{cov}\left(\partial_t u(\mathbf{z}_1), \partial_t u(\mathbf{z}_2)\right) = \frac{\partial^2 \kappa_u(\mathbf{z}_1, \mathbf{z}_2)}{\partial t_1 \partial t_2},$$

Constructing Gaussian Prior <u>with derivatives</u>

$$\mathbf{f} = [\mathbf{u}; \widehat{\mathbf{u}}; \widehat{\mathbf{u}}_t; \widehat{\mathbf{u}}_{xx}; \mathbf{g}] \Longrightarrow p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{\Sigma})$$

AutoIP - Construct Joint Prior

Next, we feed the joint prior to two likelihoods:

I. To fit the actual observations from a Gaussian noise model

$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{u}, \beta^{-1}\mathbf{I})$$

II. A virtual Gaussian likelihood that integrates the physics into the differential equation

$$p(\mathbf{0}|\mathbf{f}) = \mathcal{N}(\mathbf{0}|\widehat{\mathbf{u}}_t - \nu\widehat{\mathbf{u}}_{xx} + \gamma\widehat{\mathbf{u}} \circ (\widehat{\mathbf{u}} \circ \widehat{\mathbf{u}} - \mathbf{1}) + \mathbf{g}, v\mathbf{I})$$

AutoIP

Next, construct the Joint Probability for the model:

$$p(\mathbf{f}, \mathbf{y}, \mathbf{0}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \boldsymbol{\Sigma})\mathcal{N}(\mathbf{y}|\mathbf{u}, \beta^{-1}\mathbf{I}) \cdot \mathcal{N}(\mathbf{0}|\widehat{\mathbf{u}}_t - \nu\widehat{\mathbf{u}}_{xx} + \gamma\widehat{\mathbf{u}} \circ (\widehat{\mathbf{u}} \circ \widehat{\mathbf{u}} - \mathbf{1}) + \mathbf{g}, v\mathbf{I})$$

This joint probability distribution is maximized using a stochastic variational learning algorithm.

