Multifidelity Gaussian Process Modeling
(Multifidelity Kriging)

Elizabeth Qian, Assistant Professor

School of Aerospace Engineering

School of Computational Science and Engineering

Georgia Tech

Engineering **models** describe mathematical **functions**

$$x \longrightarrow \boxed{f : x \mapsto y} \longrightarrow y$$

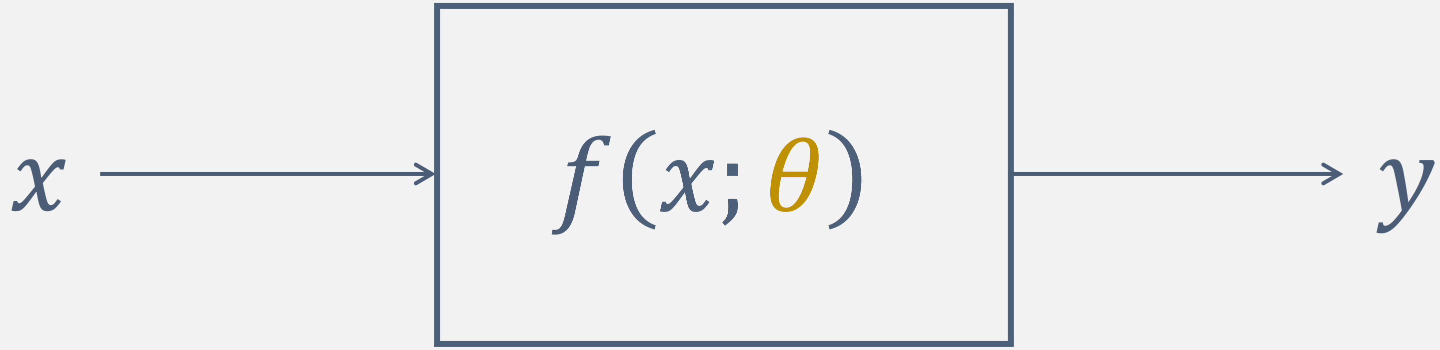
Model inputs could be…

- Design parameters
- Initial flow states
- Applied structural loads

Model outputs could be…

- Cost or efficiency metrics
- Time-dependent flow field
- Structural stresses or deformations

Data-driven modeling uses data to define the model



$$x \longrightarrow \boxed{f(x;\theta)} \longrightarrow y$$

Sometimes, data are used to determine explicit model parameters:

- Physical system parameters
- Environmental parameters
- Regression coefficients
- Neural network weights and biases

Gaussian process regression (GPR) defines the model by conditioning on the data

$$x \longrightarrow \boxed{f(x|D)} \longrightarrow y$$

- Nonparametric approach supported by universal approximation theory
- Enables uncertainty quantification: $f(x|D)$ is now a probability distribution
- Enables imposition of prior information / regularization: GPR is a Bayesian approach

Tutorial outline: Multifidelity GP modeling

1. Probability and GP foundations

2. Gaussian process regression (single-fidelity)

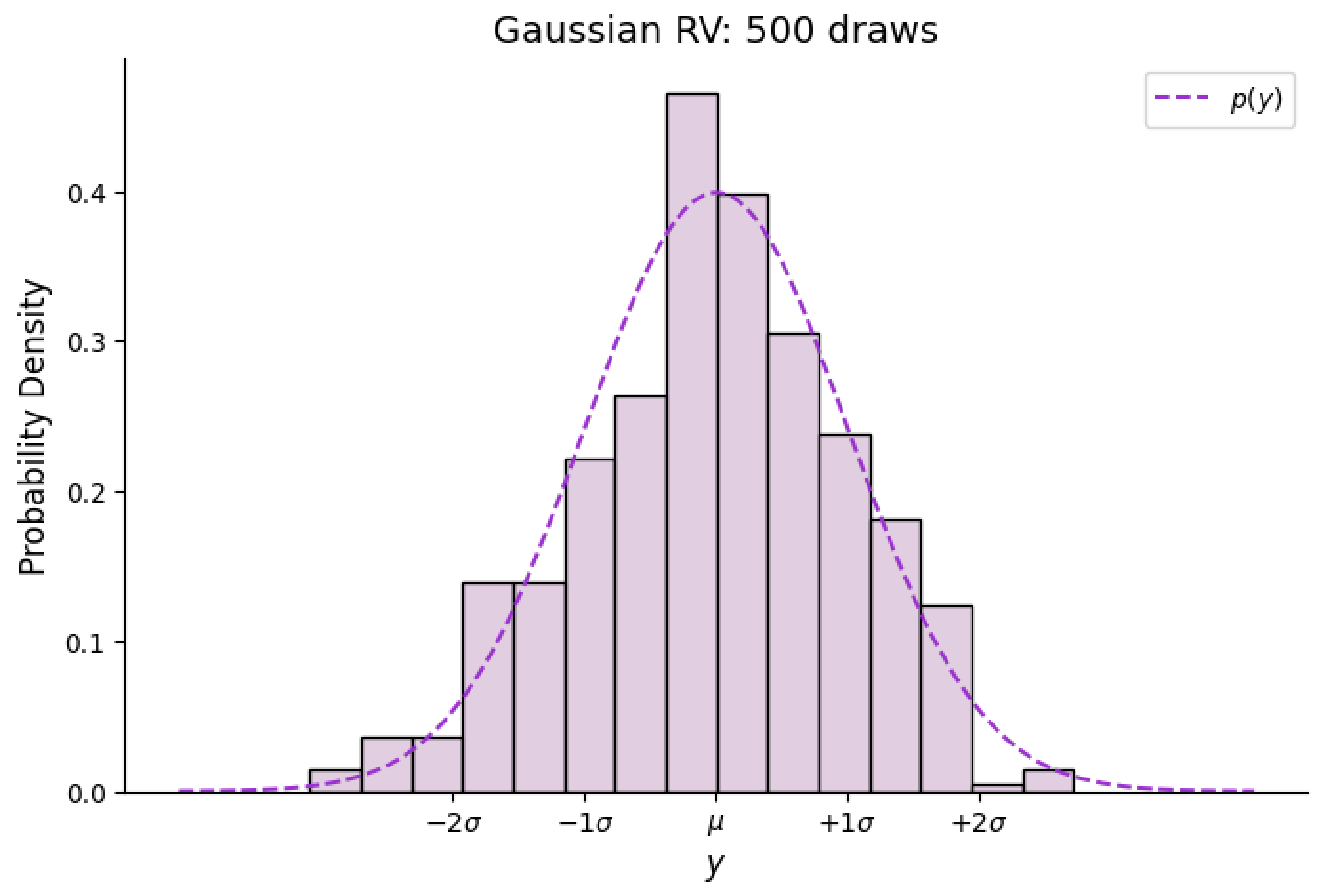
3. Multifidelity Gaussian process modeling approaches

