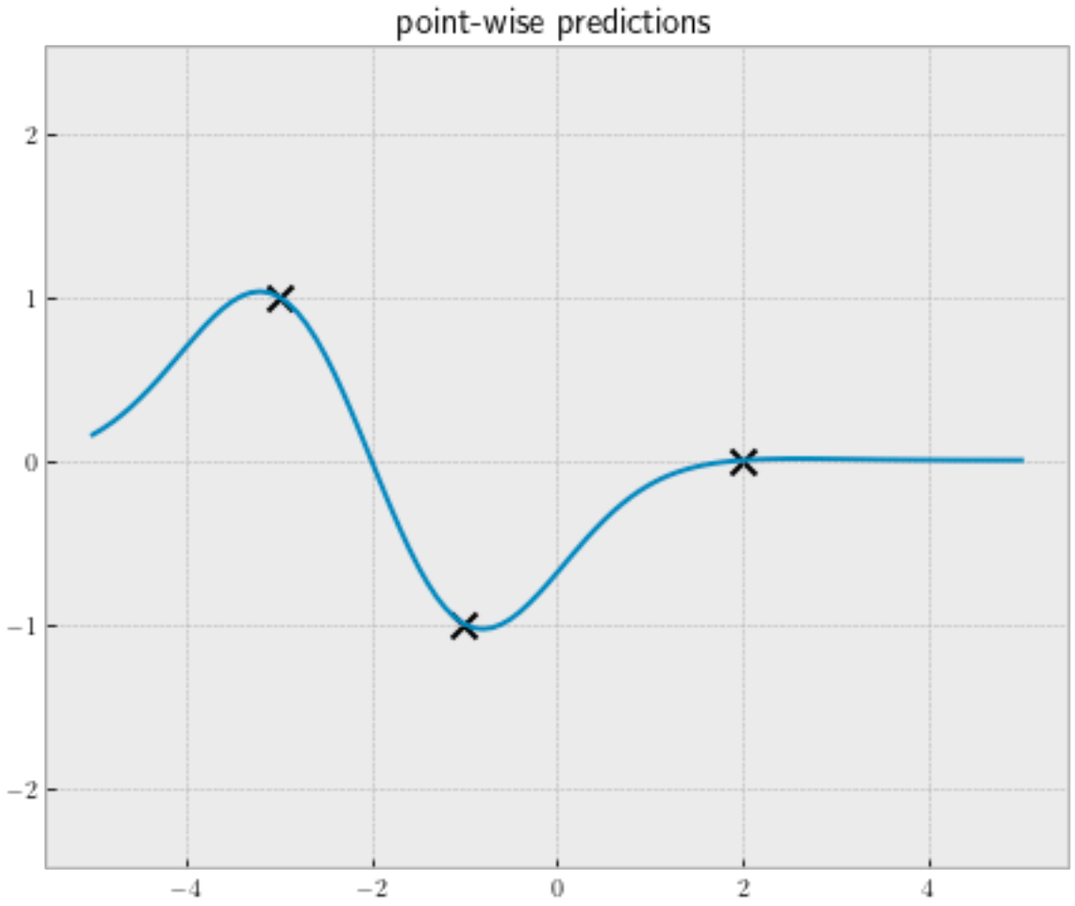
CSE 517 — MACHINE LEARNING

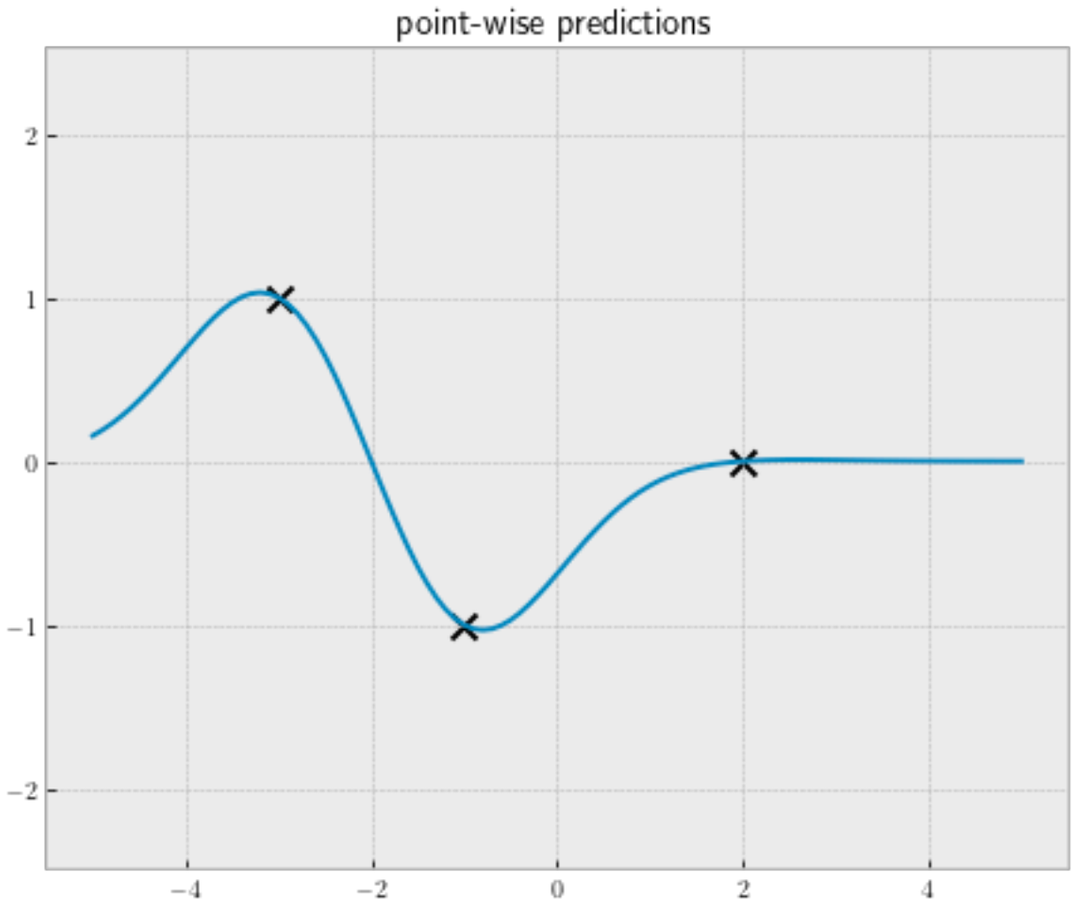
QUAN NGUYEN

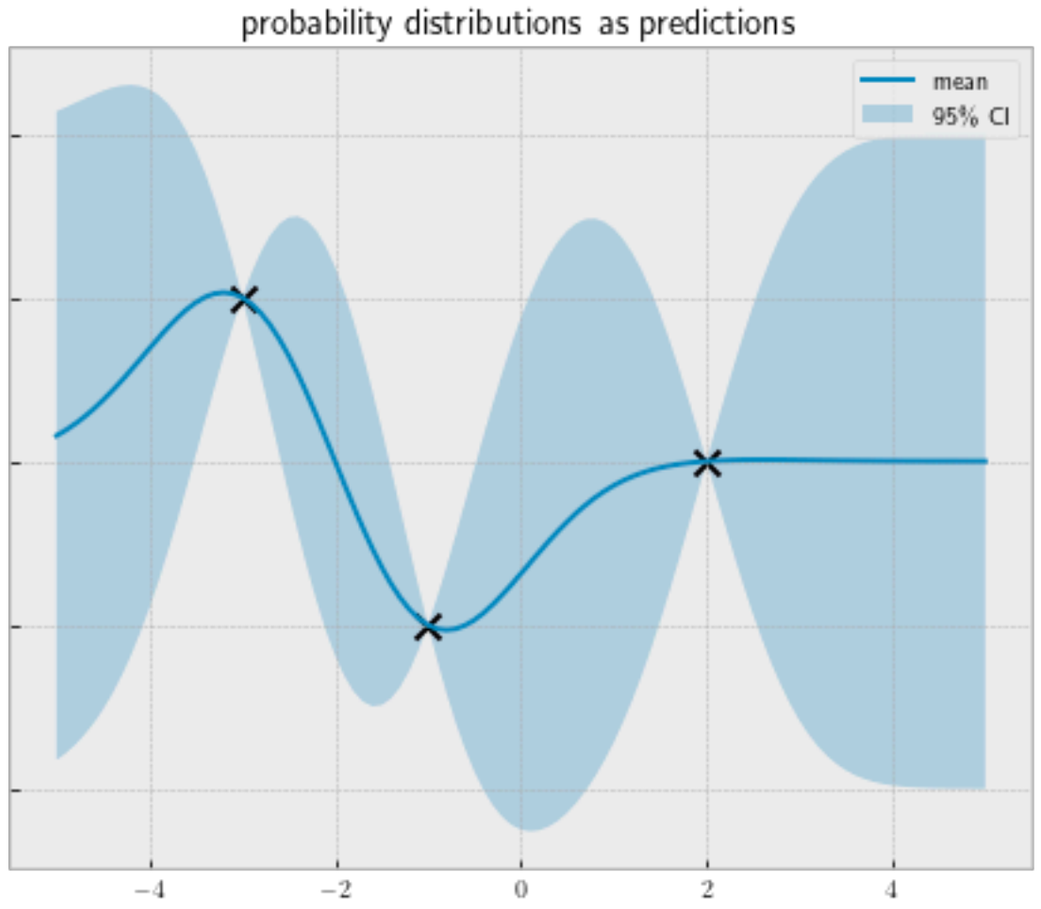
GAUSSIAN PROCESSES

POINT-ESTIMATES VS. PREDICTIVE DISTRIBUTIONS



POINT-ESTIMATES VS. PREDICTIVE DISTRIBUTIONS





BEING GAUSSIAN IS VERY CONVENIENT...

BEING GAUSSIAN IS VERY CONVENIENT…

Consider multivariate normal random variable $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ with

mean $\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ and covariance matrix $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$

BEING GAUSSIAN IS VERY CONVENIENT…

Consider multivariate normal random variable $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ with

mean $\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ and covariance matrix $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$

$$p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

$$p(y_1, y_2) = p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

$$p(y_1, y_2) = p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

$$p(y_1, y_2) = p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

Marginalization: $p(y_1) = \int_{y_2} p(y_1, y_2)\ \mathrm{d}y_2 = N(\mu_1, \Sigma_{11})$

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

$$p(y_1, y_2) = p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

Marginalization: $p(y_1) = \int_{y_2} p(y_1, y_2)\ \mathrm{d}y_2 = N(\mu_1, \Sigma_{11})$

- When we don't care about the nuisance parameter y_2

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

$$p(y_1, y_2) = p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

$$p(y_1, y_2) = p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

Summation: $p(y_1) = N(\mu_1, \Sigma_{11})$

$$p(y_2) = N(\mu_2, \Sigma_{22})$$

$$p(y_1 + y_2) = N(\mu_1 + \mu_2, \Sigma_{11} + \Sigma_{22})$$

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

$$p(y_1, y_2) = p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

$$p(y_1, y_2) = p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

Conditioning: $p(y_1 \mid y_2) = N(\mu_1', \Sigma_{11}')$

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

$$p(y_1, y_2) = p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

Conditioning: $p(y_1 \mid y_2) = N(\mu_1', \Sigma_{11}')$

$$\mu_1' = \mu_1 + \Sigma_{12}\,\Sigma_{22}^{-1}\,(y_2 - \mu_2)$$

$$\Sigma_{11}' = \Sigma_{11} - \Sigma_{12}\,\Sigma_{22}^{-1}\,\Sigma_{21}$$

. . . BECAUSE GAUSSIANITY IS PRESERVED IN MANY OPERATIONS

$$p(y_1, y_2) = p\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

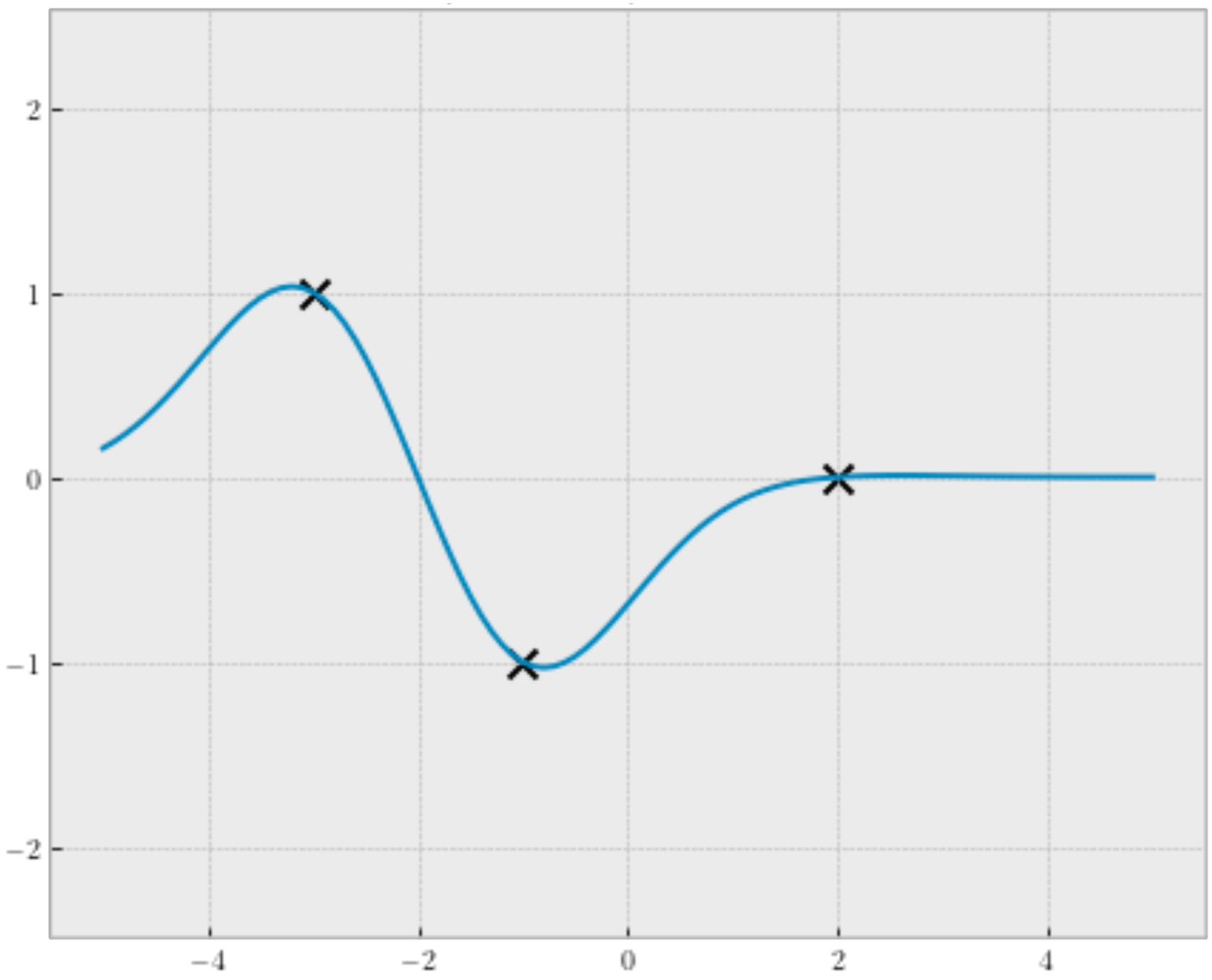
Conditioning: $p(y_1 \mid y_2) = N(\mu_1', \Sigma_{11}')$

$$\mu_1' = \mu_1 + \Sigma_{12}\,\Sigma_{22}^{-1}\,(y_2 - \mu_2)$$

$$\Sigma_{11}' = \Sigma_{11} - \Sigma_{12}\,\Sigma_{22}^{-1}\,\Sigma_{21}$$

Applies to **any** multivariate Gaussian

CONSIDER A REGRESSION TASK



THE (KERNEL) REGRESSION MODEL

THE (KERNEL) REGRESSION MODEL

$$y = f(\mathbf{x}) + \varepsilon = \mathbf{w}^\top \mathbf{x} + \varepsilon, \text{ where } \varepsilon \sim N(0, \sigma_n^2)$$

THE (KERNEL) REGRESSION MODEL

$$y = f(\mathsf{x}) + \varepsilon = \mathsf{w}^\top \mathsf{x} + \varepsilon, \text{ where } \varepsilon \sim N(0, \sigma_n^2)$$

Learning w: $\mathsf{w}^\top \mathsf{x}_* = K(\mathsf{x}_*, X)\ K^{-1}(X, X)\ \mathsf{y}$

THE (KERNEL) REGRESSION MODEL

$$y = f(\mathbf{x}) + \varepsilon = \mathbf{w}^\top \mathbf{x} + \varepsilon, \text{ where } \varepsilon \sim N(0, \sigma_n^2)$$

Learning w: $\mathbf{w}^\top \mathbf{x}_* = K(\mathbf{x}_*, X)\ K^{-1}(X, X)\ \mathbf{y}$

What is the **prediction** for label y_* of test point $\mathbf{x}_*$?

THE (KERNEL) REGRESSION MODEL

$$y = f(\mathsf{x}) + \varepsilon = \mathsf{w}^\top \mathsf{x} + \varepsilon, \text{ where } \varepsilon \sim N(0, \sigma_n^2)$$

Learning w: $\mathsf{w}^\top \mathsf{x}_* = K(\mathsf{x}_*, X)\ K^{-1}(X, X)\ \mathsf{y}$

What is the **prediction** for label y_* of test point x_*?

$$p(y_* \mid \mathsf{x}_*, \mathsf{w}) = N(\mathsf{w}^\top \mathsf{x}_*, \sigma_n^2) = N\left(K(\mathsf{x}_*, X)\ K^{-1}(X, X)\ \mathsf{y}, \sigma_n^2\right)$$

THE (KERNEL) REGRESSION MODEL

$$y = f(\mathsf{x}) + \varepsilon = \mathsf{w}^\top \mathsf{x} + \varepsilon, \text{ where } \varepsilon \sim N(0, \sigma_n^2)$$

Learning w: $\mathsf{w}^\top \mathsf{x}_* = K(\mathsf{x}_*, X)\ K^{-1}(X, X)\ \mathsf{y}$

What is the **prediction** for label y_* of test point x_*?

$$p(y_* \mid \mathsf{x}_*, \mathsf{w}) = N(\mathsf{w}^\top \mathsf{x}_*, \sigma_n^2) = N\left(K(\mathsf{x}_*, X)\ K^{-1}(X, X)\ \mathsf{y}, \sigma_n^2\right)$$

Since label y_* is what we really care about, weight vector w is a **nuisance parameter**.