Experimenting with AutoIP - Reaction Diffusion Equations

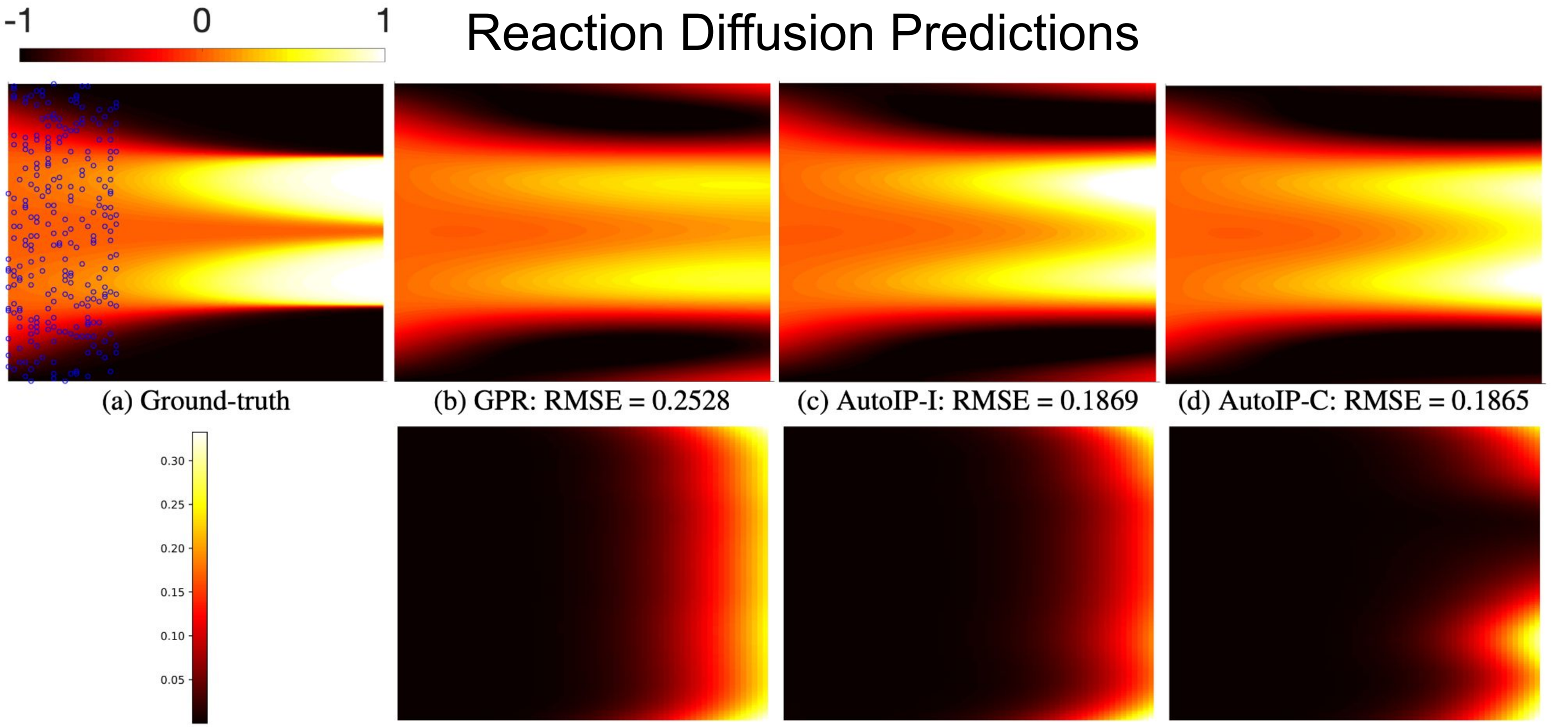
Complete Equation (AutoIP-C) $$\frac{\partial u}{\partial t} - 0.0001\frac{\partial^2 u}{\partial x^2} + 5u^3 - 5u = 0,$$

Incomplete Equation (AutoIP-I) $$\frac{\partial u}{\partial t} - 0.0001\frac{\partial^2 u}{\partial x^2} + g(x,t) = 0,$$

Comparing Modeling Methods

	GPR	AutoIP-I	AutoIP-C	PINN (100)	PINN (10K)
RMSE	0.2528	0.1869	0.1865	0.4388	**0.0169**
# Collocation Points	**100**	**100**	**100**	**100**	**10,000**

Reaction Diffusion Predictions



(a) Ground-truth (b) GPR: RMSE = 0.2528 (c) AutoIP-I: RMSE = 0.1869 (d) AutoIP-C: RMSE = 0.1865

(e) GPR predictive variance (f) AutoIP-I predictive variance (g) AutoIP-C predictive variance