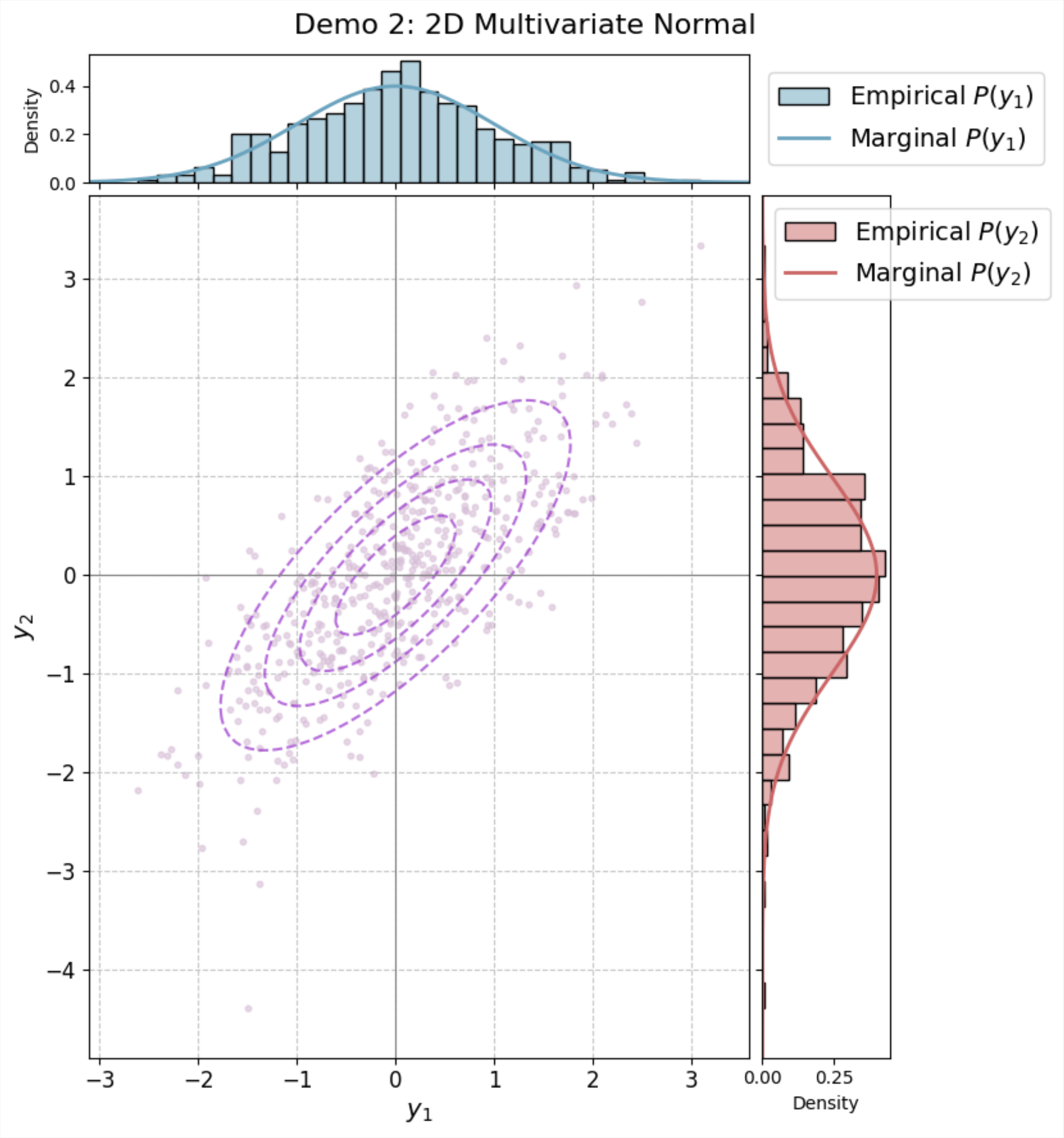
Gaussian random variables

$$p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$

In 1D (univariate): $N(\mu, \sigma^2)$

- μ is the mean
- σ is the standard deviation
- 68% of samples fall in $[\mu - \sigma, \mu + \sigma]$
- 95% of samples fall in $[\mu - 2\sigma, \mu + 2\sigma]$





Multivariate Gaussian random variables

$$p(y) = \frac{1}{\sqrt{(2\pi)^k \det(\Sigma)}} \exp\left(-\frac{1}{2}(y-\mu)^\top \Sigma^{-1}(y-\mu)\right)$$

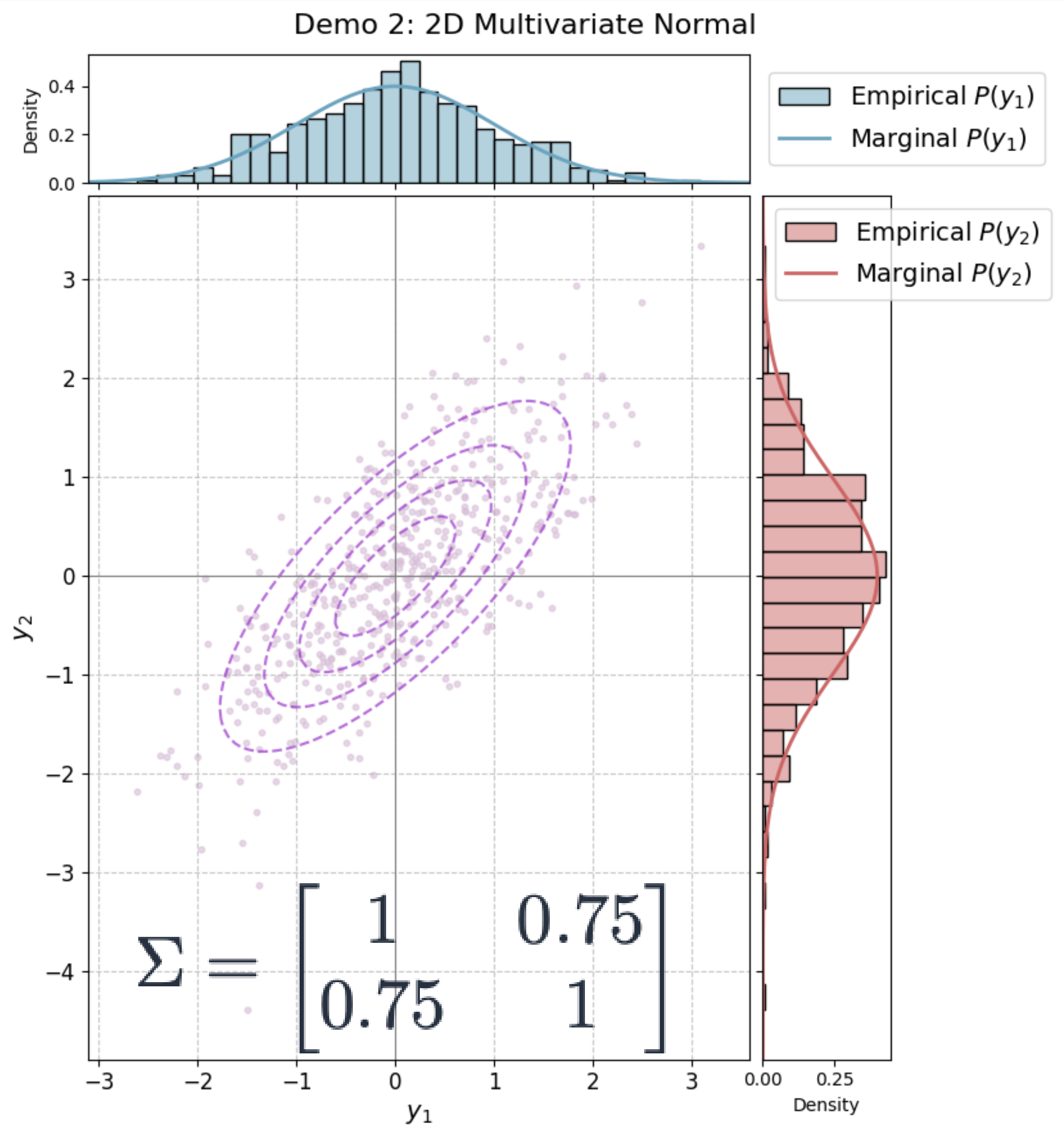
In k-D (multivariate): $N(\mu, \Sigma)$

- μ is the **mean vector**
- Σ is the **covariance matrix**

Each coordinate has an a priori **marginal distribution** given by a univariate Gaussian

$$p(y_1) \sim N(\mu_1, \Sigma_{11})$$

Correlations in multivariate Gaussians



$$p(y) = \frac{1}{\sqrt{(2\pi)^k \det(\Sigma)}} \exp\left(-\frac{1}{2}(y-\mu)^\top \Sigma^{-1}(y-\mu)\right)$$