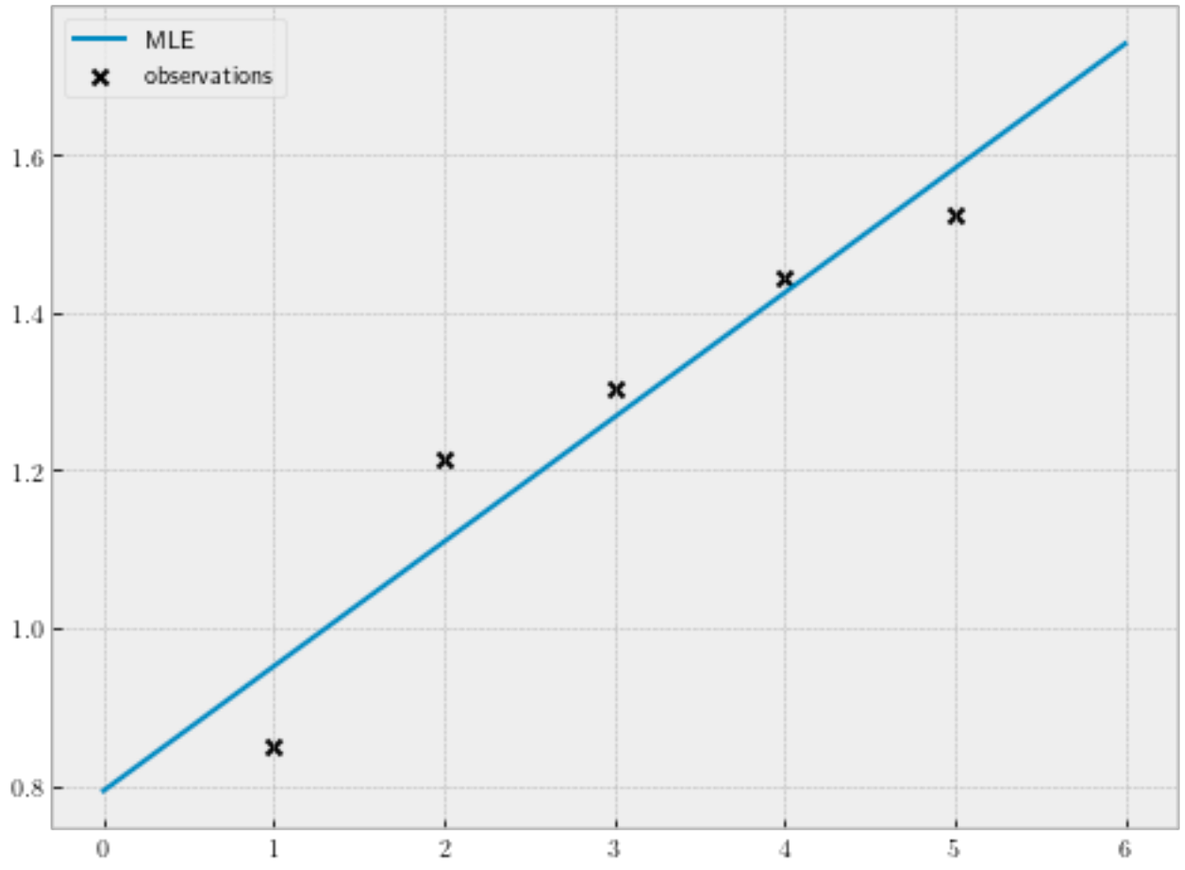
MARGINALIZING OVER WEIGHT VECTORS

MARGINALIZING OVER WEIGHT VECTORS

$$p(y \mid \mathsf{x}, D) = \int_w p(y, \mathsf{w} \mid \mathsf{x}, D)\ \mathrm{d}\mathsf{w} = \int_w p(y \mid \mathsf{x}, \mathsf{w})\ p(\mathsf{w} \mid D)\ \mathrm{d}\mathsf{w}$$

MARGINALIZING OVER WEIGHT VECTORS

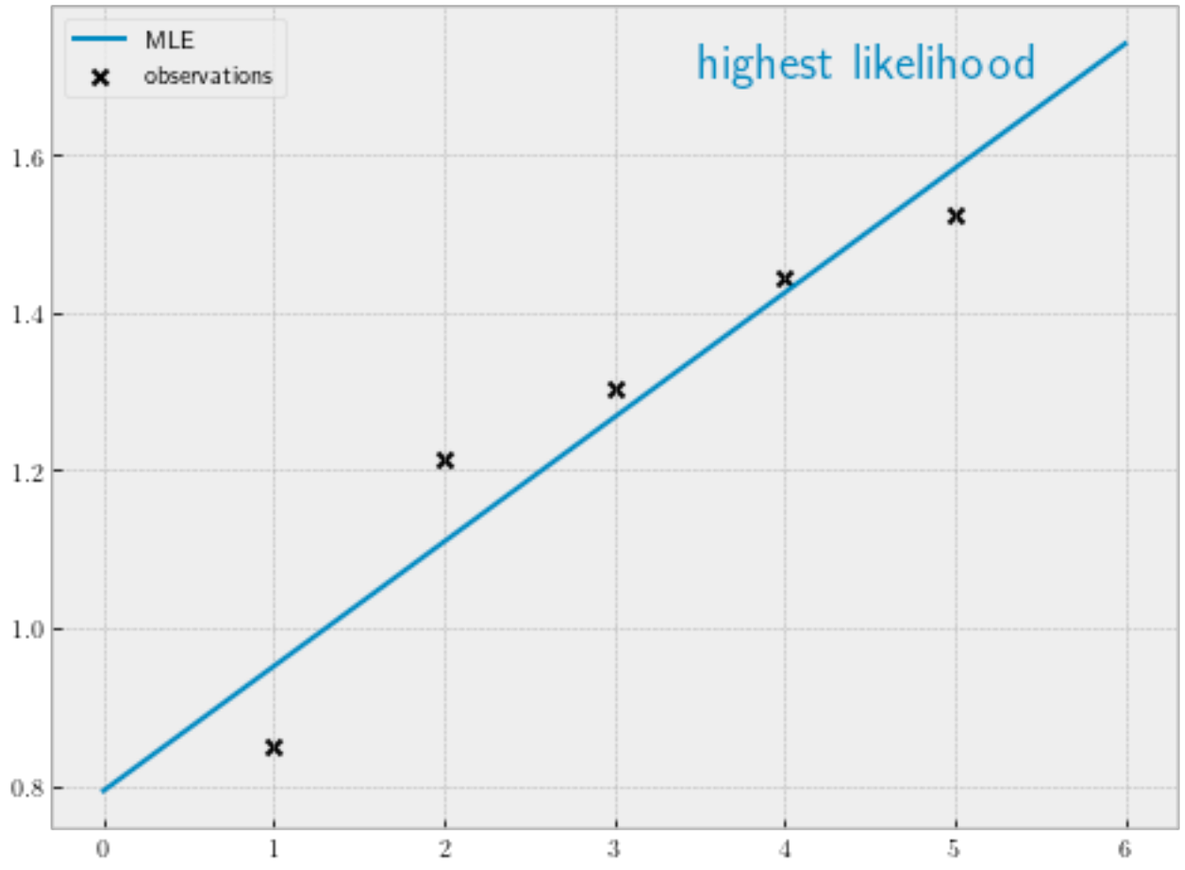
$$p(y \mid \mathrm{x}, D) = \int_w p(y, \mathrm{w} \mid \mathrm{x}, D)\ \mathrm{dw} = \int_w p(y \mid \mathrm{x}, \mathrm{w})\ p(\mathrm{w} \mid D)\ \mathrm{dw}$$



MARGINALIZING OVER WEIGHT VECTORS

$$p(y \mid \mathrm{x}, D) = \int_w p(y, \mathrm{w} \mid \mathrm{x}, D)\ \mathrm{dw} = \int_w p(y \mid \mathrm{x}, \mathrm{w})\ p(\mathrm{w} \mid D)\ \mathrm{dw}$$

MLE: max $p(D \mid \mathrm{w})$



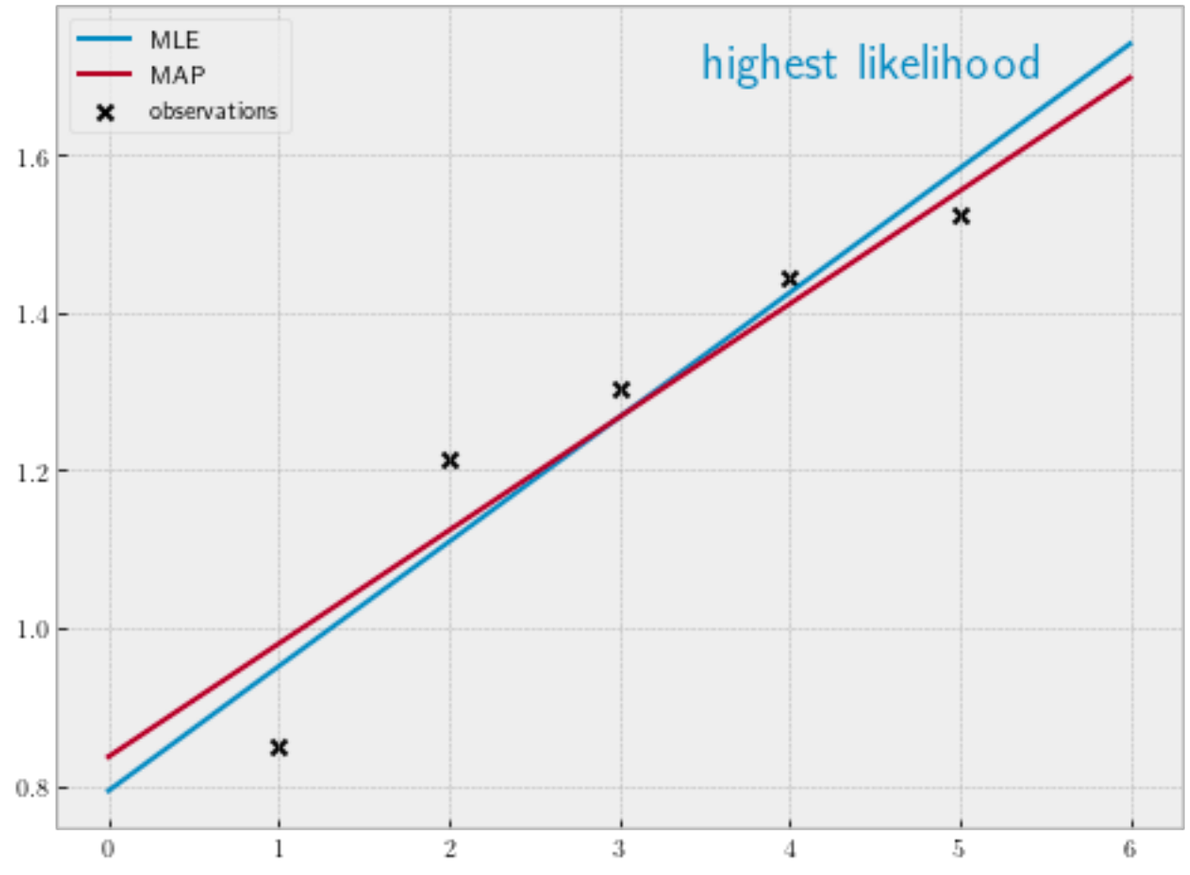
MARGINALIZING OVER WEIGHT VECTORS

$$p(y \mid \mathrm{x}, D) = \int_w p(y, \mathrm{w} \mid \mathrm{x}, D)\ \mathrm{dw} = \int_w p(y \mid \mathrm{x}, \mathrm{w})\ p(\mathrm{w} \mid D)\ \mathrm{dw}$$

MLE: max $p(D \mid \mathrm{w})$

MAP: max $p(\mathrm{w} \mid D)$

$$\propto p(D \mid \mathrm{w})\ p(\mathrm{w})$$



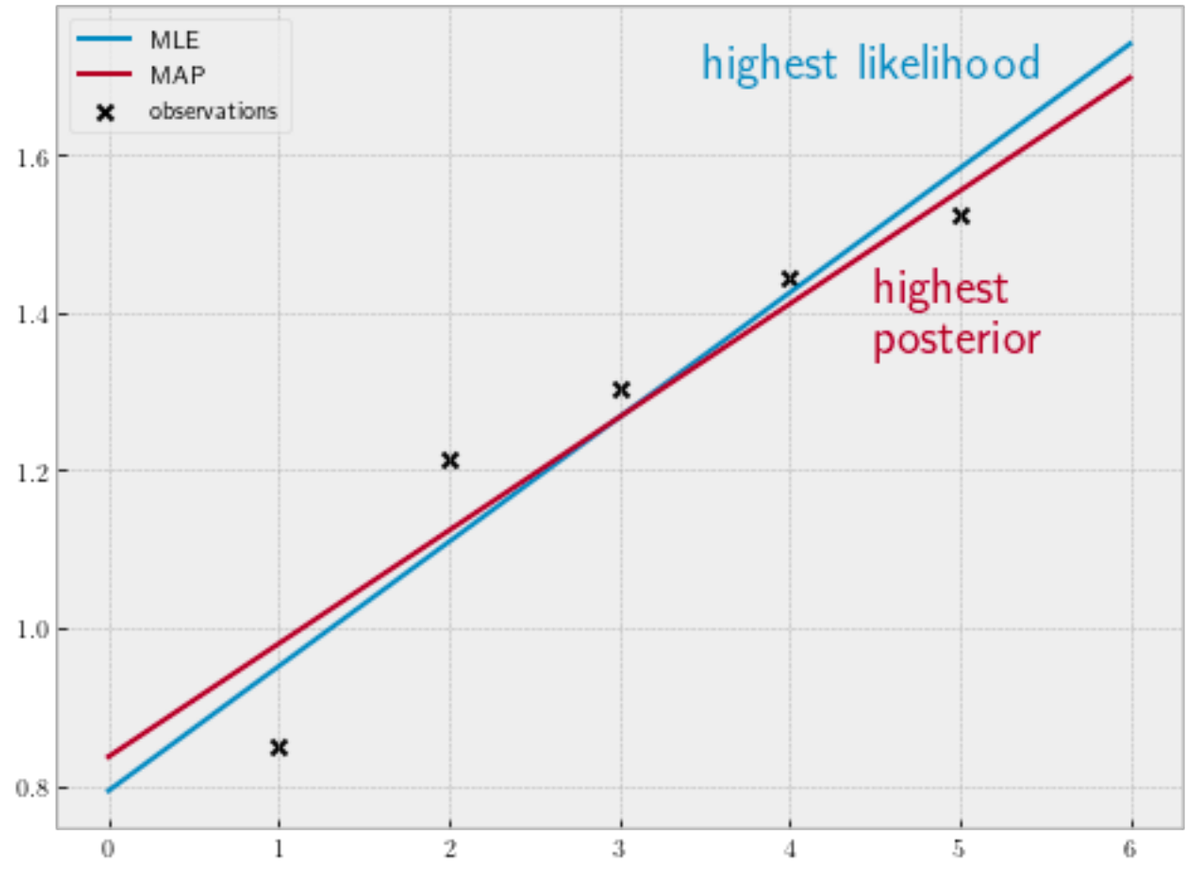
MARGINALIZING OVER WEIGHT VECTORS

$$p(y \mid \mathrm{x}, D) = \int_w p(y, \mathrm{w} \mid \mathrm{x}, D)\ \mathrm{dw} = \int_w p(y \mid \mathrm{x}, \mathrm{w})\ p(\mathrm{w} \mid D)\ \mathrm{dw}$$

MLE: max $p(D \mid \mathrm{w})$

MAP: max $p(\mathrm{w} \mid D)$

$$\propto p(D \mid \mathrm{w})\ p(\mathrm{w})$$



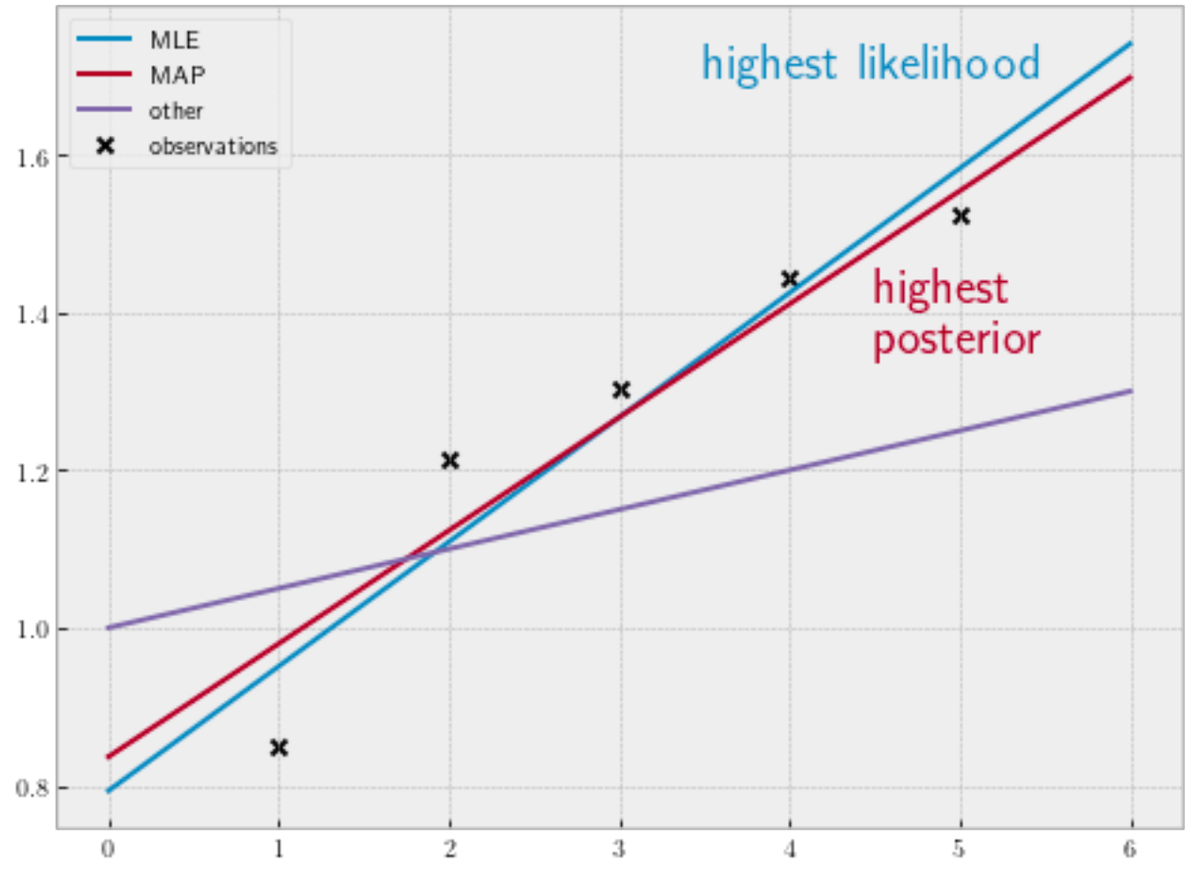
MARGINALIZING OVER WEIGHT VECTORS

$$p(y \mid \mathrm{x}, D) = \int_w p(y, \mathrm{w} \mid \mathrm{x}, D)\ \mathrm{dw} = \int_w p(y \mid \mathrm{x}, \mathrm{w})\ p(\mathrm{w} \mid D)\ \mathrm{dw}$$

MLE: max $p(D \mid \mathrm{w})$

MAP: max $p(\mathrm{w} \mid D)$

$$\propto p(D \mid \mathrm{w})\ p(\mathrm{w})$$



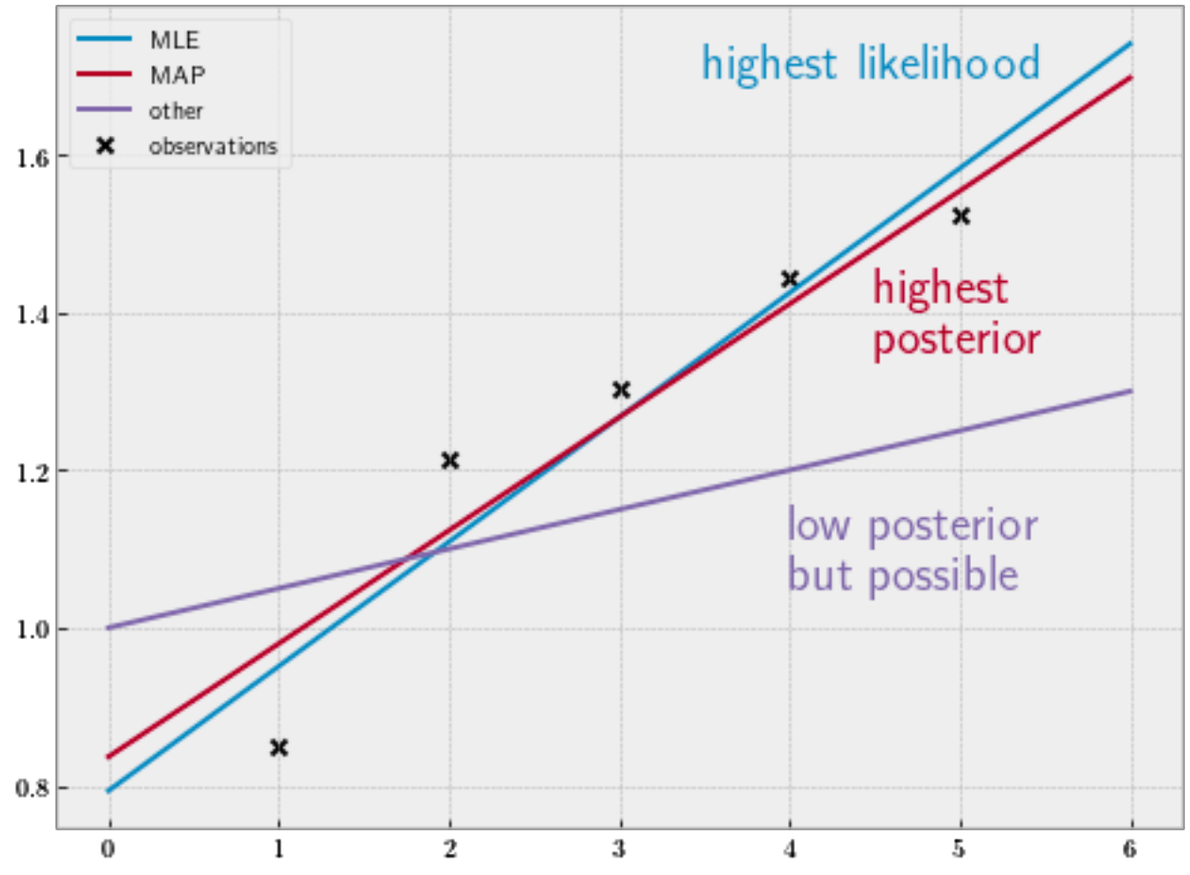
MARGINALIZING OVER WEIGHT VECTORS

$$p(y \mid \mathrm{x}, D) = \int_w p(y, \mathrm{w} \mid \mathrm{x}, D)\ \mathrm{dw} = \int_w p(y \mid \mathrm{x}, \mathrm{w})\ p(\mathrm{w} \mid D)\ \mathrm{dw}$$