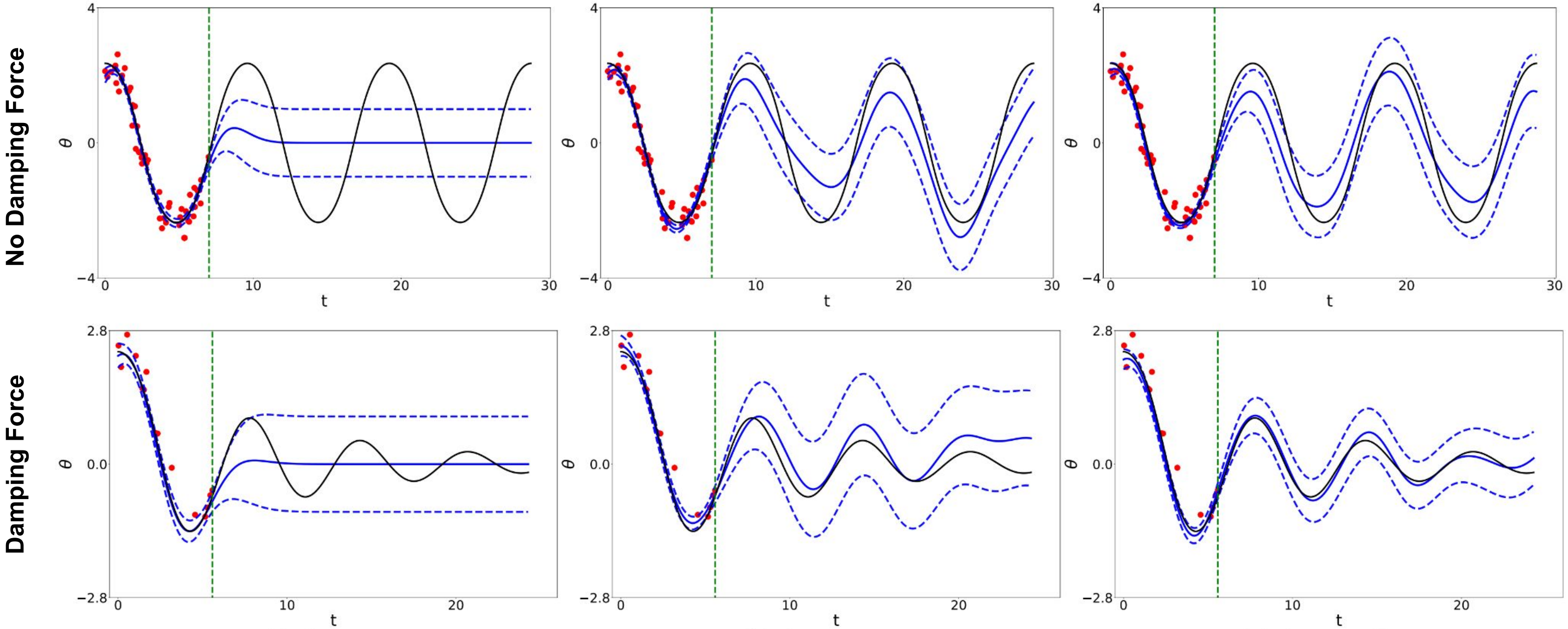
Experimenting with AutoIP - Nonlinear Pendulum

Method (CL Points)	No damping/Exact training	No damping/Noisy training	Damping/Exact training	Damping/Noisy training
PINN-50 (10k)	1.367 ± 0.575	1.658 ± 0.074	$\mathbf{0.00007 \pm 0.00001}$	0.157 ± 0.051
PINN-100 (10k)	1.862 ± 0.584	1.993 ± 0.357	$\mathbf{0.00007 \pm 0.00002}$	0.186 ± 0.045
AutoIP-I (20)	0.585 ± 0.017	0.691 ± 0.030	0.212 ± 0.014	0.268 ± 0.013
AutoIP-C (20)	$\mathbf{0.416 \pm 0.050}$	$\mathbf{0.488 \pm 0.036}$	0.096 ± 0.004	$\mathbf{0.133 \pm 0.010}$

AutoIP-C $$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + \sin(\theta) + b\frac{\mathrm{d}\theta}{\mathrm{d}t} = 0,$$

AutoIP-I $$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + g(t) = 0.$$

Nonlinear Pendulum Predictions



(a) GPR (b) AutoIP-I (c) AutoIP-C

No damping	RMSE
GPR	1.354 ± 0.005
AutoIP-I	0.585 ± 0.017
AutoIP-C	$\mathbf{0.416 \pm 0.050}$
With damping	
GPR	0.262 ± 0.0003
AutoIP-I	0.212 ± 0.014
AutoIP-C	$\mathbf{0.096 \pm 0.0035}$

Remarks

- Adding physical constraints to GPs improves performance and ability to extrapolate outside of training regions
- Adding more collocation points improves performance
- This method scales with at least O(N^3) points, making it computationally expensive for large datasets.
- AutoIP outperformed PINNs on sparse and noisy data