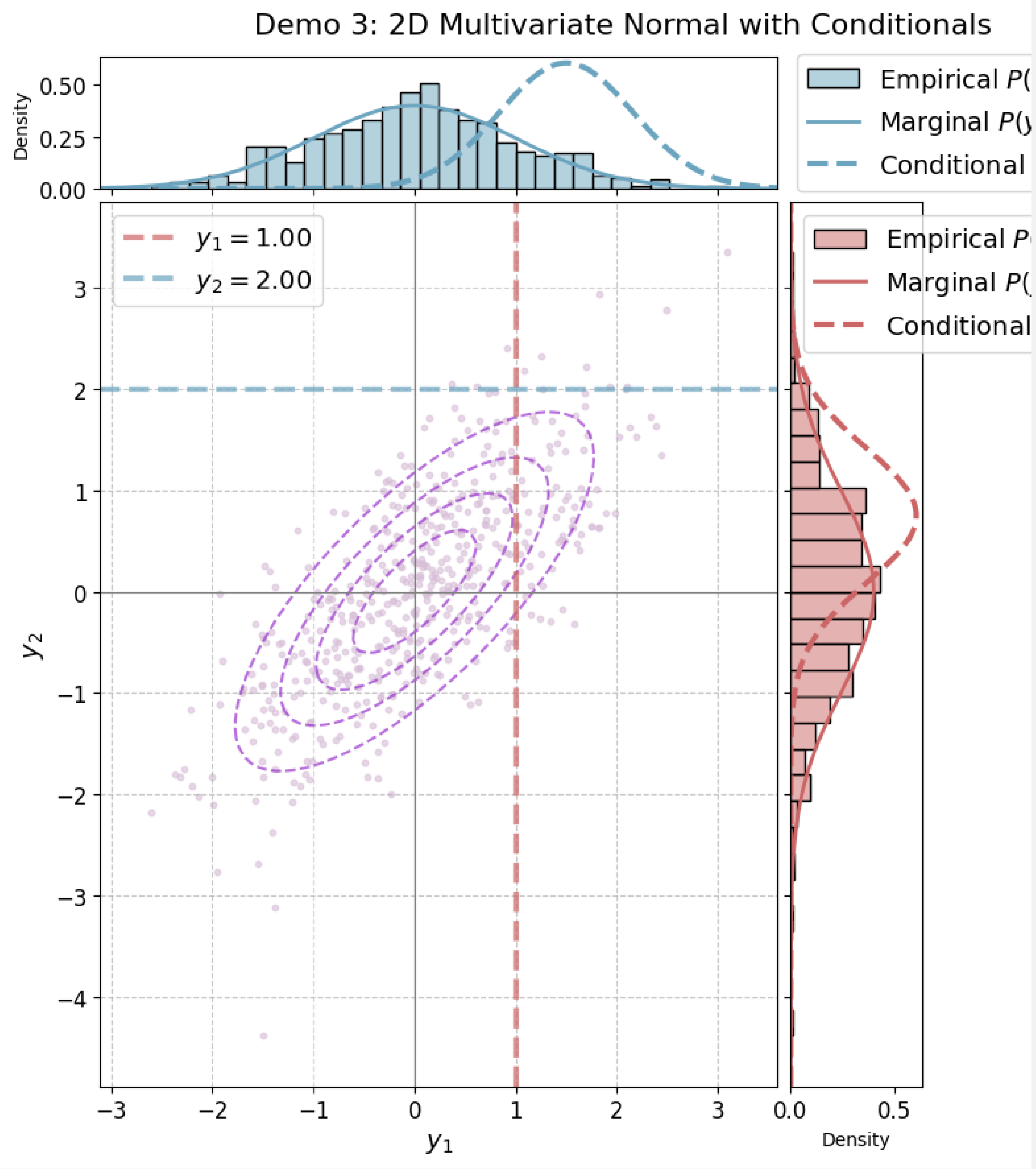
When Σ has off-diagonal entries, the coordinates are **correlated**

Pearson correlation coefficient:

$$\rho_{ij} = \Sigma_{ij} / \sqrt{\Sigma_{ii}\Sigma_{jj}}$$

If two coordinates are correlated, observing one coordinate updates our belief on the other

Exploiting correlations by conditioning on observations



If we observe y_2, we can update our belief on y_1 from the marginal distribution

$$p(y_1) \sim N(\mu_1, \Sigma_{11})$$

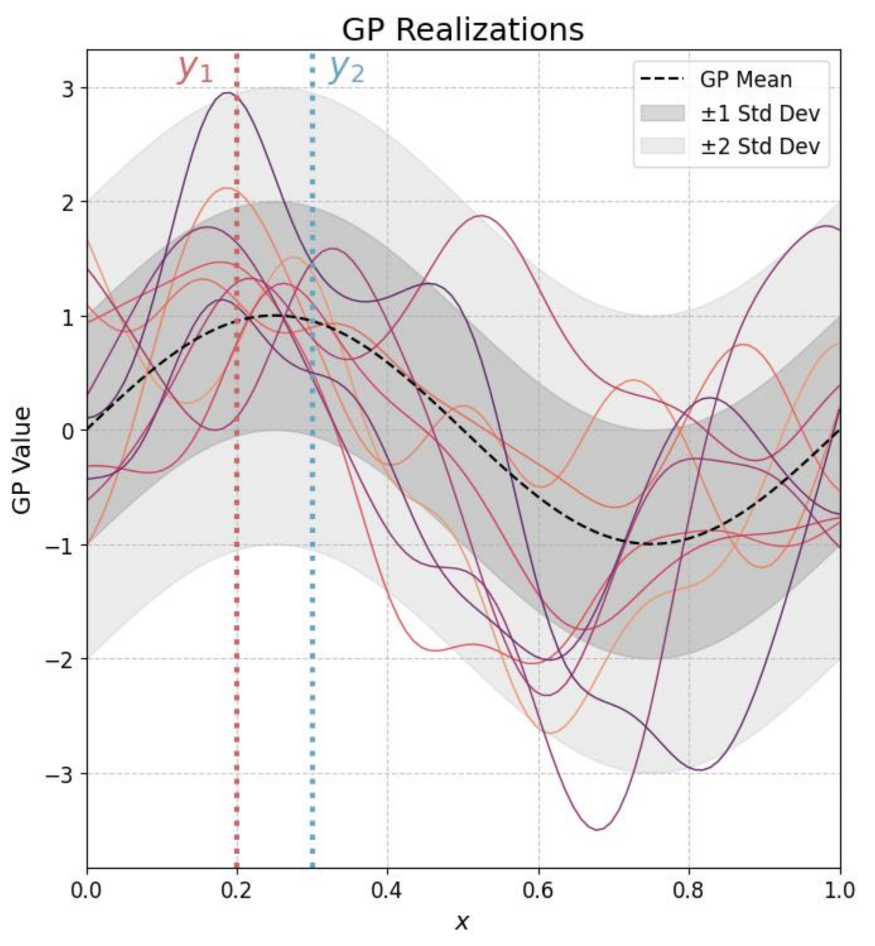
to the **conditional distribution**:

$$p(y_1|y_2) \sim N(\mu_{1|2}, \Sigma_{1|2})$$

$$\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(y_2 - \mu_2)$$

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

Gaussian processes are multivariate Gaussians extended from vectors to functions



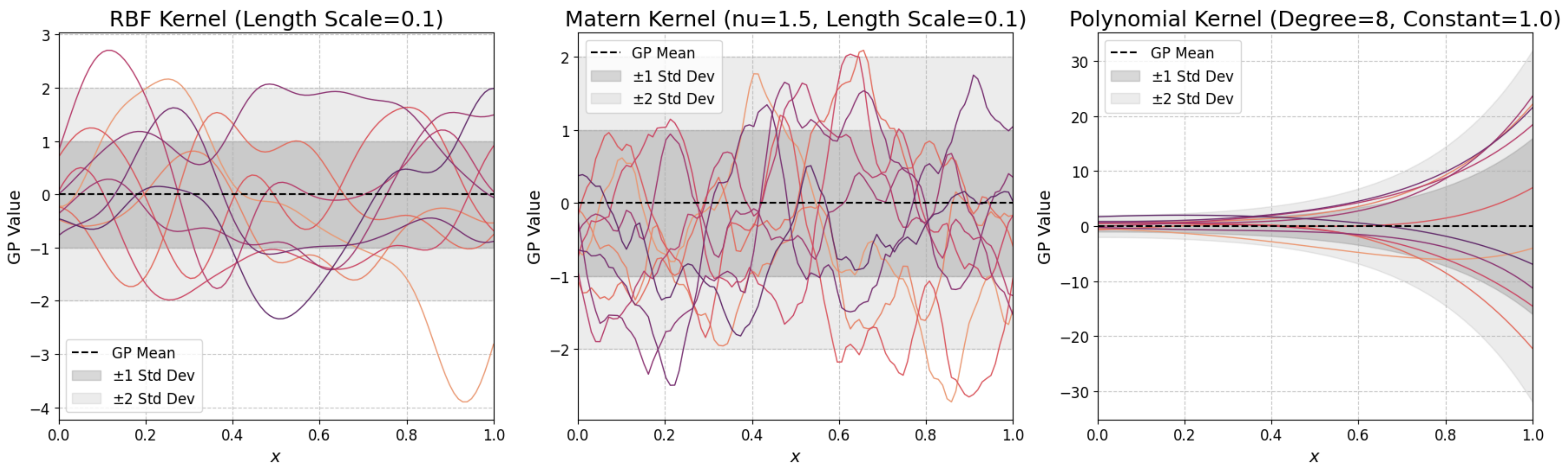
We write $y \sim \mathrm{GP}(\mu, k)$ where