MLE: max $p(D \mid \mathrm{w})$

MAP: max $p(\mathrm{w} \mid D)$

$$\propto p(D \mid \mathrm{w})\ p(\mathrm{w})$$



MARGINALIZING OVER WEIGHT VECTORS

$$p(y \mid \mathrm{x}, D) = \int_w p(y, \mathrm{w} \mid \mathrm{x}, D)\ \mathrm{dw} = \int_w p(y \mid \mathrm{x}, \mathrm{w})\ p(\mathrm{w} \mid D)\ \mathrm{dw}$$

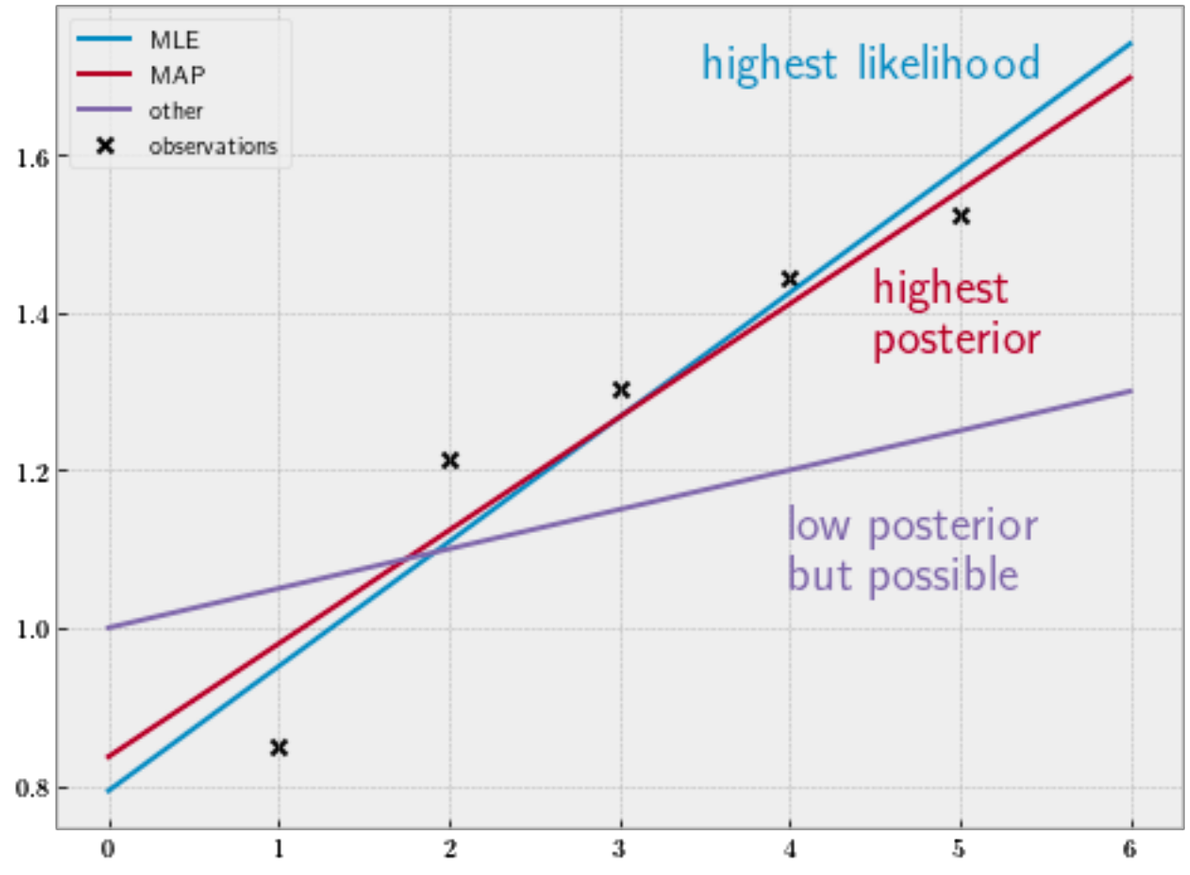
MLE: max $p(D \mid \mathrm{w})$

MAP: max $p(\mathrm{w} \mid D)$

$\propto p(D \mid \mathrm{w})\ p(\mathrm{w})$

Others: low $p(D \mid \mathrm{w})$

or low $p(\mathrm{w} \mid D)$



SIDE-STEPPING THE WEIGHT VECTORS

SIDE-STEPPING THE WEIGHT VECTORS

Training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ and test point $\mathbf{x}_*$

SIDE-STEPPING THE WEIGHT VECTORS

Training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ and test point $\mathbf{x}_*$

$$p\left(\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \\ y_* \end{bmatrix} \mid \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \mathbf{x}_*\right) = N(\mu, \Sigma)$$

SIDE-STEPPING THE WEIGHT VECTORS

Training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ and test point $\mathbf{x}_*$

$$p\left(\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \\ y_* \end{bmatrix} \mid \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \mathbf{x}_*\right) = N(\mu, \Sigma)$$

Conditioning on the training data D:

SIDE-STEPPING THE WEIGHT VECTORS

Training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ and test point $\mathbf{x}_*$

$$p\left(\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \\ y_* \end{bmatrix} \mid \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \mathbf{x}_*\right) = N(\mu, \Sigma)$$

Conditioning on the training data D:

$$p(y_* \mid \mathbf{x}_*, D) = N(\mu_*, \sigma_*^2)$$

CLOSED-FORM PREDICTIONS

CLOSED-FORM PREDICTIONS

Given training set $D = (X, \mathrm{y})$ and test point $\mathbf{x}_*$:

$$p(y_* \mid \mathrm{x}_*, D) = N(\mu_*, \sigma_*^2),$$

CLOSED-FORM PREDICTIONS

Given training set $D = (X, \mathsf{y})$ and test point x_*:

$$p(y_* \mid \mathsf{x}_*, D) = N(\mu_*, \sigma_*^2),$$

where

$$\mu_* = K(\mathsf{x}_*, X)\ K^{-1}(X, X)\ \mathsf{y}$$

$$\sigma_*^2 = K(\mathsf{x}_*, \mathsf{x}_*) - K(\mathsf{x}_*, X)\ K^{-1}(X, X)\ K^\top(\mathsf{x}_*, X)$$

CLOSED-FORM PREDICTIONS

Given training set $D = (X, \mathrm{y})$ and test point x_*:

$$p(y_* \mid \mathrm{x}_*, D) = N(\mu_*, \sigma_*^2),$$

where

$$\mu_* = K(\mathrm{x}_*, X)\ K^{-1}(X, X)\ \mathrm{y}$$

$$\sigma_*^2 = K(\mathrm{x}_*, \mathrm{x}_*) - K(\mathrm{x}_*, X)\ K^{-1}(X, X)\ K^\top(\mathrm{x}_*, X)$$

$K(\mathrm{x}, \mathrm{x}')$ is the covariance between x and x'.

- PSD.

INTERPRETING THE PREDICTIONS

INTERPRETING THE PREDICTIONS

$$p(y_* \mid \mathbf{x}_*, D) = N(\mu_*, \sigma_*^2),$$

$$\mu_* = K(\mathbf{x}_*, X)\ K^{-1}(X, X)\ \mathbf{y}$$

$$\sigma_*^2 = K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, X)\ K^{-1}(X, X)\ K^{\top}(\mathbf{x}_*, X)$$

INTERPRETING THE PREDICTIONS

$$p(y_* \mid \mathbf{x}_*, D) = N(\mu_*, \sigma_*^2),$$

$$\mu_* = K(\mathbf{x}_*, X)\ K^{-1}(X, X)\ \mathbf{y}$$ **: kernel regression!!**

$$\sigma_*^2 = K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, X)\ K^{-1}(X, X)\ K^{\top}(\mathbf{x}_*, X)$$

INTERPRETING THE PREDICTIONS

$$p(y_* \mid \mathbf{x}_*, D) = N(\mu_*, \sigma_*^2),$$

$\mu_* = K(\mathbf{x}_*, X)\ K^{-1}(X, X)\ \mathbf{y}$: **kernel regression!!**

$\sigma_*^2 = K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, X)\ K^{-1}(X, X)\ K^\top(\mathbf{x}_*, X)$

- $\sigma_*^2 \leq K(\mathbf{x}_*, \mathbf{x}_*)$: uncertainty **never** increases

INTERPRETING THE PREDICTIONS

$$p(y_* \mid \mathbf{x}_*, D) = N(\mu_*, \sigma_*^2),$$

$\mu_* = K(\mathbf{x}_*, X)\ K^{-1}(X, X)\ \mathbf{y}$: **kernel regression!!**

$\sigma_*^2 = K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, X)\ K^{-1}(X, X)\ K^\top(\mathbf{x}_*, X)$

- $\sigma_*^2 \leq K(\mathbf{x}_*, \mathbf{x}_*)$: uncertainty **never** increases
- Uncertainty quantification

INTERPRETING THE PREDICTIONS

$$p(y_* \mid \mathbf{x}_*, D) = N(\mu_*, \sigma_*^2),$$

$$\mu_* = K(\mathbf{x}_*, X)\ K^{-1}(X, X)\ \mathbf{y}$$ **: kernel regression!!**

$$\sigma_*^2 = K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, X)\ K^{-1}(X, X)\ K^\top(\mathbf{x}_*, X)$$

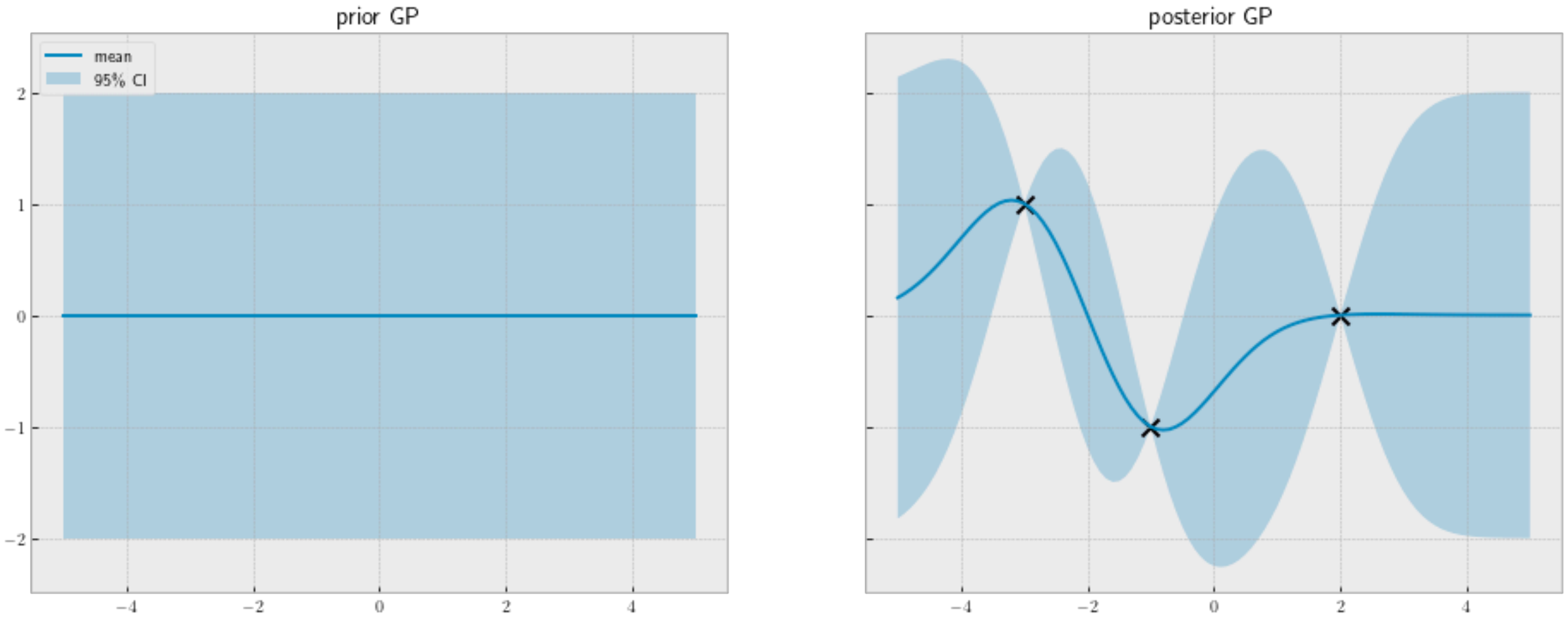
- $\sigma_*^2 \leq K(\mathbf{x}_*, \mathbf{x}_*)$: uncertainty **never** increases
- Uncertainty quantification

$\mathbf{x}_*$ can be anything

- Distribution over **functions**

GP IN ACTION

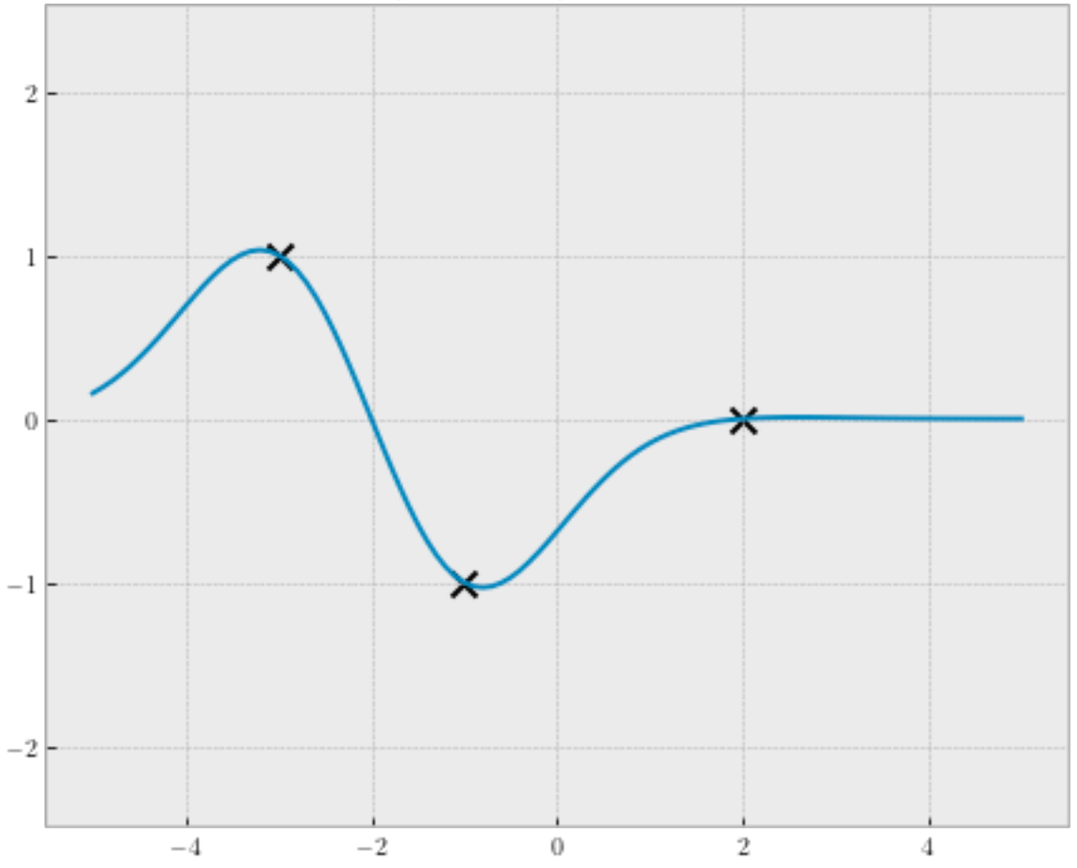
RBF kernel: $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2}\right)$



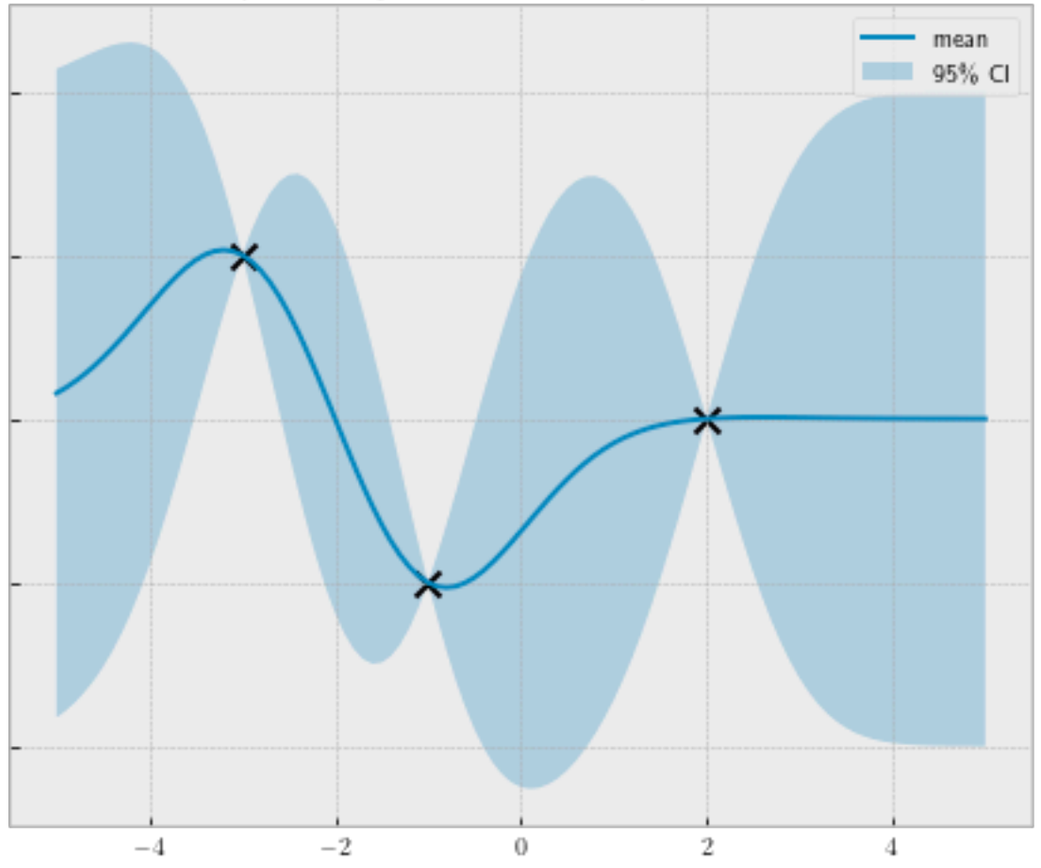
GP IN ACTION

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

point-wise predictions

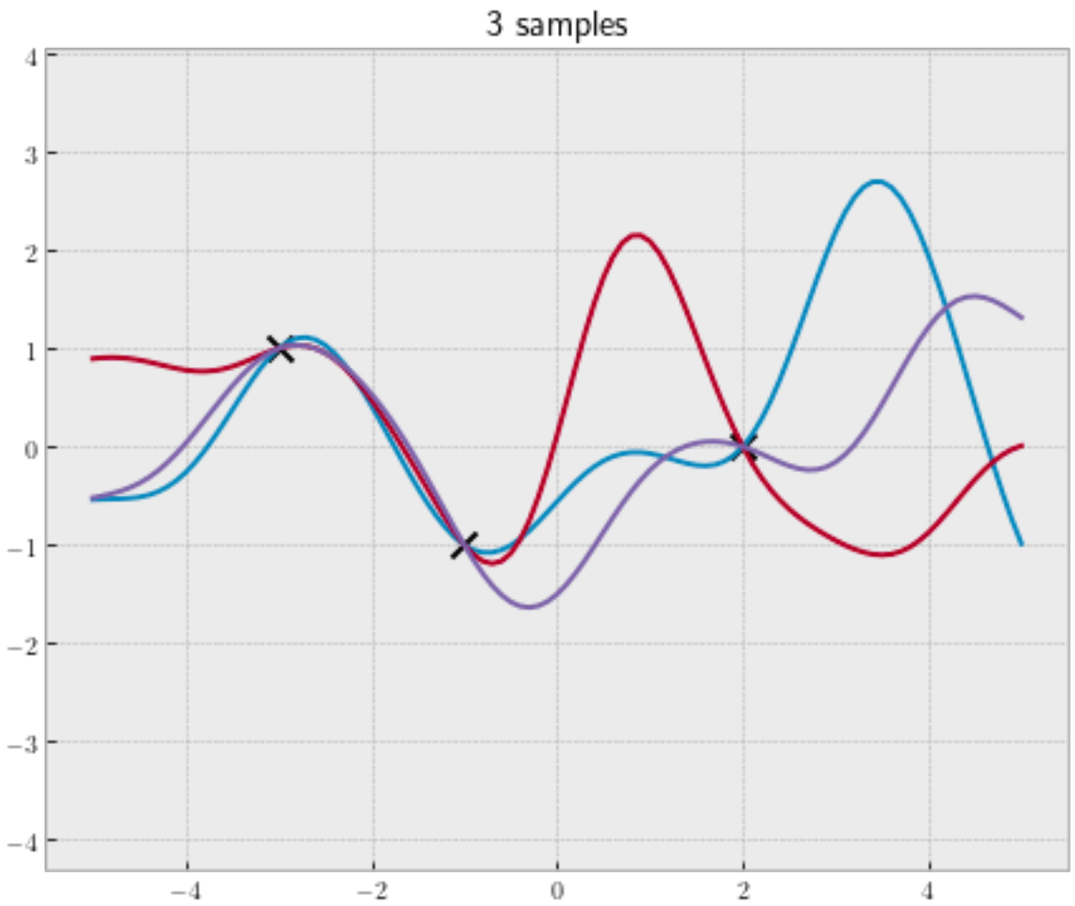


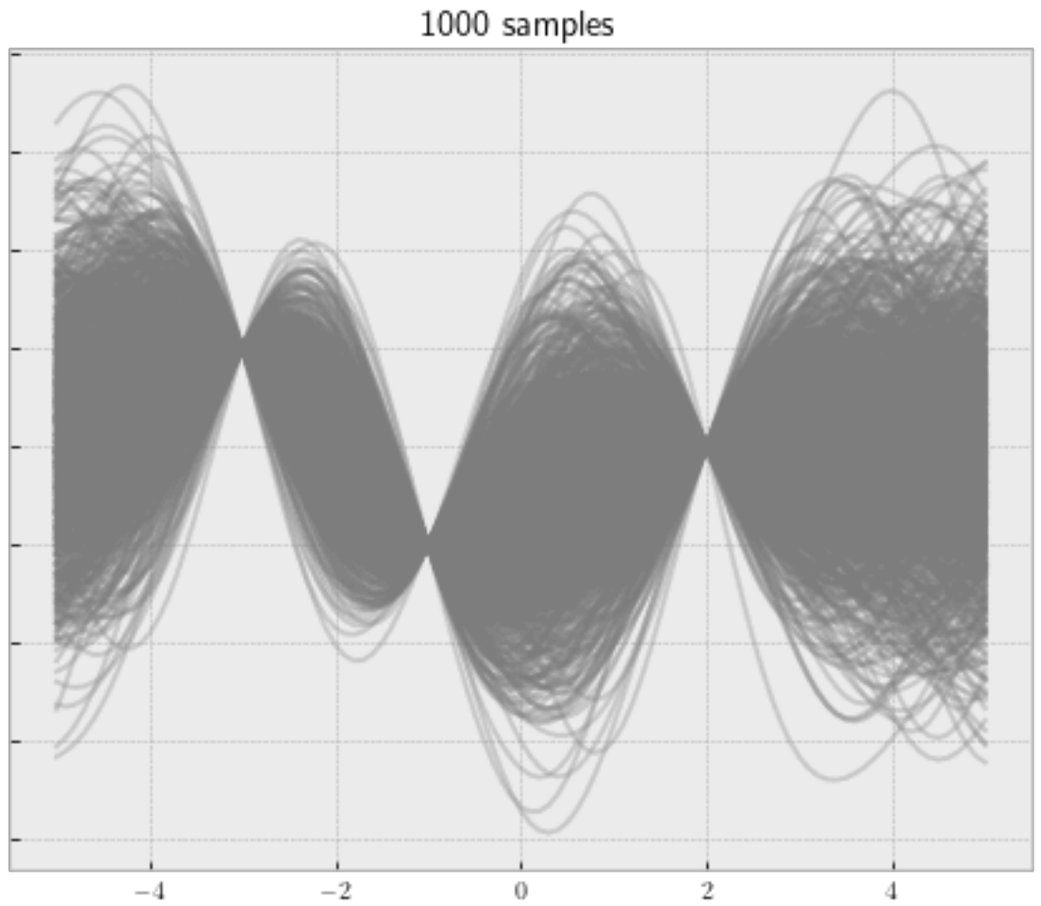
probability distributions as predictions



GP IN ACTION

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





NOISY OBSERVATIONS

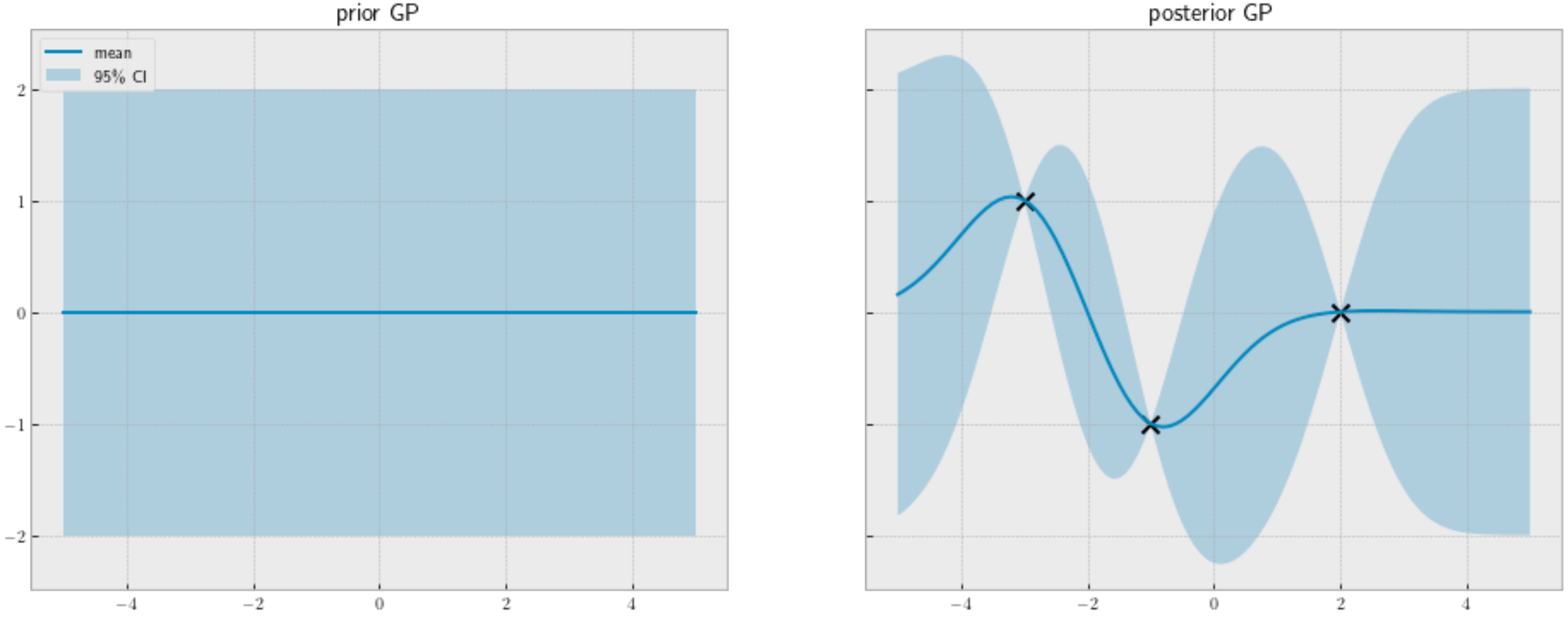
Each observation y follows a slightly different normal:

$$p(y \mid \mathrm{x}) = N\left(y; \mu(\mathrm{x}), K(\mathrm{x}, \mathrm{x}) + \sigma_n^2\right) \qquad K(X, X) \to K(X, X) + \sigma_n^2 I$$

NOISY OBSERVATIONS

Each observation y follows a slightly different normal:

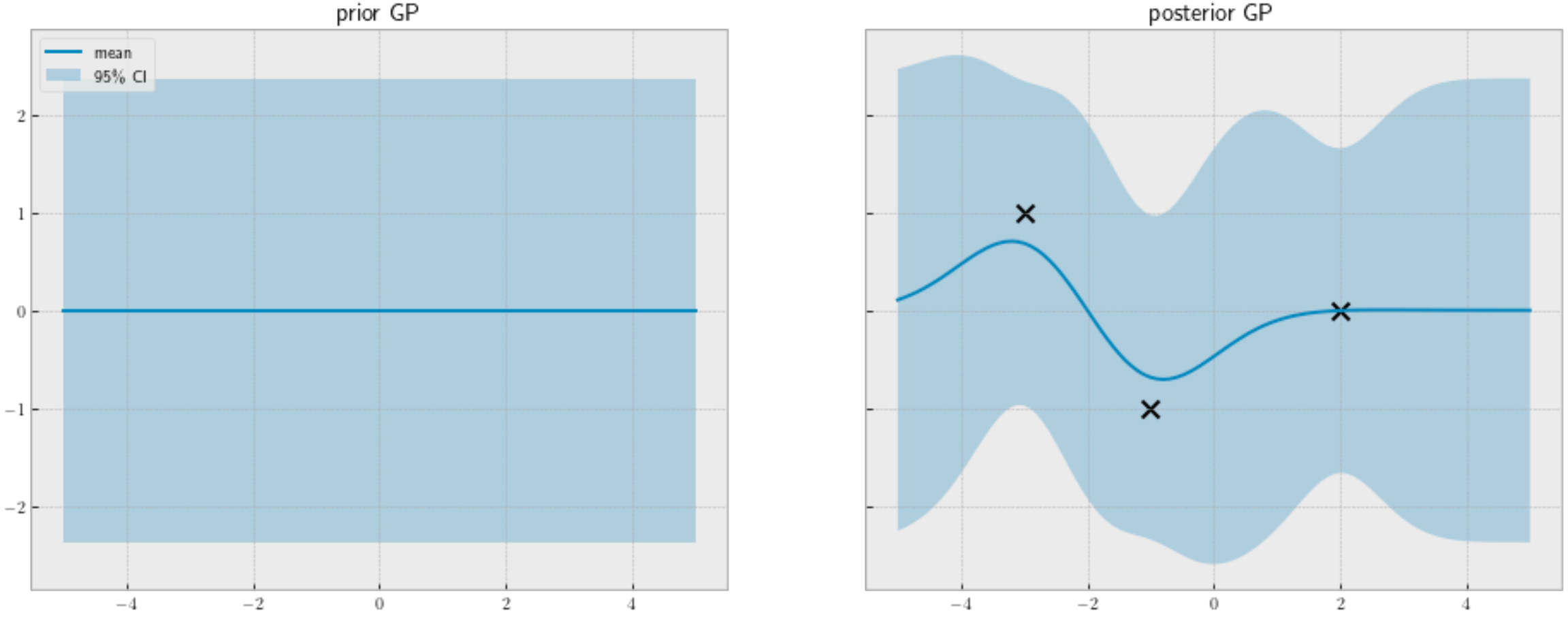
$$p(y \mid \mathrm{x}) = N\left(y; \mu(\mathrm{x}), K(\mathrm{x}, \mathrm{x}) + \sigma_n^2\right) \qquad K(X, X) \to K(X, X) + \sigma_n^2 I$$



NOISY OBSERVATIONS

Each observation y follows a slightly different normal:

$$p(y \mid \mathrm{x}) = N\left(y; \mu(\mathrm{x}), K(\mathrm{x}, \mathrm{x}) + \sigma_n^2\right) \qquad K(X, X) \rightarrow K(X, X) + \sigma_n^2 I$$



FORMAL DEFINITION OF A GP

FORMAL DEFINITION OF A GP

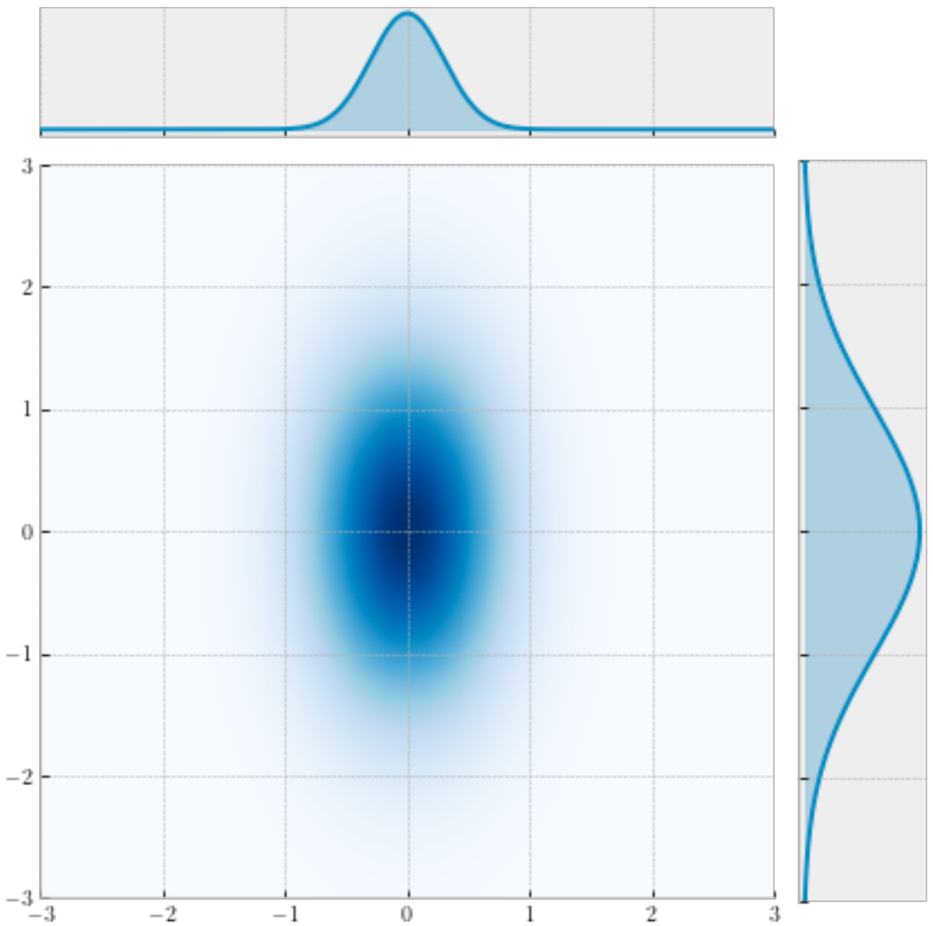
Definition: A GP is a (potentially infinite) collection of RVs such that the **joint** distribution of **every** finite subset of RVs is multivariate Gaussian.