- $\mu = \mu(x)$ is a **mean function**
- $k = k(x, x')$ is a **covariance kernel**

The covariance between $y_1 = f(x_1)$ and $y_2 = f(x_2)$ is given by $k(x_1, x_2)$

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2}\left(\frac{|x - x'|}{\ell}\right)^2\right)$$

Gaussian processes are powerful modeling tools



Computational costs of GPs

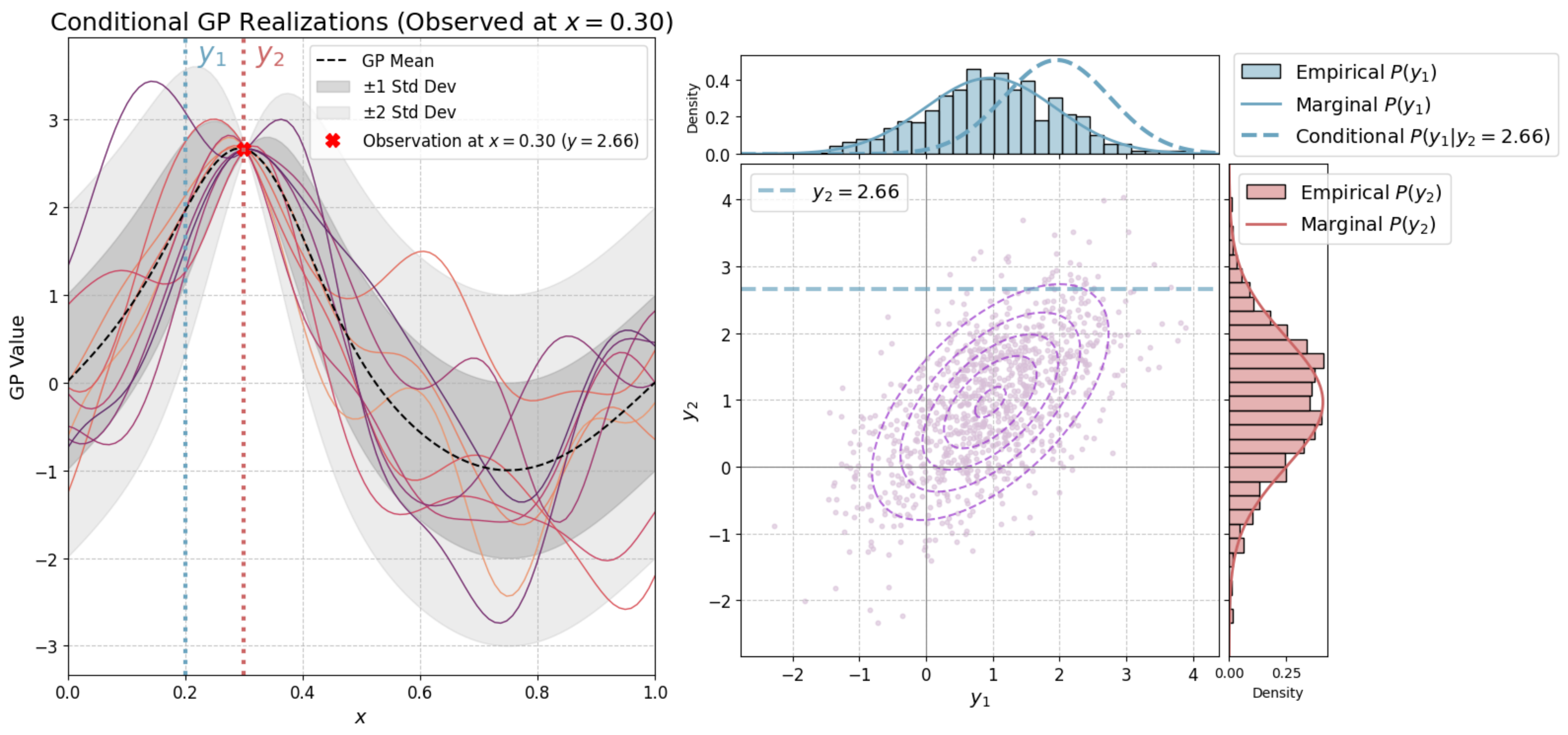
Although the abstract GP represents a continuous process, we compute by discretizing: $\boldsymbol{x}' = [x_1', \dots, x_n']^\top$

Then $\boldsymbol{y}' = y(\boldsymbol{x}') \sim N\big(\mu(\boldsymbol{x}'), k(\boldsymbol{x}', \boldsymbol{x}')\big)$

Let $\boldsymbol{L}\boldsymbol{L}^\top = k(\boldsymbol{x}', \boldsymbol{x}')$. **Cost to compute:** $O(n^3)$

Then if $\boldsymbol{\xi}$ is a standard n-dim Gaussian, $\boldsymbol{L}\boldsymbol{\xi}$ are samples of $y(\boldsymbol{x}')$

Gaussian process regression exploits correlations by conditioning on point observations



Gaussian process regression: key formulas

Multivariate normal

Conditioning on observed y_2 updates belief on y_1:

$$p(y_1|y_2) \sim N(\mu_{1|2}, \Sigma_{1|2})$$

$$\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(y_2 - \mu_2)$$

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

Gaussian process

Conditioning on observed

$$X = [x_1, \ldots, x_n]^\top$$

$$Y = [y_1, \ldots, y_n]^\top$$

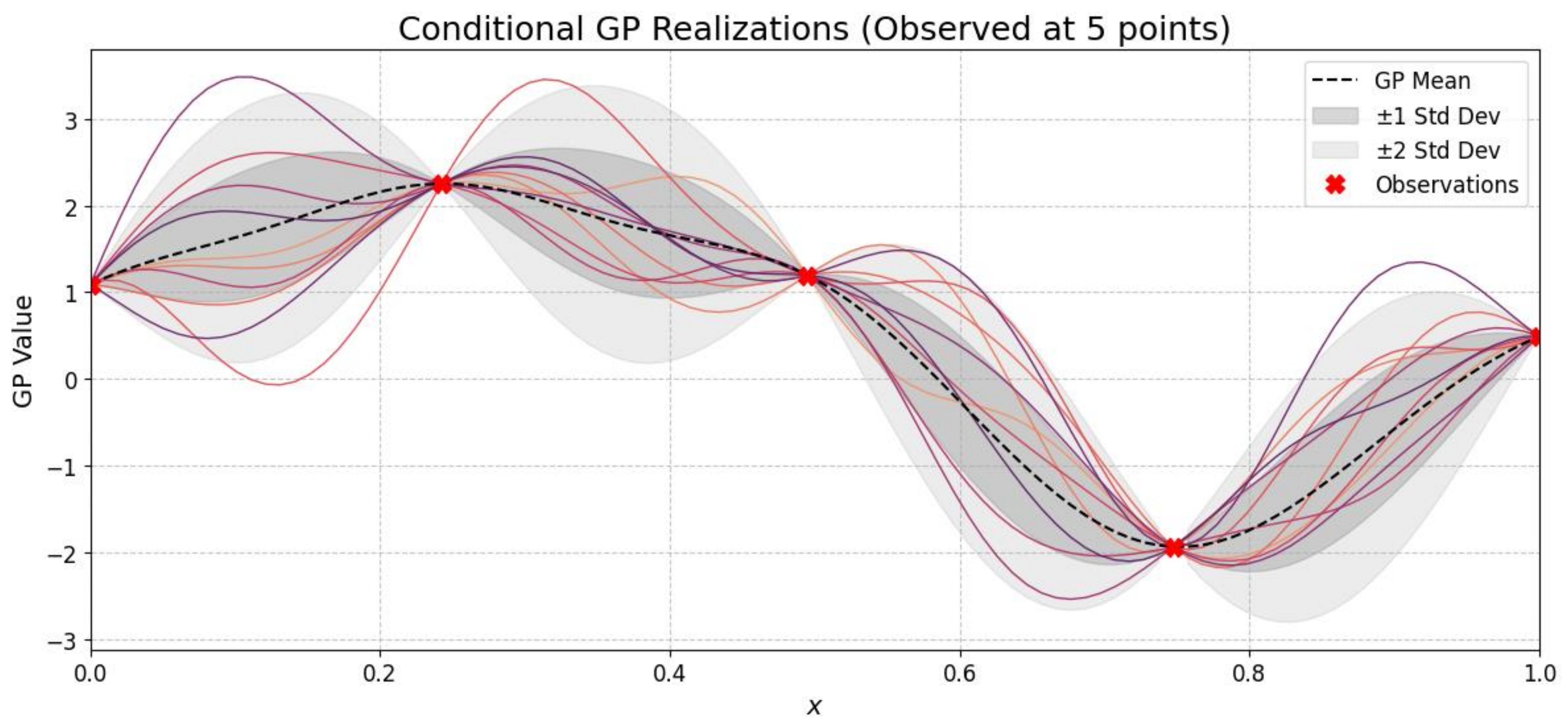
Updates belief on all remaining $y(x)$:

$$y(x) \sim N(\mu(x; X, Y), \Sigma(X, Y))$$

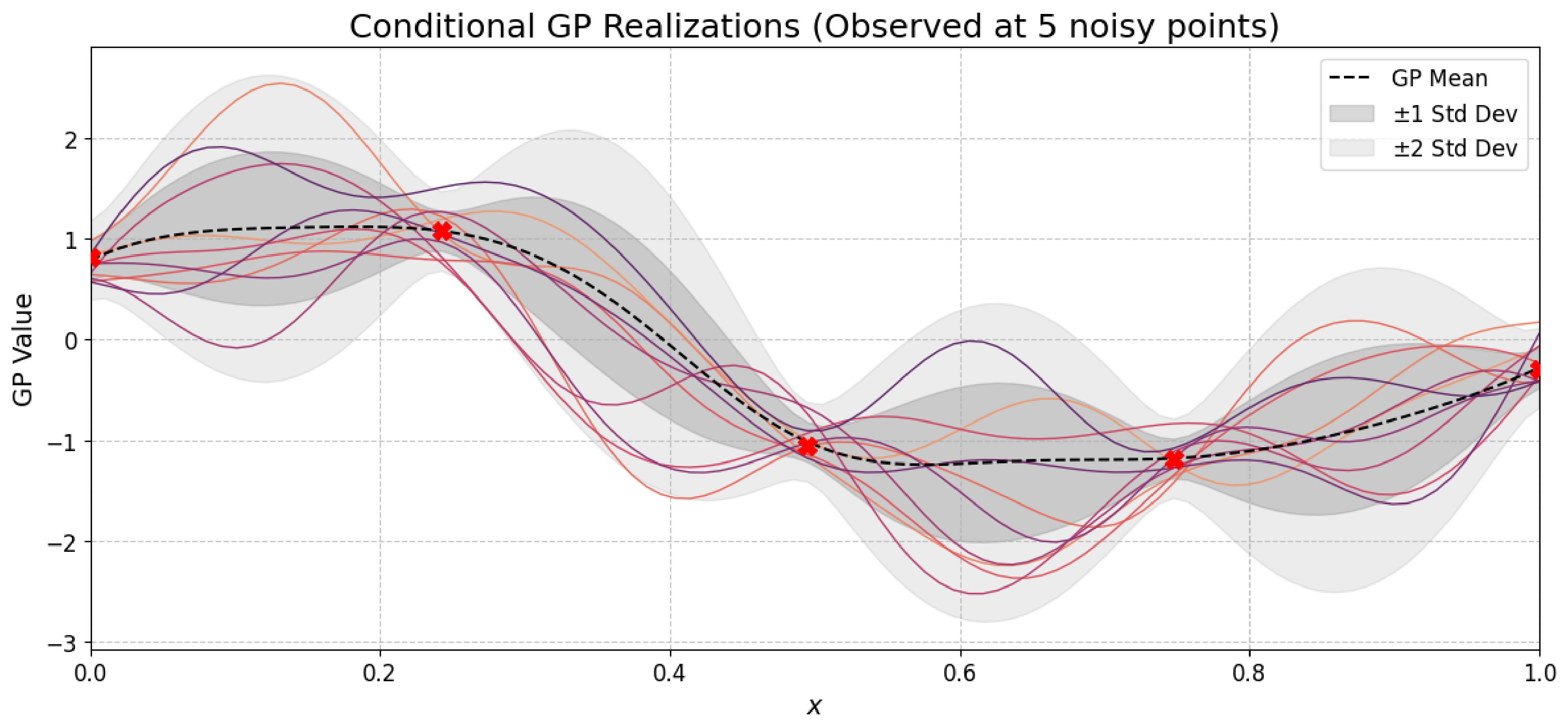
$$\mu(x; X, Y) = \mu(x) + k(x, X)[k(X, X)]^{-1}(Y - \mu(X))$$

$$\Sigma(X, Y) = k(x, x) - k(x, X)[k(X, X)]^{-1}k(X, x)$$

More data helps pin down the function



Gaussian process regression handles uncertainty in observed data in a natural way