UNCORRELATED VARIABLES

$$\Sigma = \begin{bmatrix} 0.3 & 0 \\ 0 & 1 \end{bmatrix}$$

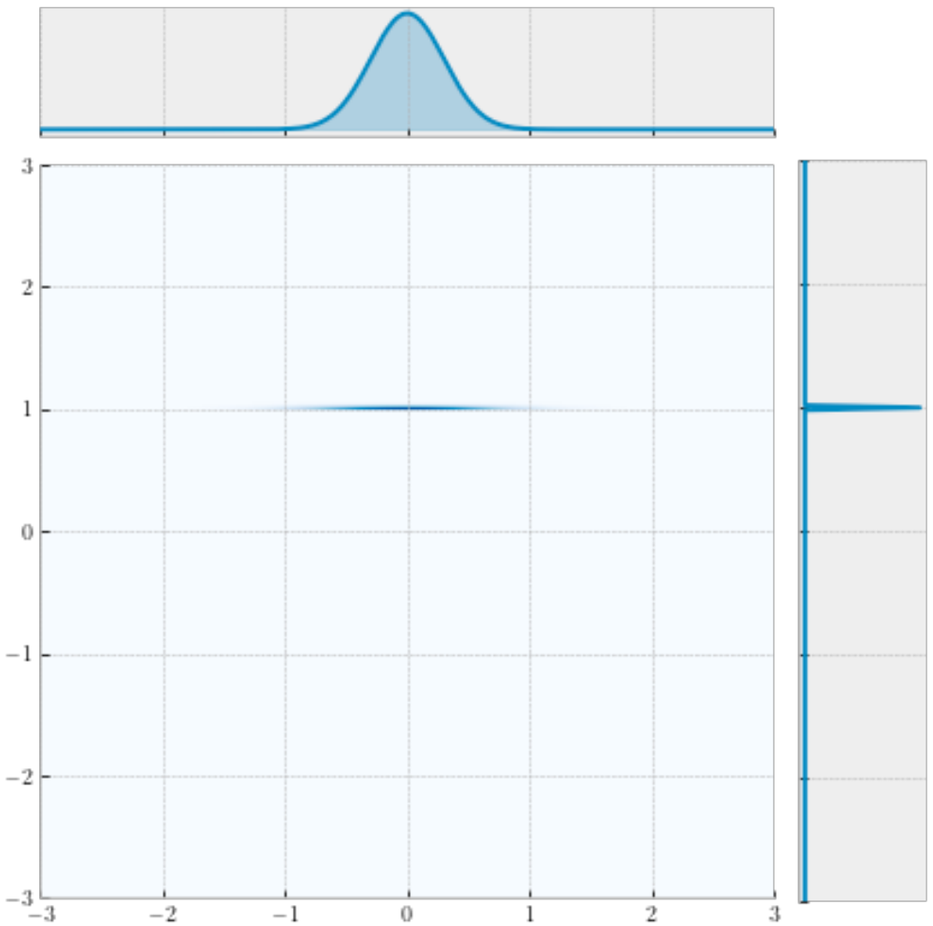
UNCORRELATED VARIABLES

$$\Sigma = \begin{bmatrix} 0.3 & 0 \\ 0 & 1 \end{bmatrix}$$



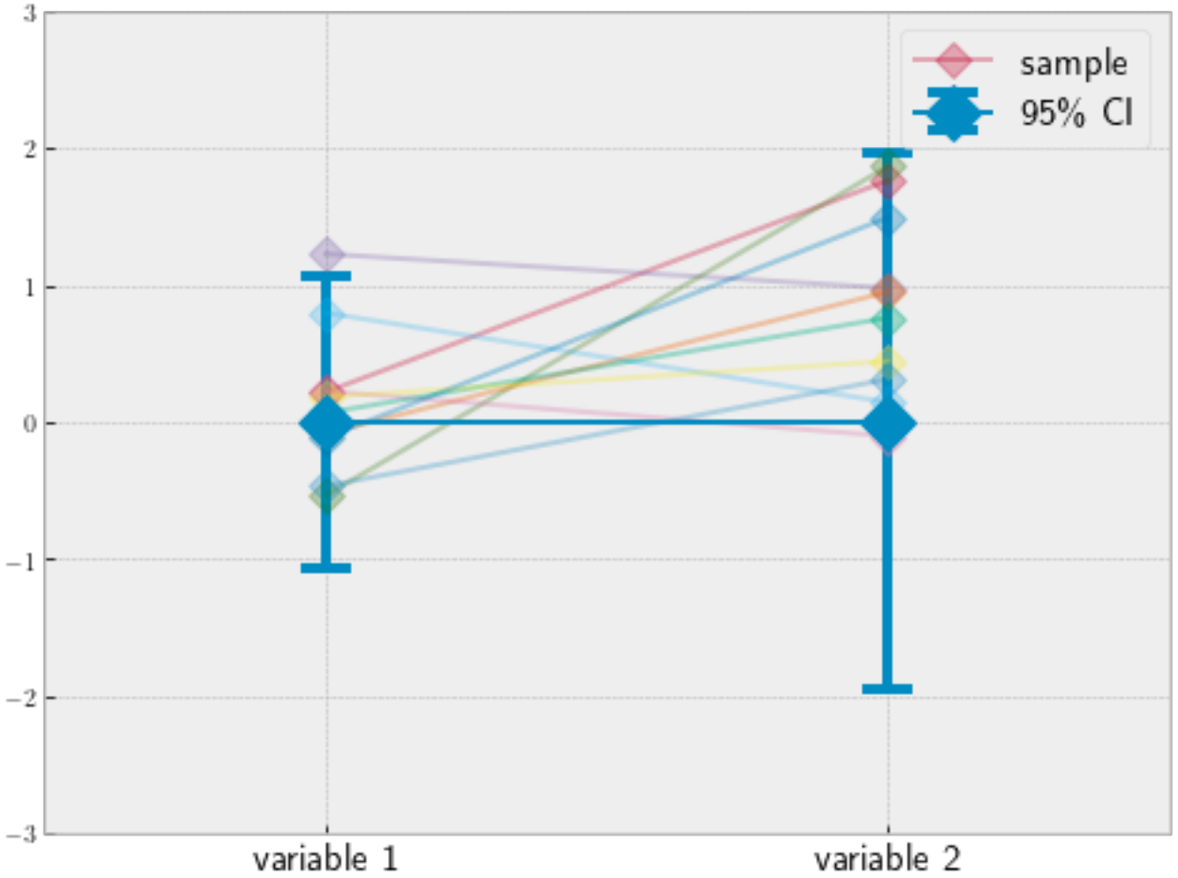
UNCORRELATED VARIABLES

$$\Sigma = \begin{bmatrix} 0.3 & 0 \\ 0 & 1 \end{bmatrix}$$



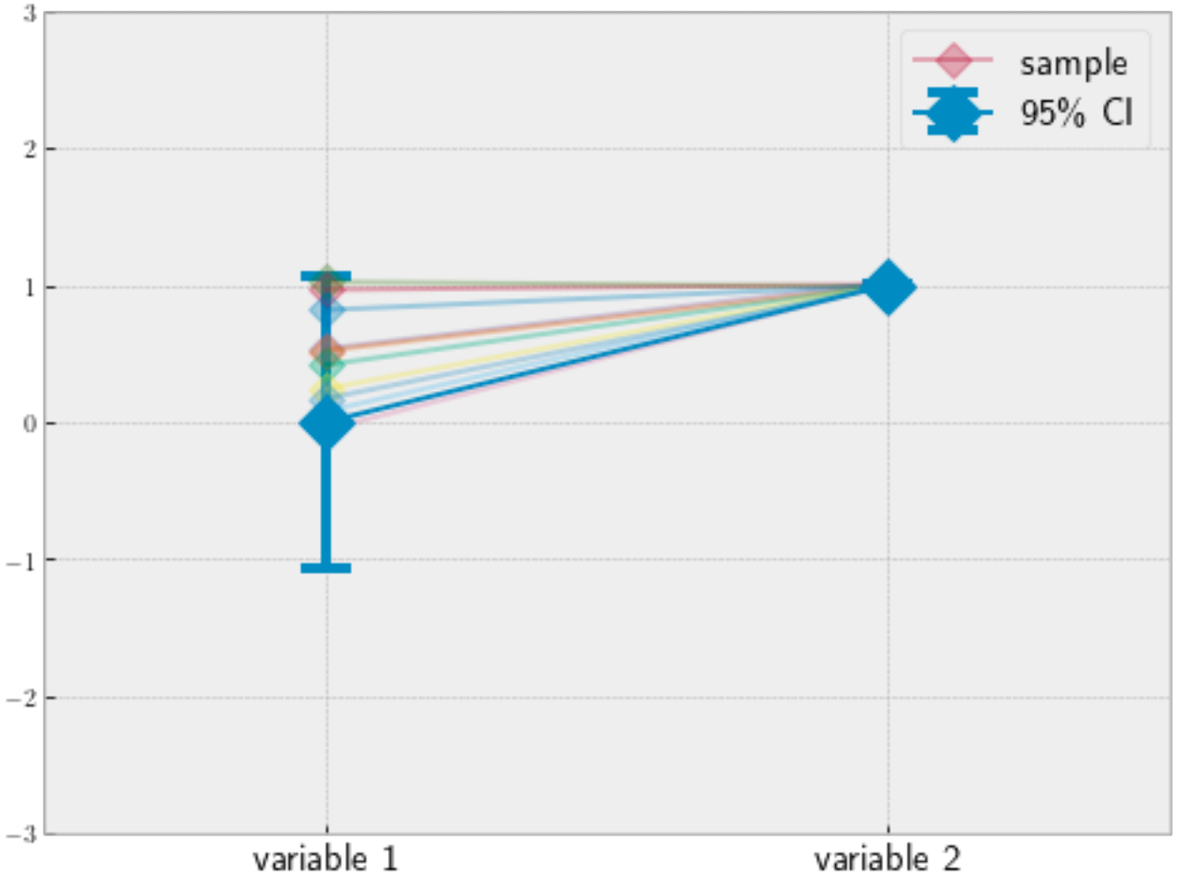
UNCORRELATED VARIABLES

$$\Sigma = \begin{bmatrix} 0.3 & 0 \\ 0 & 1 \end{bmatrix}$$



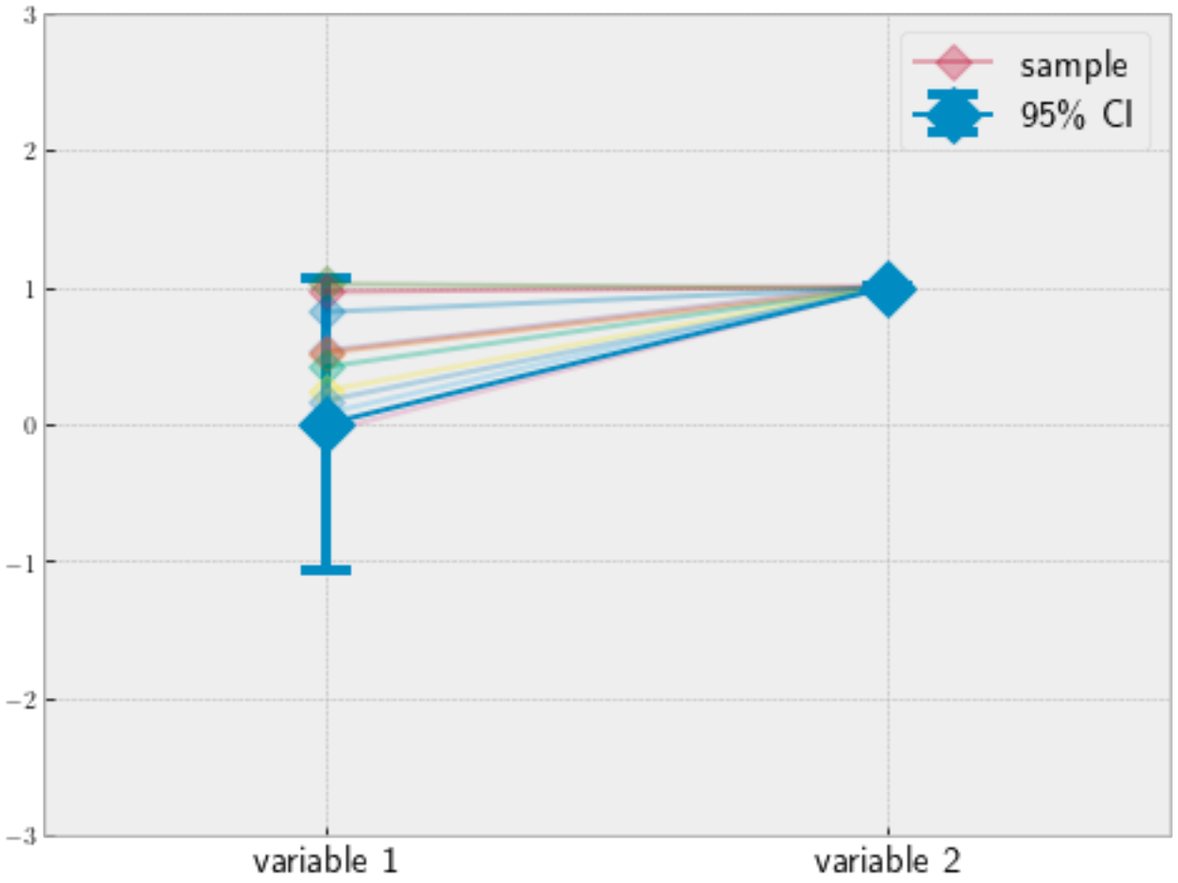
UNCORRELATED VARIABLES

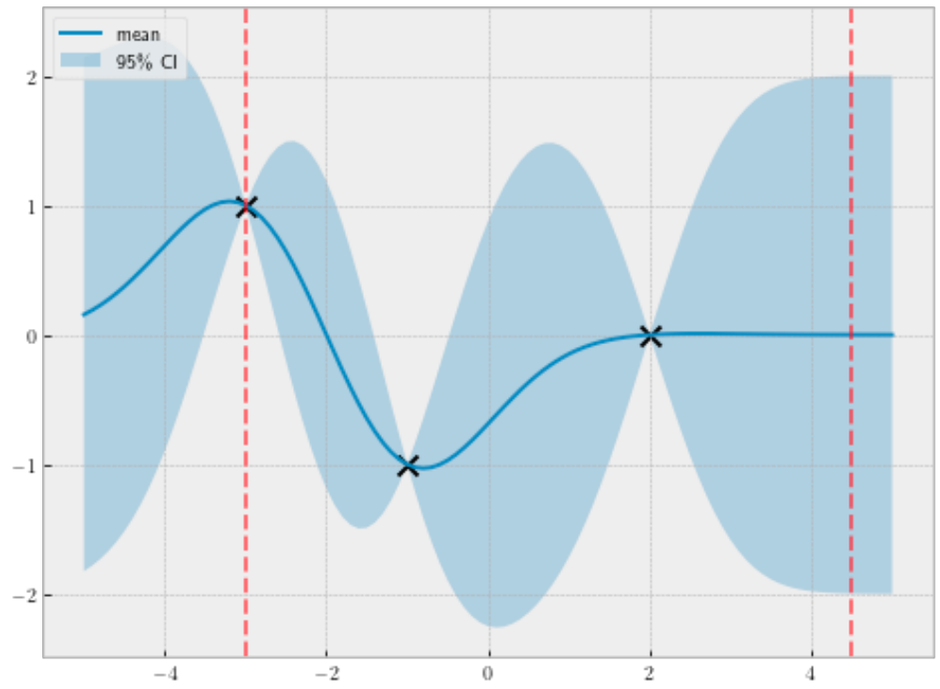
$$\Sigma = \begin{bmatrix} 0.3 & 0 \\ 0 & 1 \end{bmatrix}$$