Gaussian process regression: Bayesian interpretation

Bayes' rule: $p(y(x)|X,Y) \propto p(y(x)) \cdot p(Y|X,y(x))$

Prior: $p(y(x)) = N(\mu(x), k(x,x))$

Likelihood: $p(Y|X,y(x)) = N(\mu(X), k(X,X))$

Bayesian posterior: $y(x) \sim N(\mu(x;X,Y), \Sigma(X,Y))$

$$\mu(x;X,Y) = \mu(x) + k(x,X)[k(X,X)]^{-1}(Y - \mu(X))$$

$$\Sigma(X,Y) = k(x,x) - k(x,X)[k(X,X)]^{-1}k(X,x)$$

Gaussian process regression: Bayesian interpretation handles noise

Bayes’ rule: $p(y(x)|X,Y) \propto p(y(x)) \cdot p(Y|X,y(x))$

Prior: $p(y(x)) = N(\mu(x), k(x,x))$

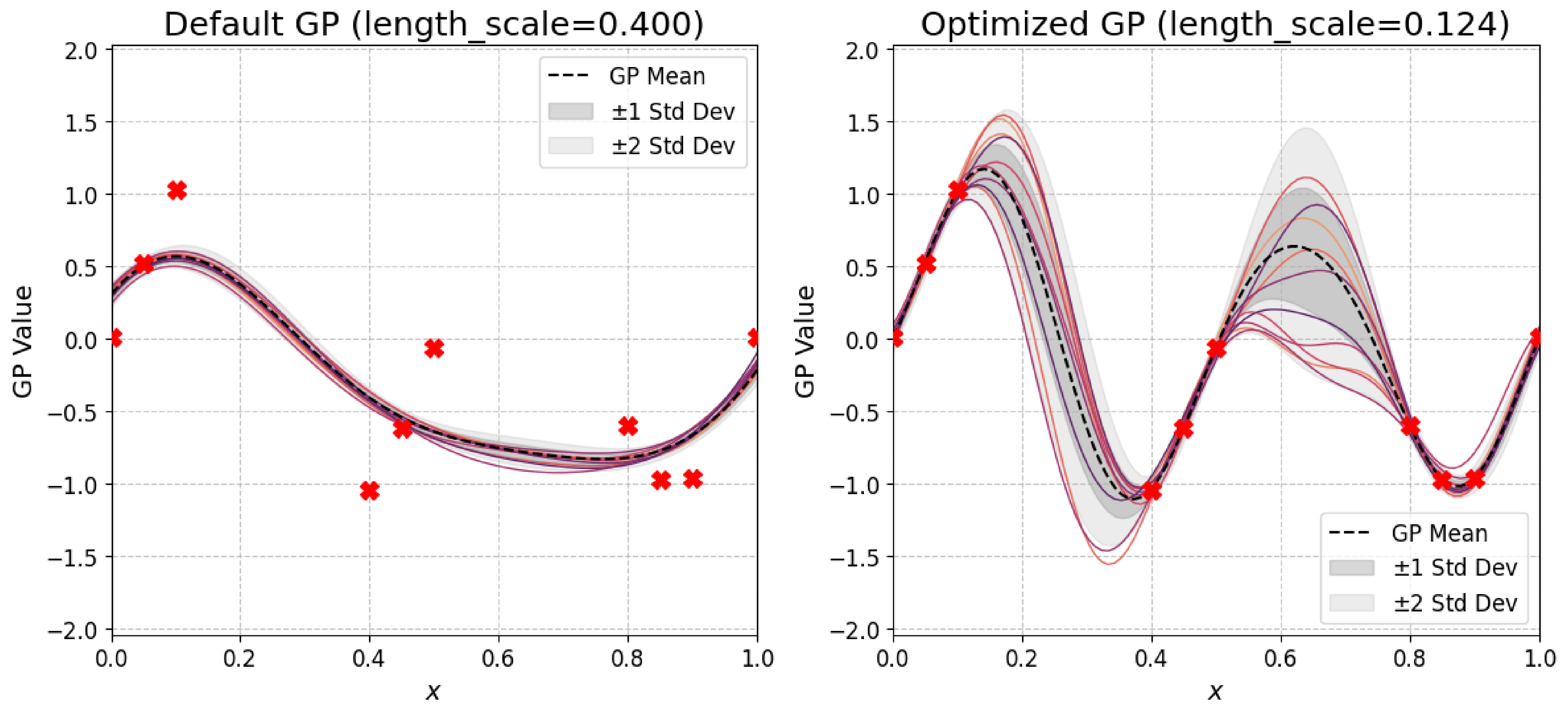
Noisy likelihood: $p(Y|X,y(x)) = N(\mu(X), k(X,X) + \boxed{\Sigma_Y})$

Bayesian posterior: $y(x) \sim N(\mu(x;X,Y), \Sigma(X,Y))$

$$\mu(x;X,Y) = \mu(x) + k(x,X)[k(X,X) + \boxed{\Sigma_Y}]^{-1}(Y - \mu(X))$$

$$\Sigma(X,Y) = k(x,x) - k(x,X)[k(X,X) + \boxed{\Sigma_Y}]^{-1}k(X,x)$$

Hyperparameters



Computational costs of Gaussian Process Regression

Posterior predictive: kernel matrix inversion is **dominant cost $O(n^3)$**

$$\mu(x; X, Y) = \mu(x) + k(x, X)[k(X, X) + \Sigma_Y]^{-1}(Y - \mu(X))$$

$$\Sigma(X, Y) = k(x, x) - k(x, X)[k(X, X) + \Sigma_Y]^{-1}k(X, x)$$

Hyperparameter optimization requires repeated kernel solves

Additionally, there is a **data acquisition cost** to acquire (x_i, y_i) for $i = 1 \dots n$

- available high-fidelity data may be limited by finite cost budgets
- when data are limited, GPR stays close to the prior

Motivates **multifidelity Gaussian process modeling** approaches

Multifidelity Gaussian Process Modeling

Single-fidelity GPR: $\{(x_i, y_i)\}_{i=1}^{n}$ all from one source (model/exp)

Multifidelity GP: how do we leverage data from different sources?

- $\left\{\left(x_i, y_i^{(1)}\right)\right\}_{i=1}^{n_1}$
- $\left\{\left(x_i, y_i^{(2)}\right)\right\}_{i=1}^{n_2}$

$\vdots$

- $\left\{\left(x_i, y_i^{(K)}\right)\right\}_{i=1}^{n_K}$

Two general paradigms:

1. Cokriging – all data go into one mega-GP
2. Autoregressive – sequentially build up from lowest to highest fidelity

Cokriging

Main idea: Treat high- and low-fidelity data as jointly Gaussian

$$\begin{bmatrix} Y^{(1)} \\ Y^{(2)} \\ \vdots \\ Y^{(K)} \end{bmatrix} \sim N\left(\begin{bmatrix} \mu^{(1)}(X^{(1)}) \\ \mu^{(2)}(X^{(2)}) \\ \vdots \\ \mu^{(K)}(X^{(K)}) \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1K} \\ \Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{K1} & \Sigma_{K2} & \cdots & \Sigma_{KK} \end{bmatrix} \right)$$

Pros: each $\Sigma_{ij} = k_{ij}\left(X^{(i)}, X^{(j)}\right)$ is a block matrix describing correlation between fidelities i and j, allowing information transfer across all fidelities

Cons: flexibility requires many hyperparameters, and prediction requires inverting giant kernel matrix

Autoregressive multifidelity GPs

Main idea: train GP on lowest-fidelity data, then recursively train GPs on each level of fidelity using the fidelity immediately below it

$$\boxed{f_K} \rightarrow \boxed{f_{K-1}} \dashrightarrow \boxed{f_2} \rightarrow \boxed{f_1}$$

Kennedy-O'Hagan (linear autoregressive): $f_\ell(x) = \rho f_{\ell+1}(x) + \delta_\ell(x)$

- where $\delta_\ell(x)$ is a GP trained on the discrepancy between the models

Nonlinear autoregressive GPs: train nonlinear functions between models

Pros: easier to train fidelity-specific corrections than one large joint model

Cons: "Markovian" information transfer between fidelities

MAGPI: Multifidelity-Augmented Gaussian Process Inputs

Main idea: Combine features of cokriging and autoregressive approaches

Cokriging exploits correlations across data of all fidelities:

MAGPI exploits correlations across *predictions* of all fidelities

Autoregressive approaches enable training at one fidelity level at a time:

MAGPI trains low-fidelity predictions one level at a time

MAGPI

Cokriging is based on a joint distribution of all the data:

$$\begin{bmatrix} Y^{(1)} \\ Y^{(2)} \\ \vdots \\ Y^{(K)} \end{bmatrix} \sim N\left(\begin{bmatrix} \mu^{(1)}(X^{(1)}) \\ \mu^{(2)}(X^{(2)}) \\ \vdots \\ \mu^{(K)}(X^{(K)}) \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1K} \\ \Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{K1} & \Sigma_{K2} & \cdots & \Sigma_{KK} \end{bmatrix} \right)$$

MAGPI is based on a joint distribution of the high-fidelity data and *all* the low-fidelity predictions

$$\begin{bmatrix} Y^{(1)} \\ y^{(2)}(x) \\ \vdots \\ y^{(K)(x)} \end{bmatrix} \sim N\left(\begin{bmatrix} \mu^{(1)}(X^{(1)}) \\ \mu^{(2)}(x) \\ \vdots \\ \mu^{(K)}(x) \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma'_{12} & \cdots & \Sigma'_{1K} \\ \Sigma'_{21} & \Sigma'_{22} & \cdots & \Sigma'_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma'_{K1} & \Sigma'_{K2} & \cdots & \Sigma'_{KK} \end{bmatrix} \right)$$

MAGPI: recursive training approach

Training approach:

1. Train $\hat{f}_K(x)$ on lowest-fidelity data $X^{(K)}, Y^{(K)}$
2. For $\ell = K-1, K-2, \dots, 2$:
   - Train $\hat{f}_\ell$ on ℓ-th fidelity data $X^{(\ell)}, Y^{(\ell)}$ and predictions from $\hat{f}_{\ell+1}, \dots, \hat{f}_K$
3. Train MAGPI GP on $X^{(1)}, Y^{(1)}$ and predictions from $\hat{f}_2, \dots, \hat{f}_K$

Models at lower-fidelity levels need not be GPs!

- Can avoid challenges of inverting huge kernels when low-fidelity data are plentiful

MAGPI: recursive prediction approach

Prediction approach:

1. Predict at new inputs using lowest-fidelity trained model $\hat{f}_K(x)$
2. For $\ell = K-1, K-2, \ldots, 1$: predict $\hat{f}_\ell\left(x, \hat{f}_{\ell+1}, \ldots, \hat{f}_K\right)$

Theoretically justified: augmented features can achieve marginal likelihoods at least as high as non-augmented features under mild assumptions

A toy example

High-fidelity:

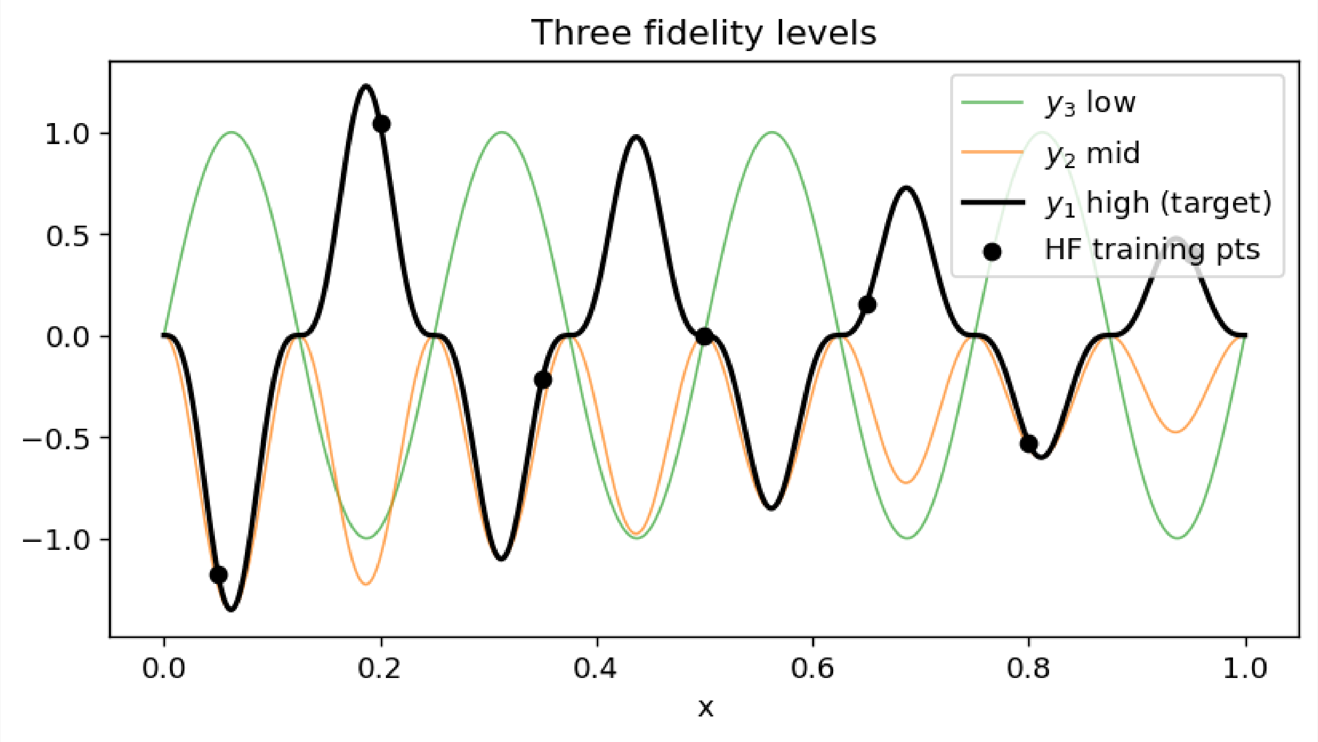
$$y = \sin 8\pi x$$

Middle-fidelity:

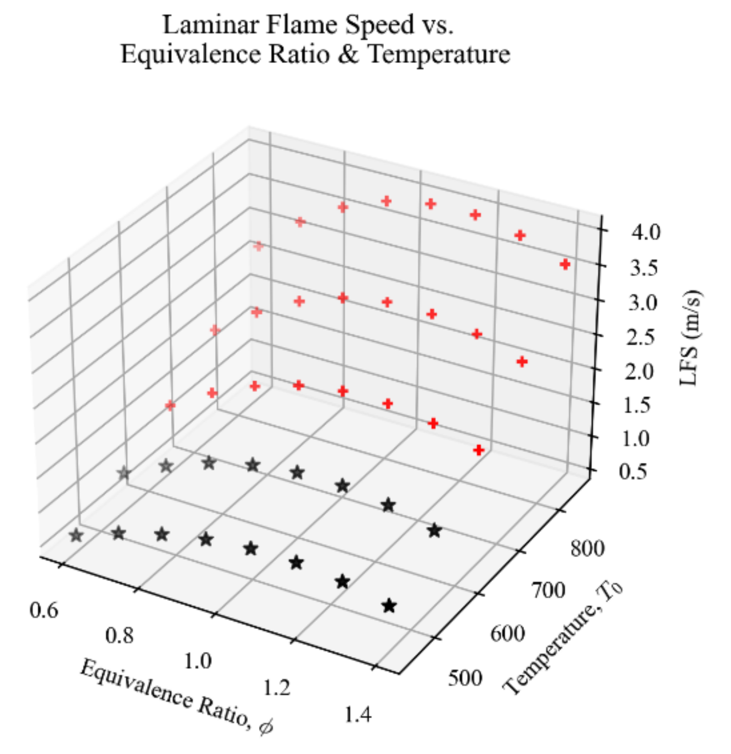
$$y = \left(x - \sqrt{2}\right) \sin^2 8\pi x$$

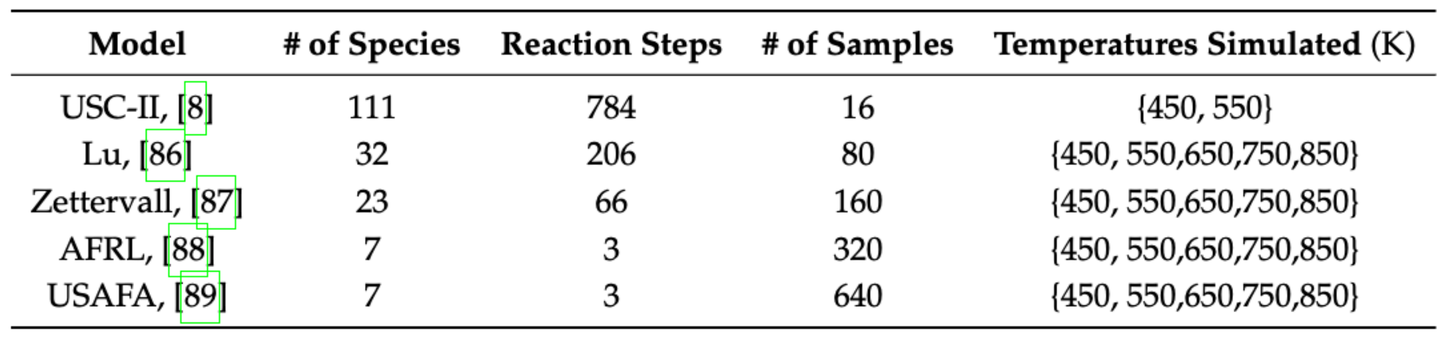
Low-fidelity:

$$y = \left(x - \sqrt{2}\right) \sin^3 8\pi x$$



Applications: Flame speed prediction





Model	# of Species	Reaction Steps	# of Samples	Temperatures Simulated (K)
USC-II, [8]	111	784	16	{450, 550}
Lu, [86]	32	206	80	{450, 550,650,750,850}
Zettervall, [87]	23	66	160	{450, 550,650,750,850}
AFRL, [88]	7	3	320	{450, 550,650,750,850}
USAFA, [89]	7	3	640	{450, 550,650,750,850}

Train on the **black** high-fidelity data points from USC-II model

Test on the **red** high-fidelity data points from USC-II model (unseen during training)