UNCORRELATED VARIABLES

$$\Sigma = \begin{bmatrix} 0.3 & 0 \\ 0 & 1 \end{bmatrix}$$



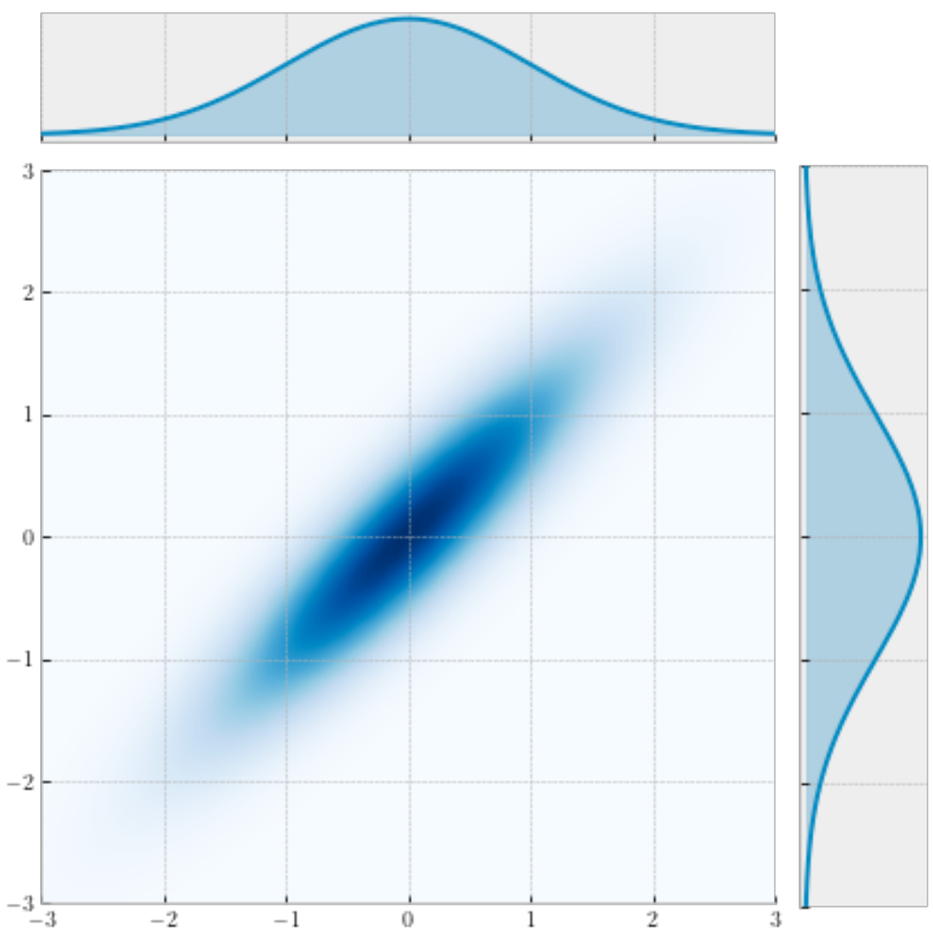


CORRELATED VARIABLES

$$\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$

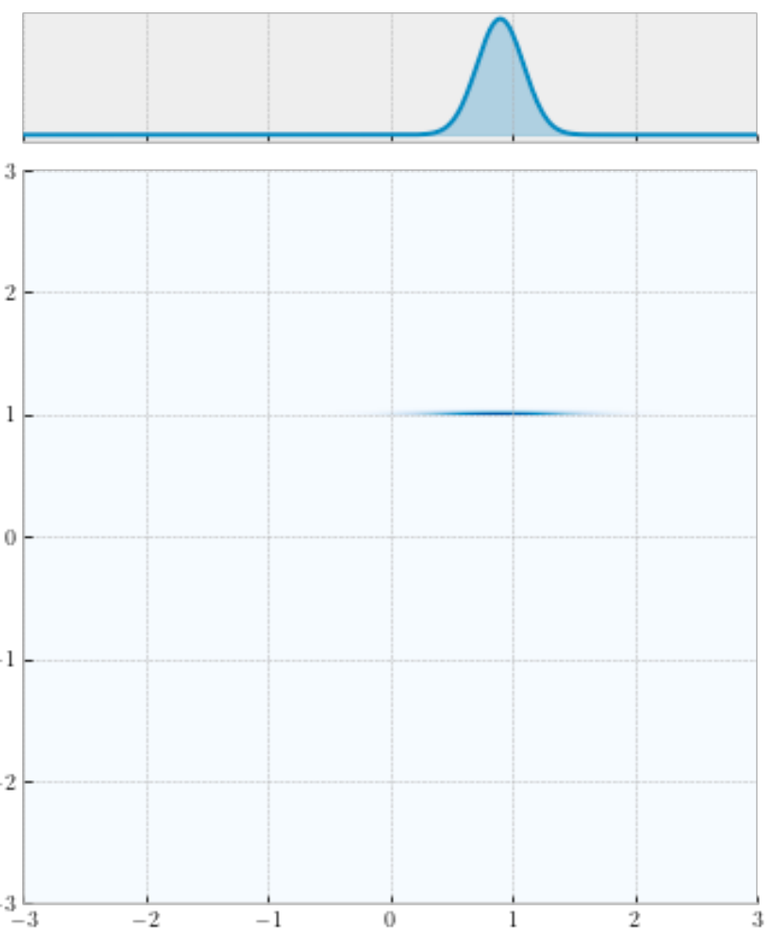
CORRELATED VARIABLES

$$\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$



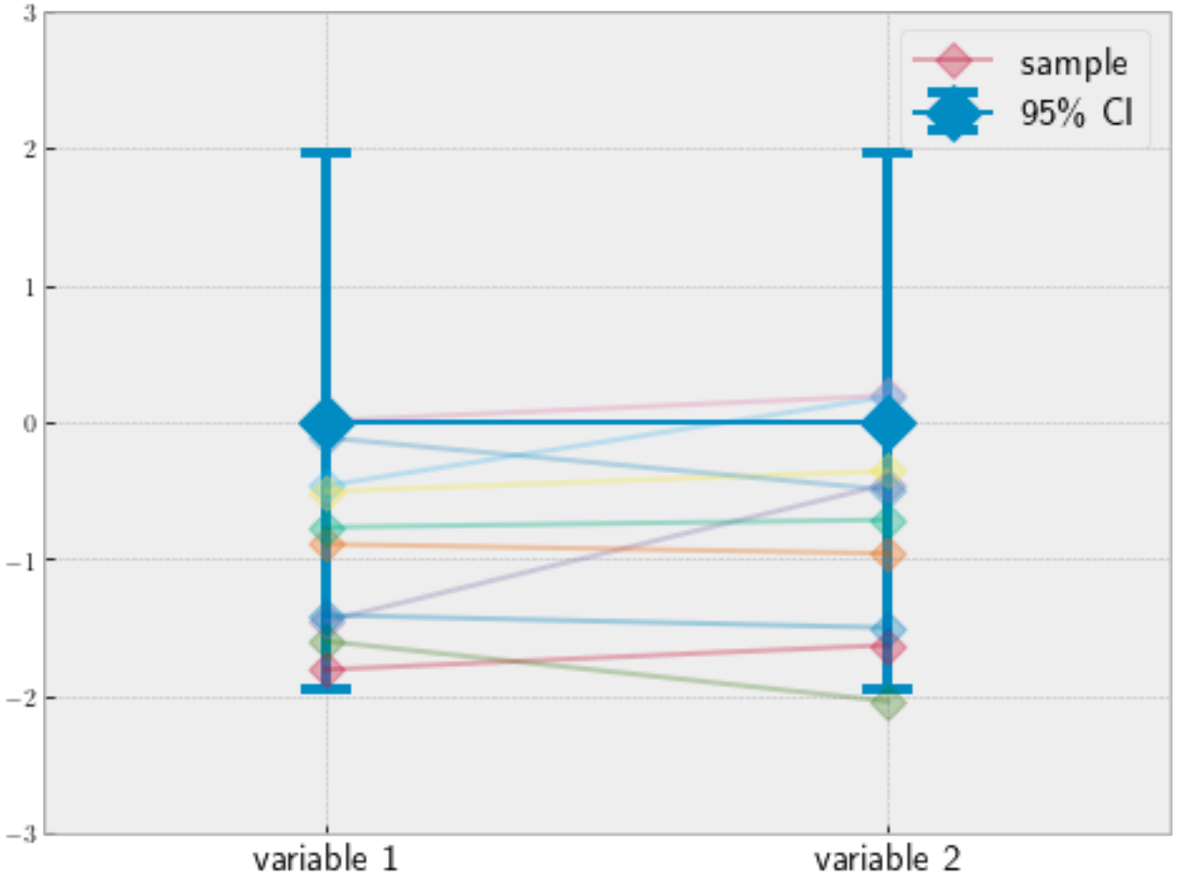
CORRELATED VARIABLES

$$\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$



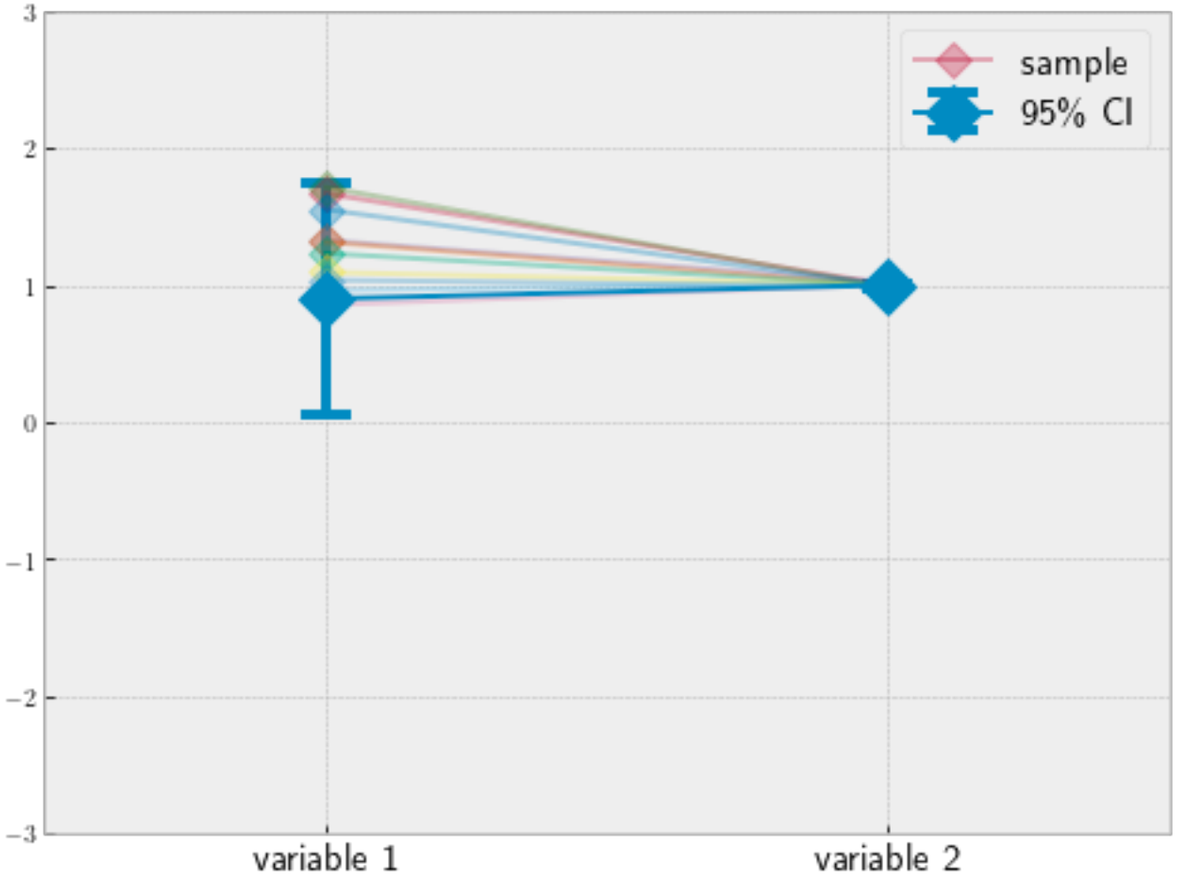
CORRELATED VARIABLES

$$\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$



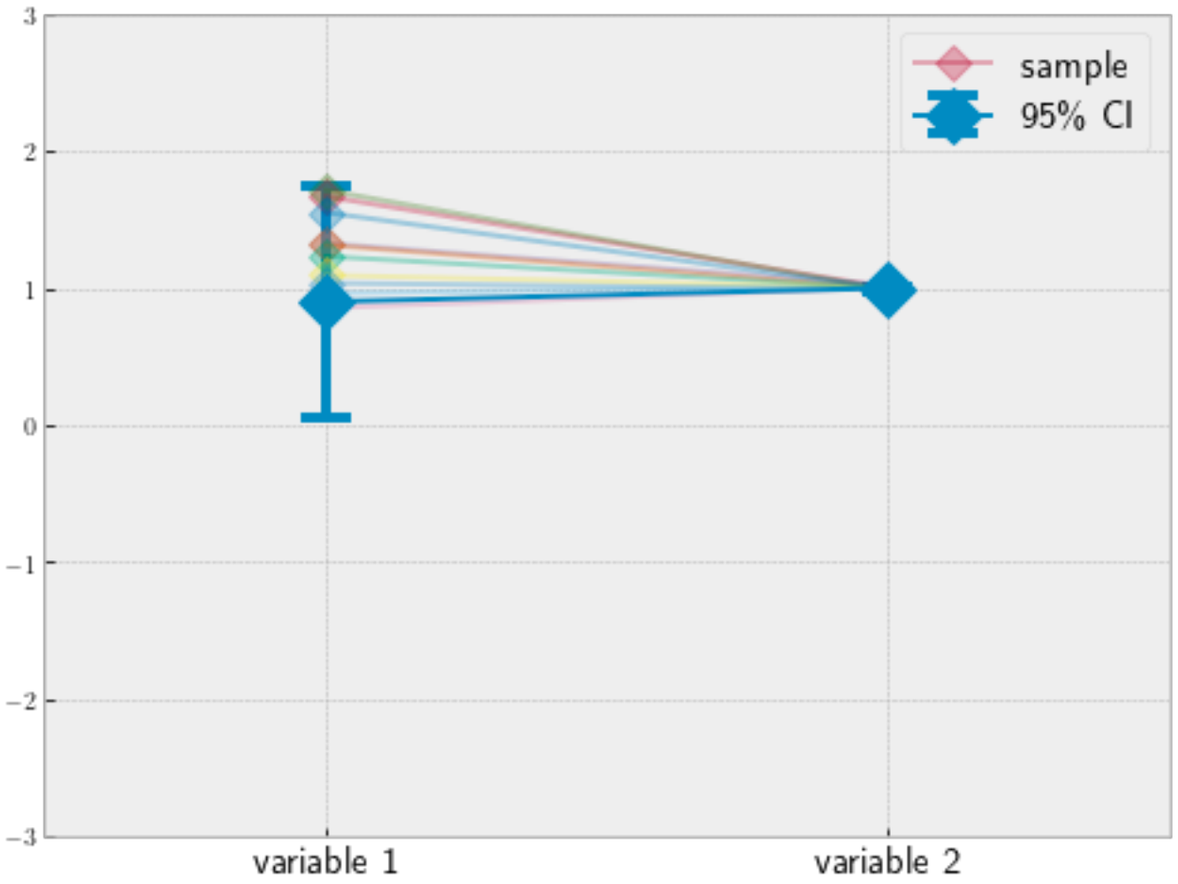
CORRELATED VARIABLES

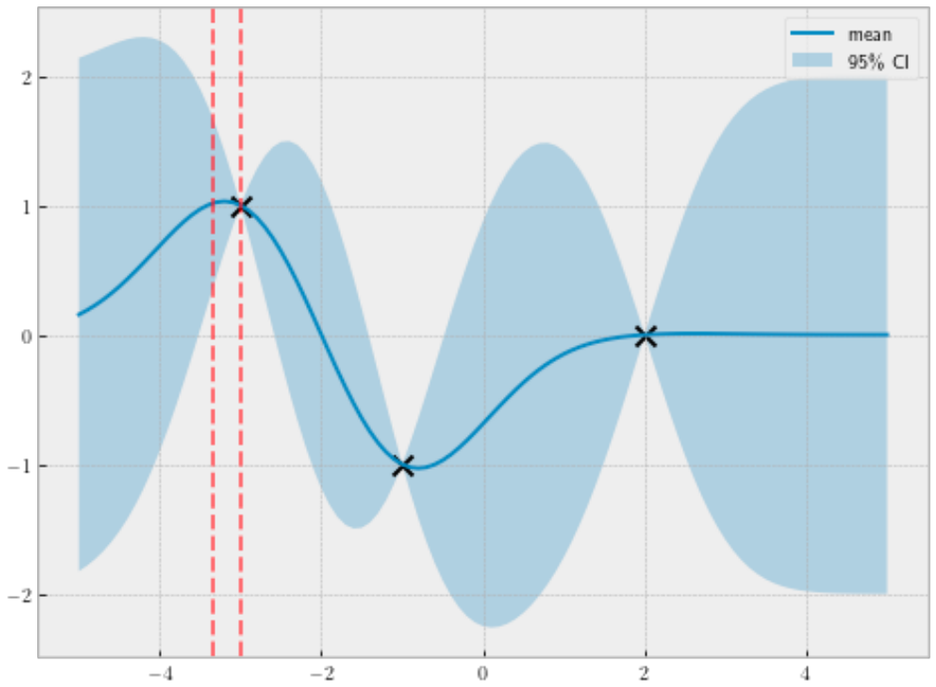
$$\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$



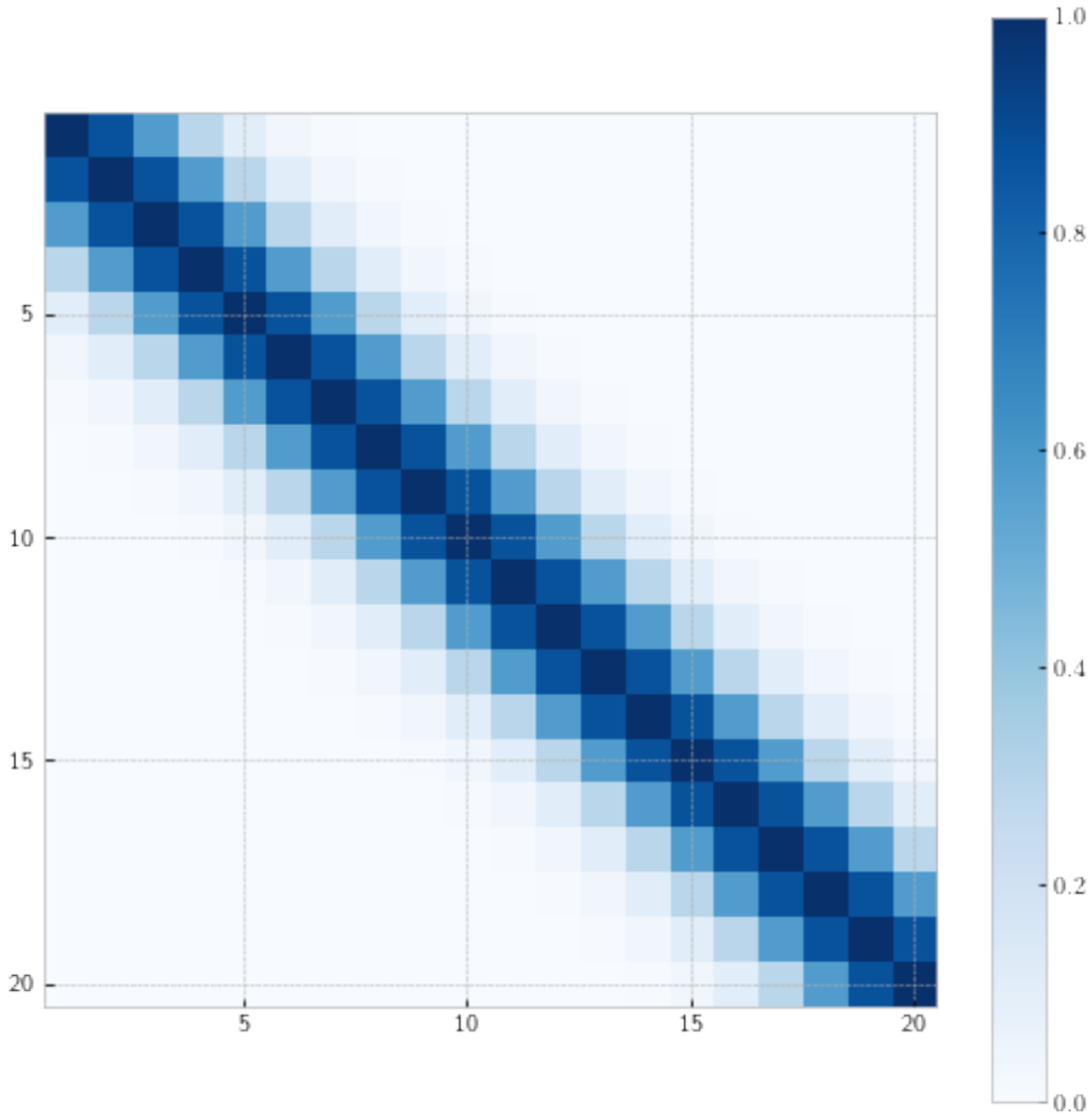
CORRELATED VARIABLES

$$\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$

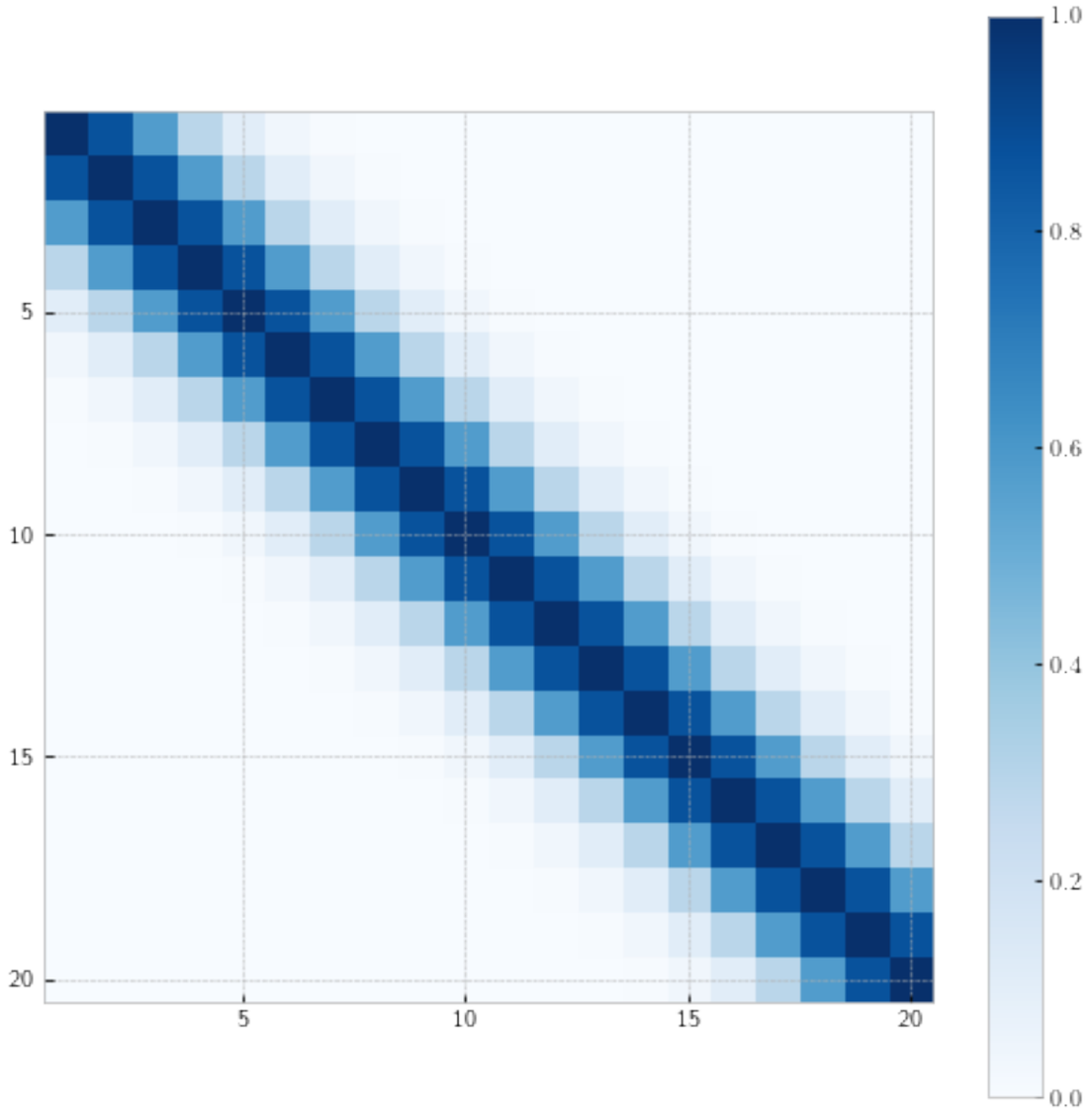


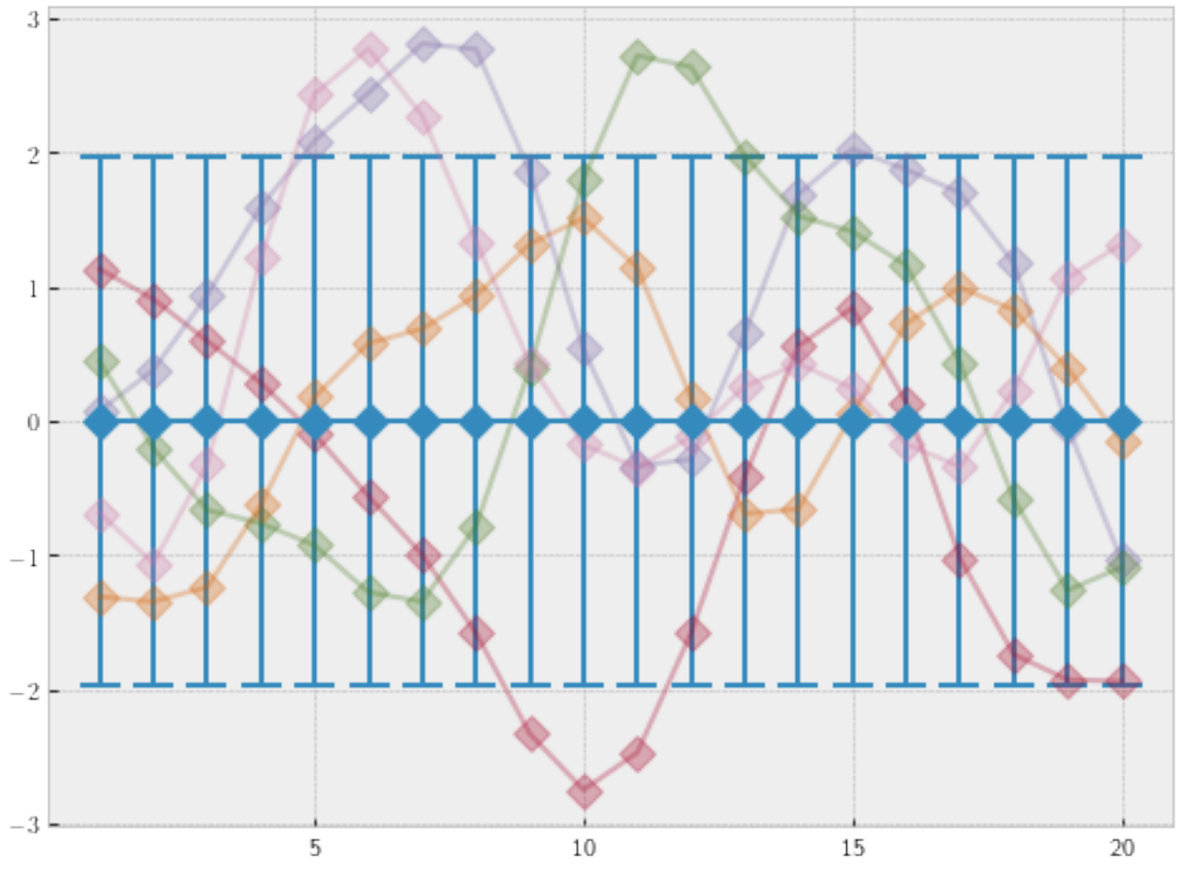


FROM 2D TO HIGHER/INFINITE DIMENSIONS

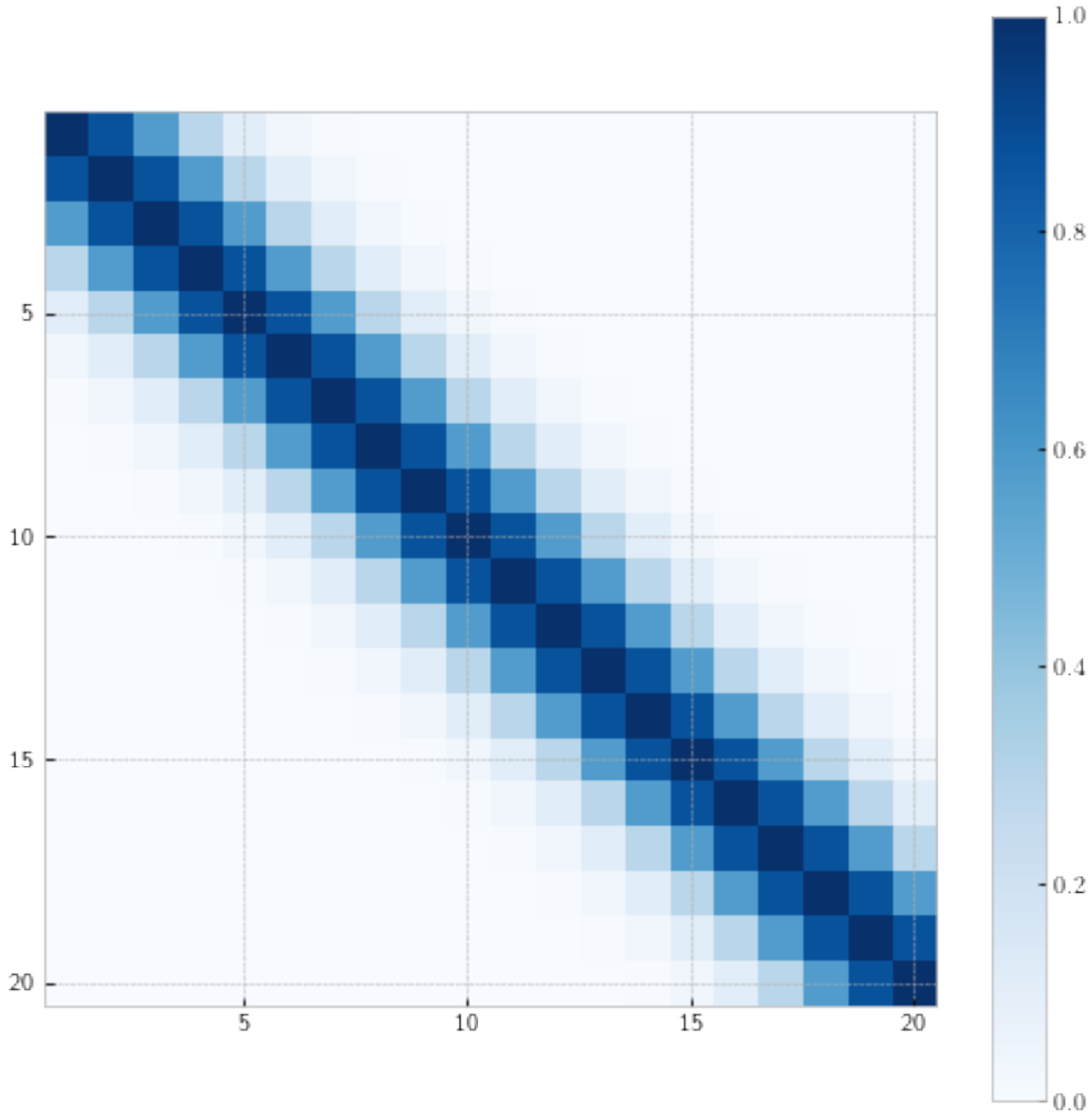


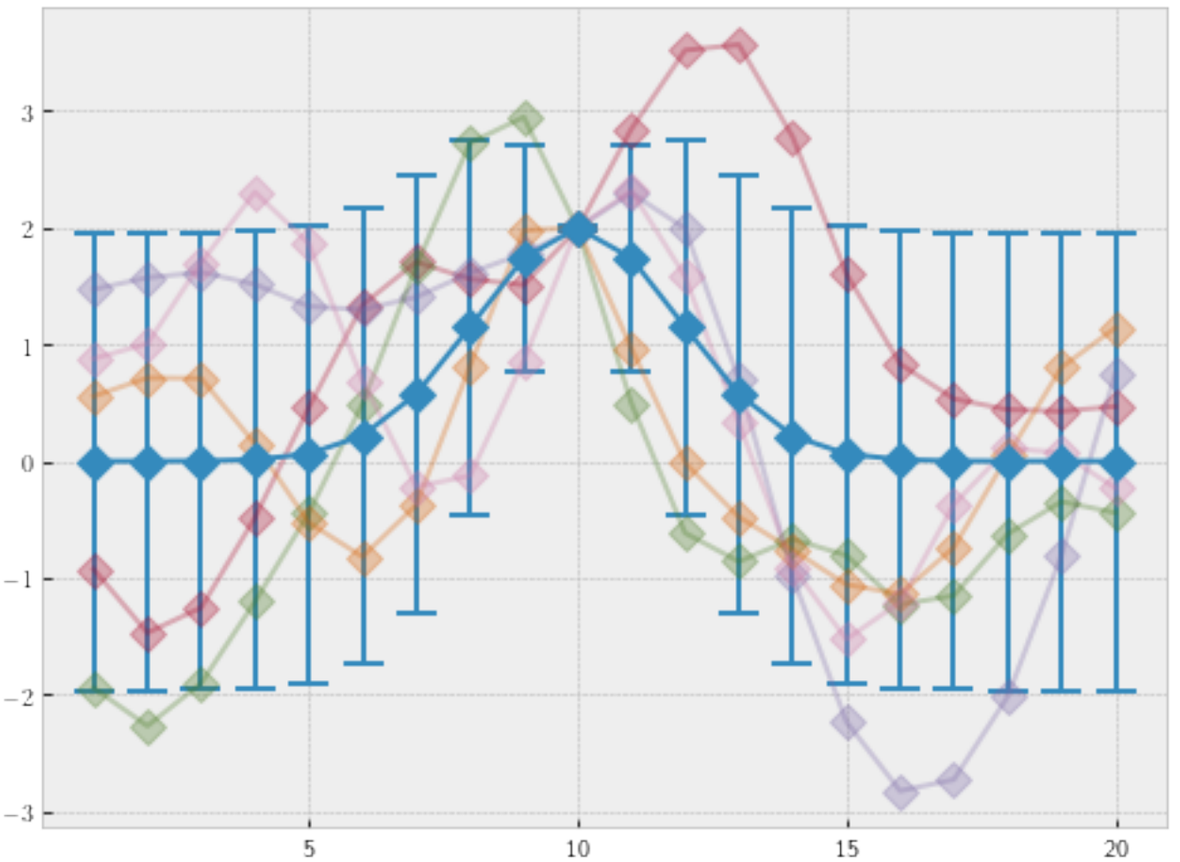
FROM 2D TO HIGHER/INFINITE DIMENSIONS



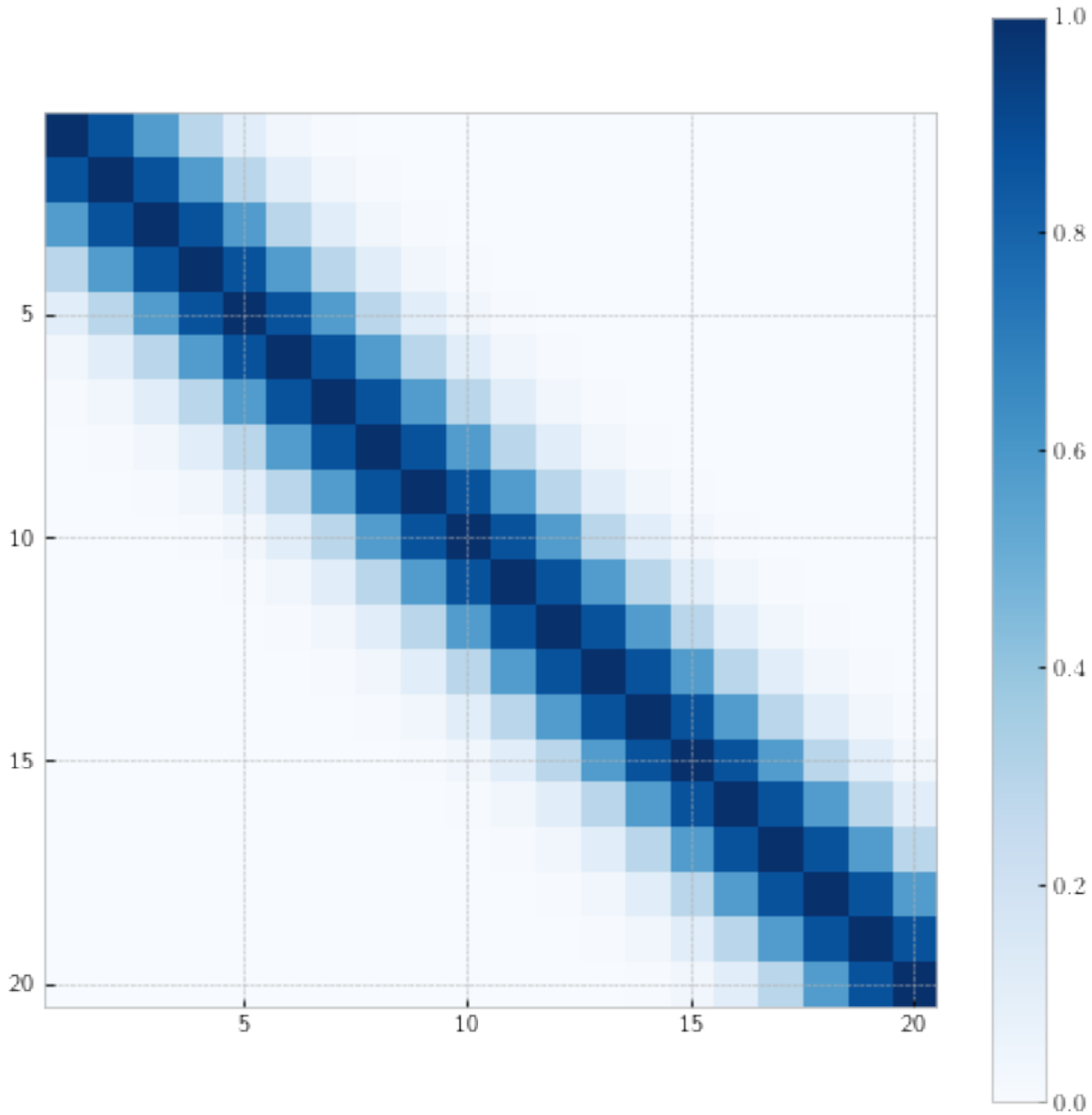


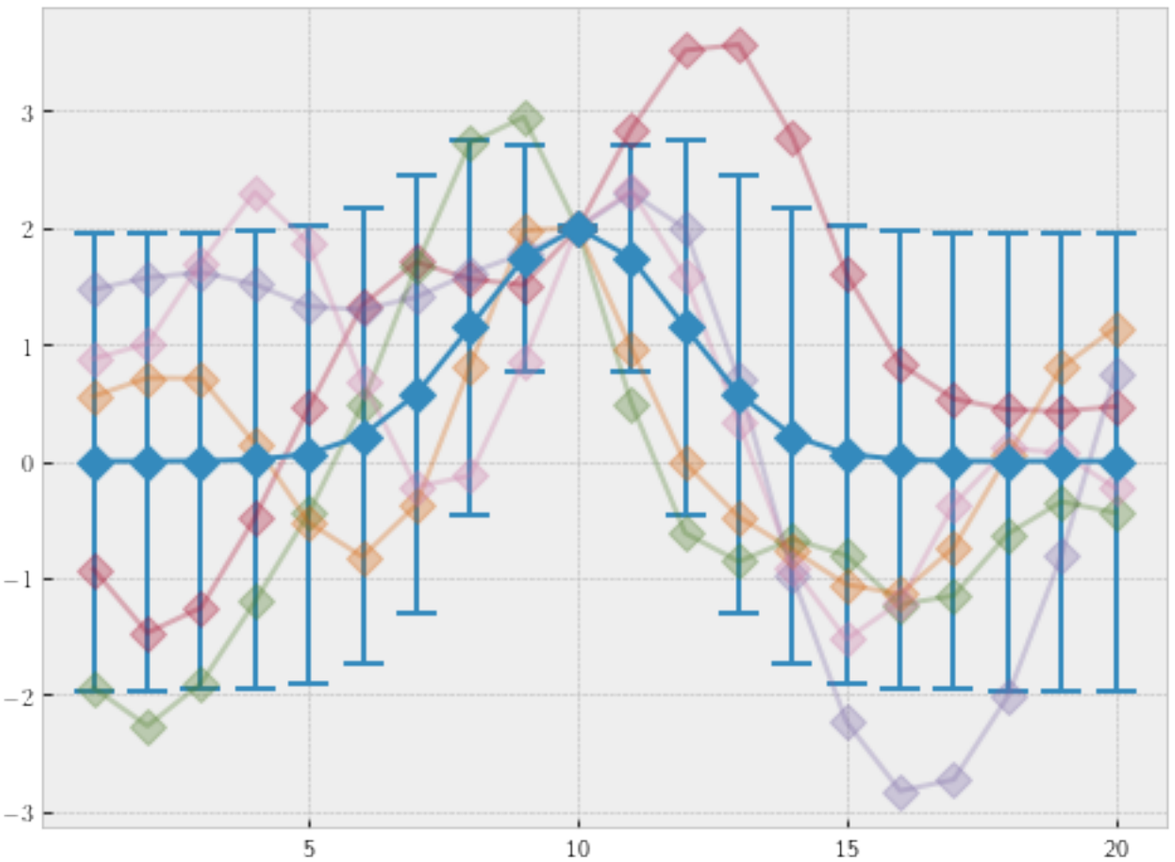
FROM 2D TO HIGHER/INFINITE DIMENSIONS





FROM 2D TO HIGHER/INFINITE DIMENSIONS





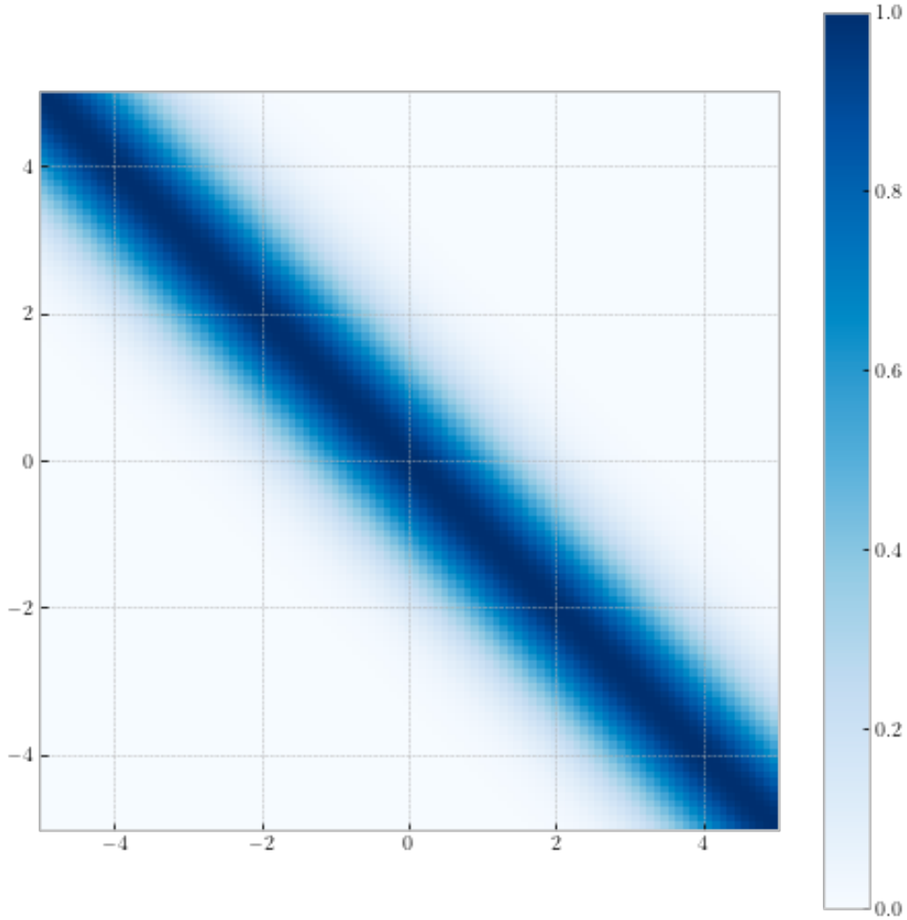
–> Tying a **knot**

RBF KERNEL

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

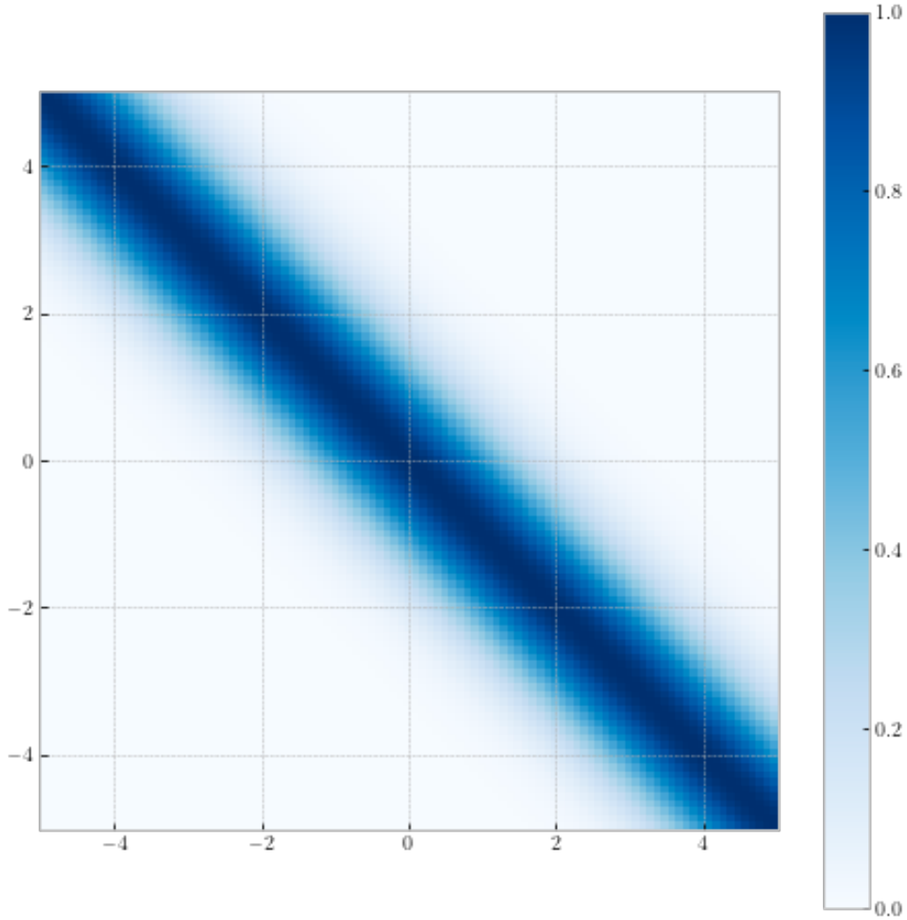
RBF KERNEL

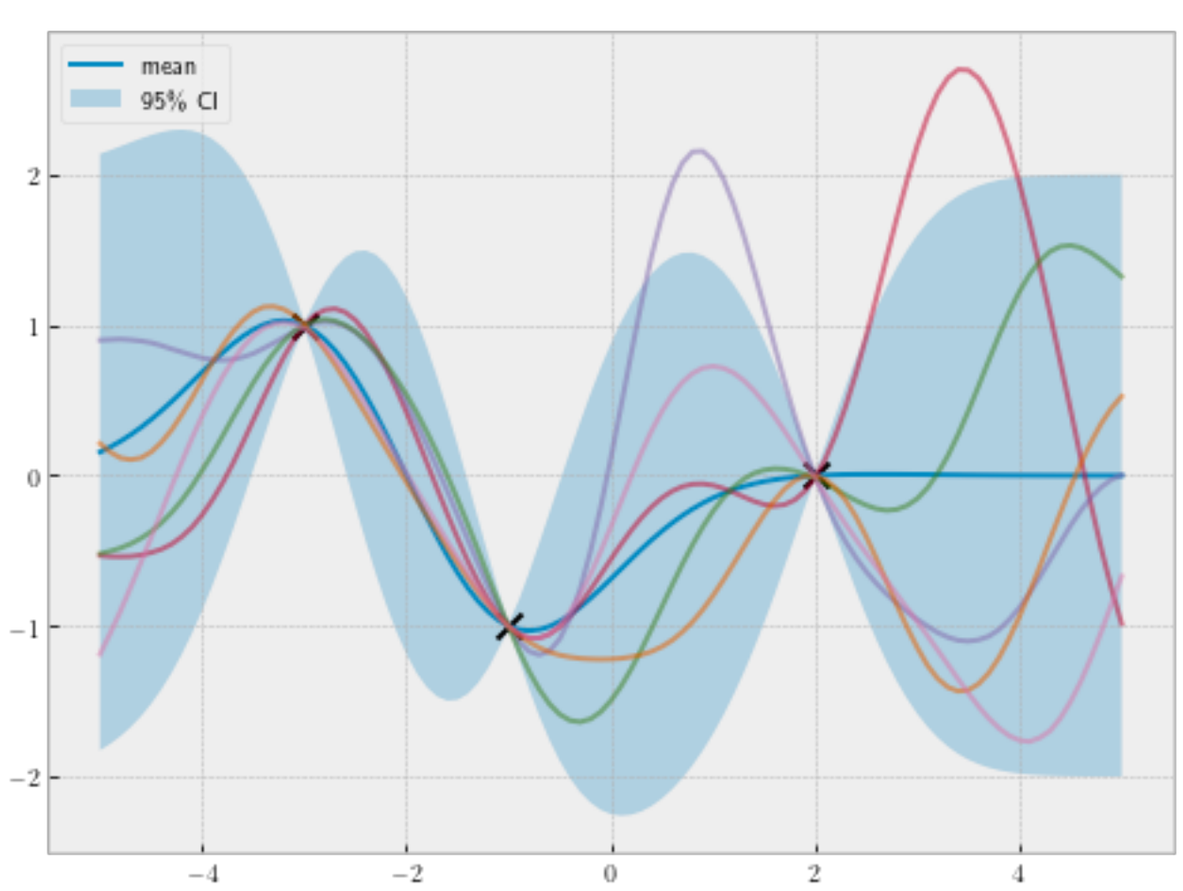
$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2}\right)$$