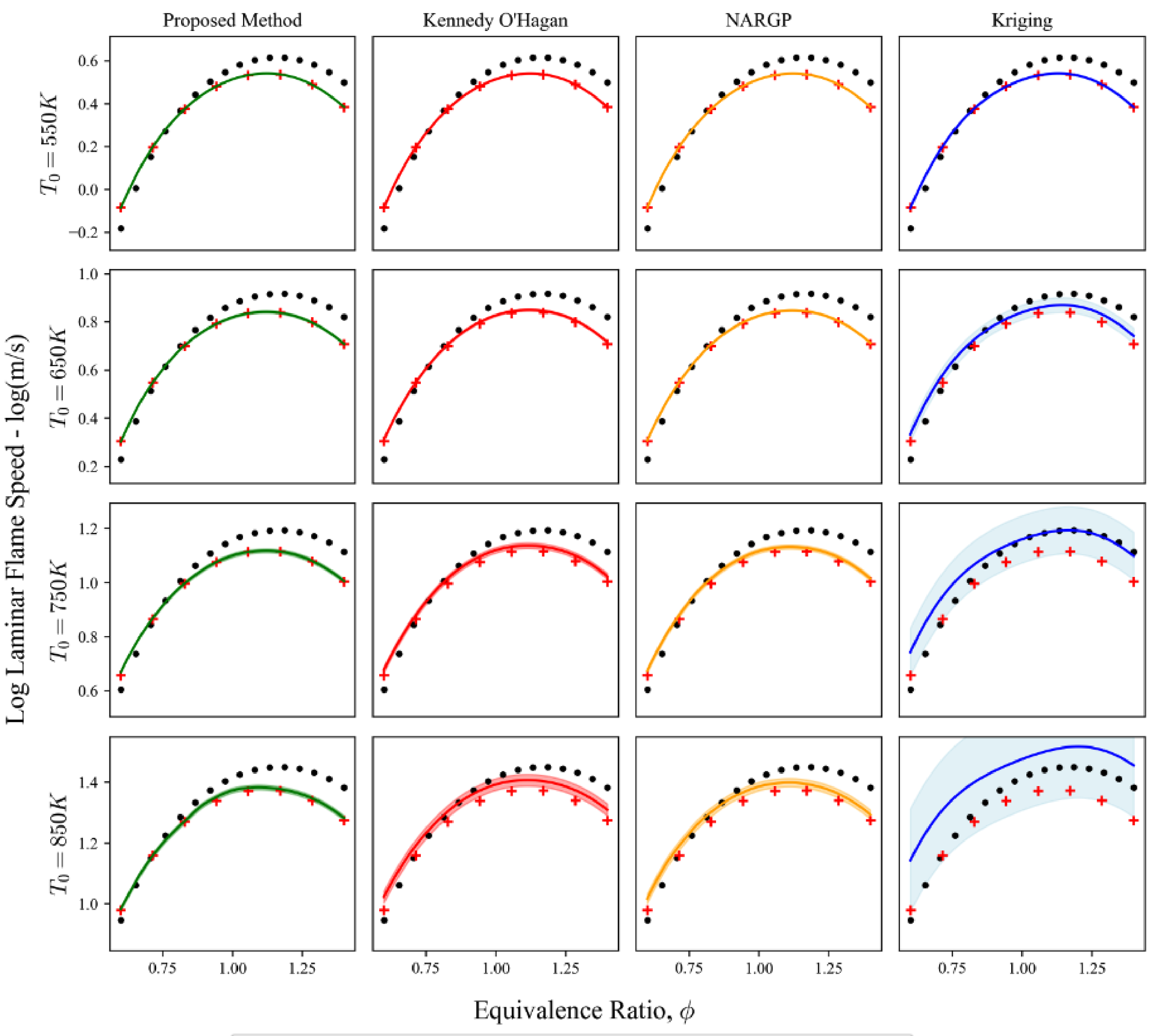
Flame speed prediction: results

$T_0 = 550$ (top row) is within high-fidelity training range

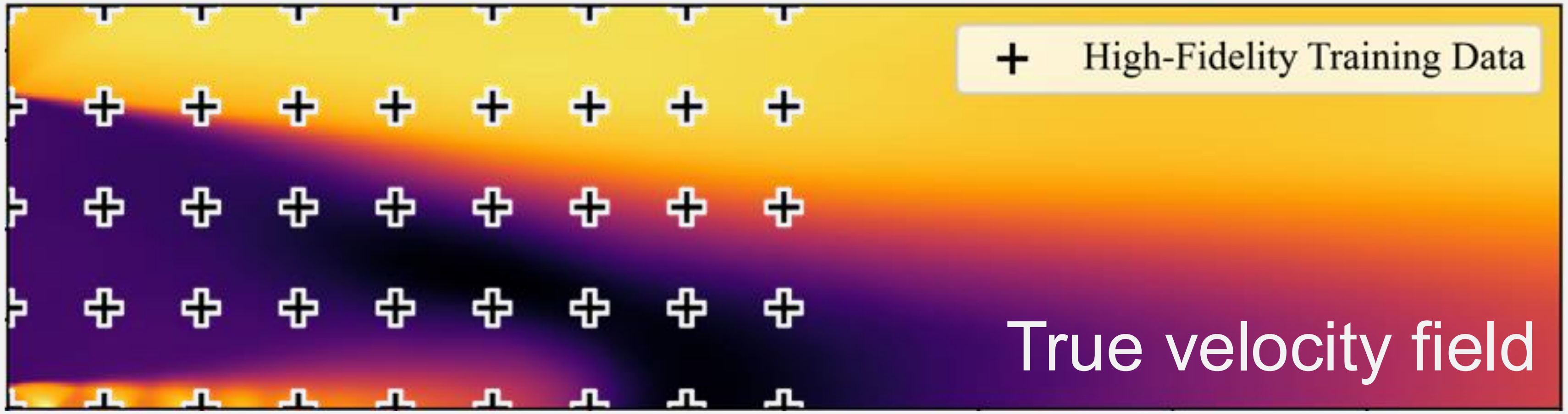
Higher temperatures (lower rows) not seen in high-fidelity data

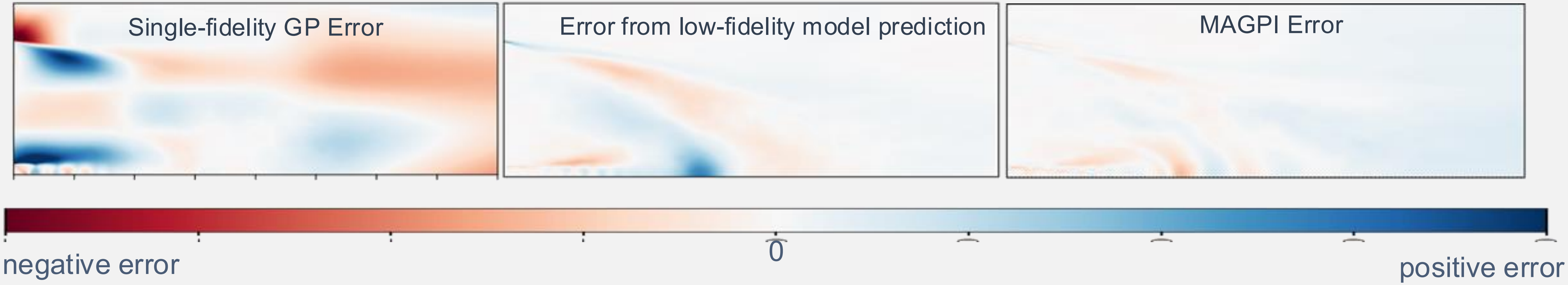
Red +'s are true high-fidelity data

Black ●'s are fidelity-2 data (biased)



Applications: Flow field reconstruction from sparse measurement probes





Thank you!

Atticus Rex, Elizabeth Qian, & David Peterson (2026).
MAGPI: Multifidelity-augmented Gaussian process inputs for surrogate modeling from scarce data.
https://arxiv.org/pdf/2603.22050

Support from the National Science Foundation award number CMMI-2442140 is gratefully acknowledged

Questions?

eqian@gatech.edu
https://www.elizabethqian.com

MAGPI work led by GT PhD student Atticus Rex

https://atticusdrex.github.io/website/