RBF KERNEL

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



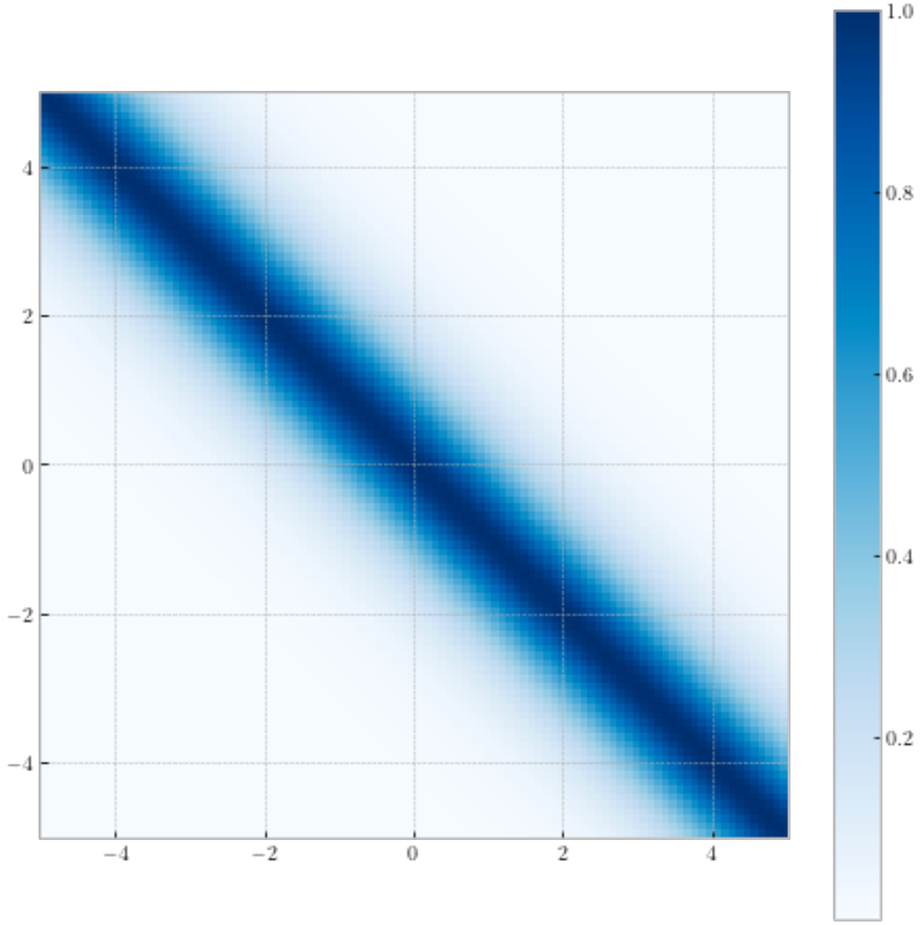


MATERN 5/2 KERNEL

$$K(\mathrm{x}, \mathrm{x}') = \left(1 + \sqrt{5}d + 5d^2\right) \exp\left(-\sqrt{5}d\right)$$

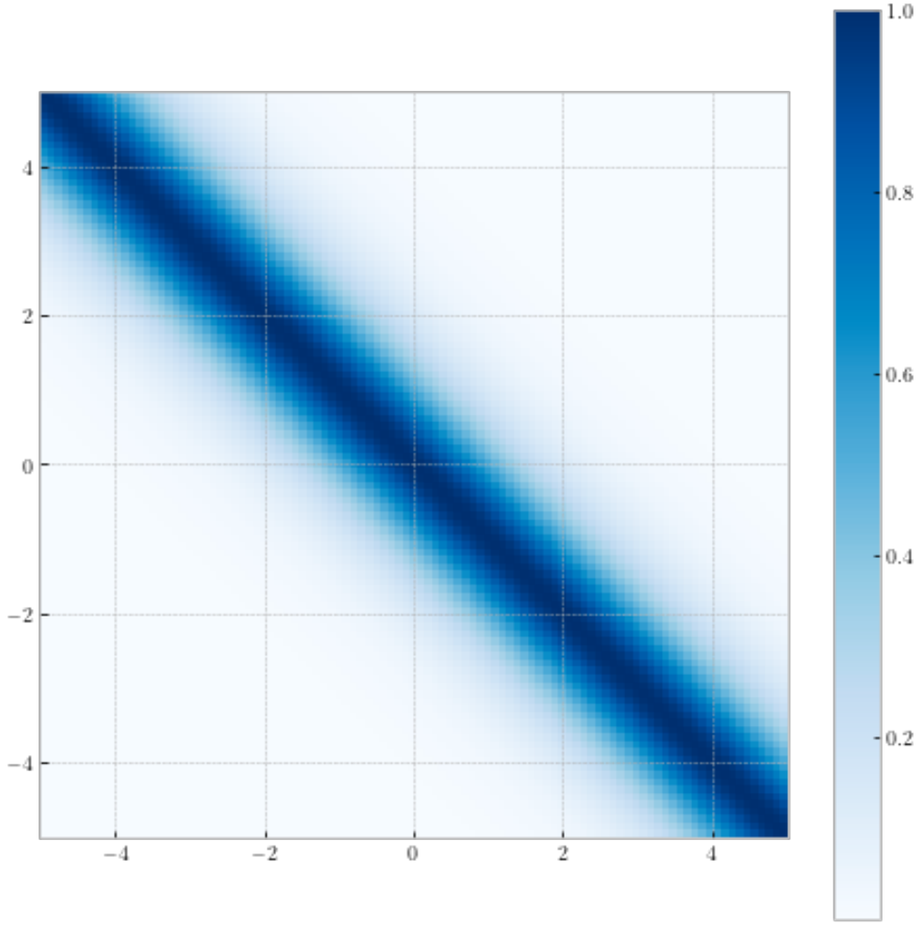
MATERN 5/2 KERNEL

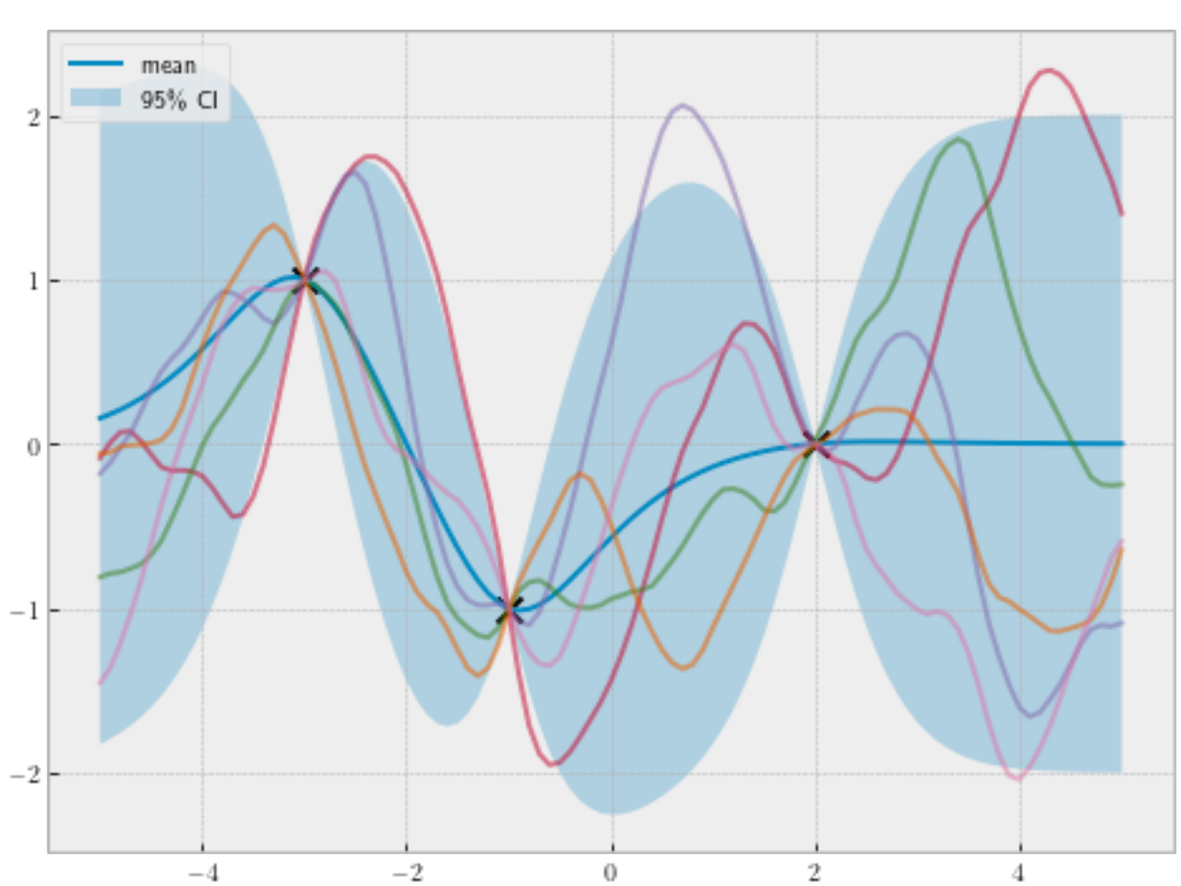
$$K(\mathrm{x}, \mathrm{x}') = \left(1 + \sqrt{5}d + 5d^2\right) \exp\left(-\sqrt{5}d\right)$$



MATERN 5/2 KERNEL

$$K(\mathrm{x}, \mathrm{x}') = \left(1 + \sqrt{5}d + 5d^2\right) \exp\left(-\sqrt{5}d\right)$$



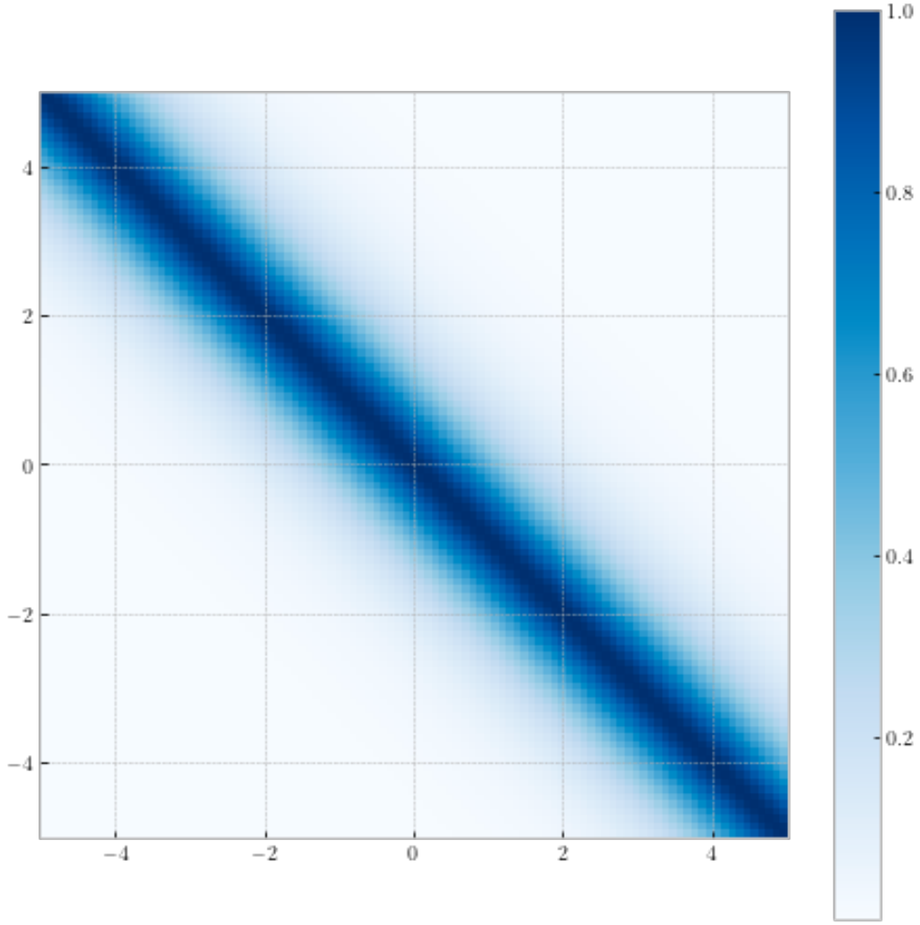


MATERN 3/2 KERNEL

$$K(\mathrm{x}, \mathrm{x}') = \left(1 + \sqrt{3}d\right) \exp\left(-\sqrt{3}d\right)$$

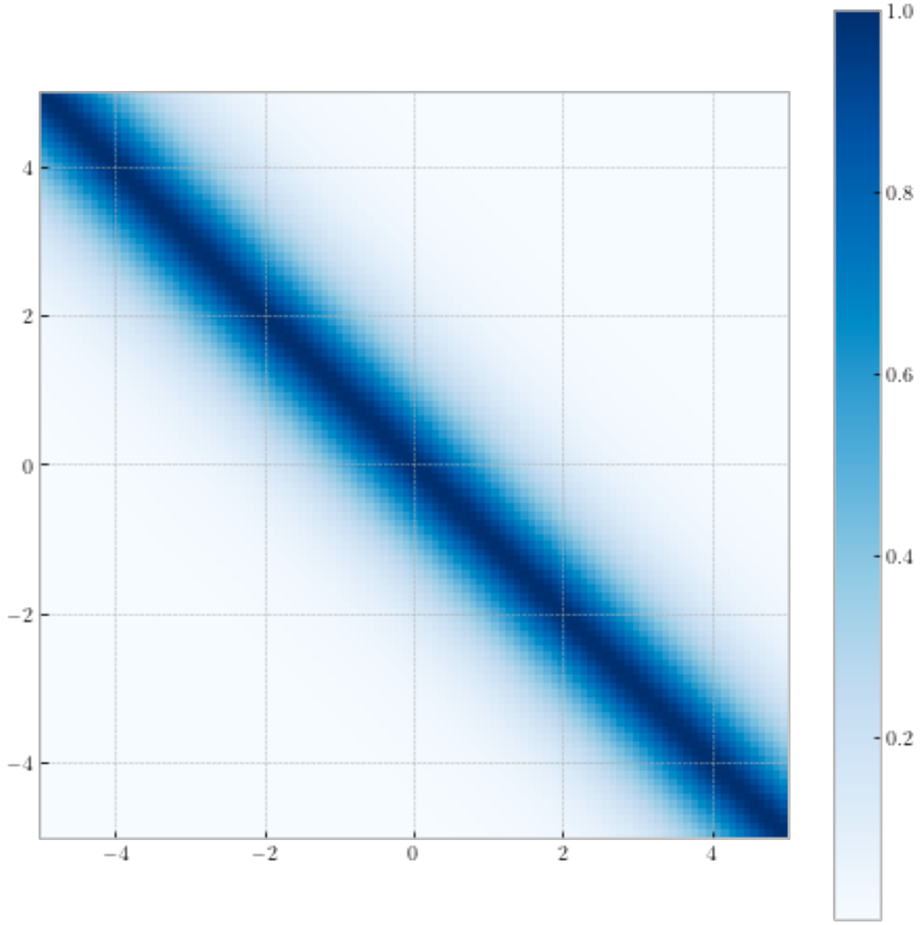
MATERN 3/2 KERNEL

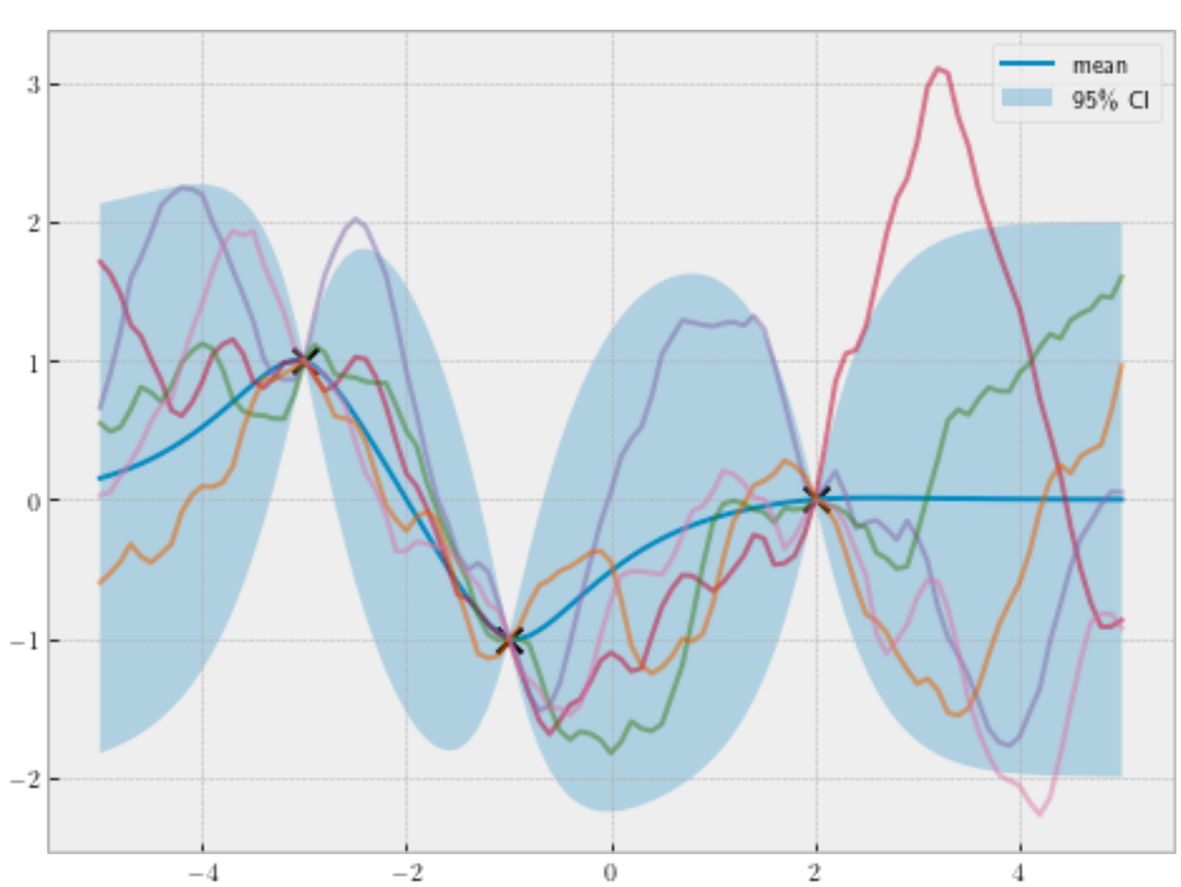
$$K(\mathrm{x}, \mathrm{x}') = \left(1 + \sqrt{3}d\right) \exp\left(-\sqrt{3}d\right)$$



MATERN 3/2 KERNEL

$$K(\mathrm{x}, \mathrm{x}') = \left(1 + \sqrt{3}d\right) \exp\left(-\sqrt{3}d\right)$$



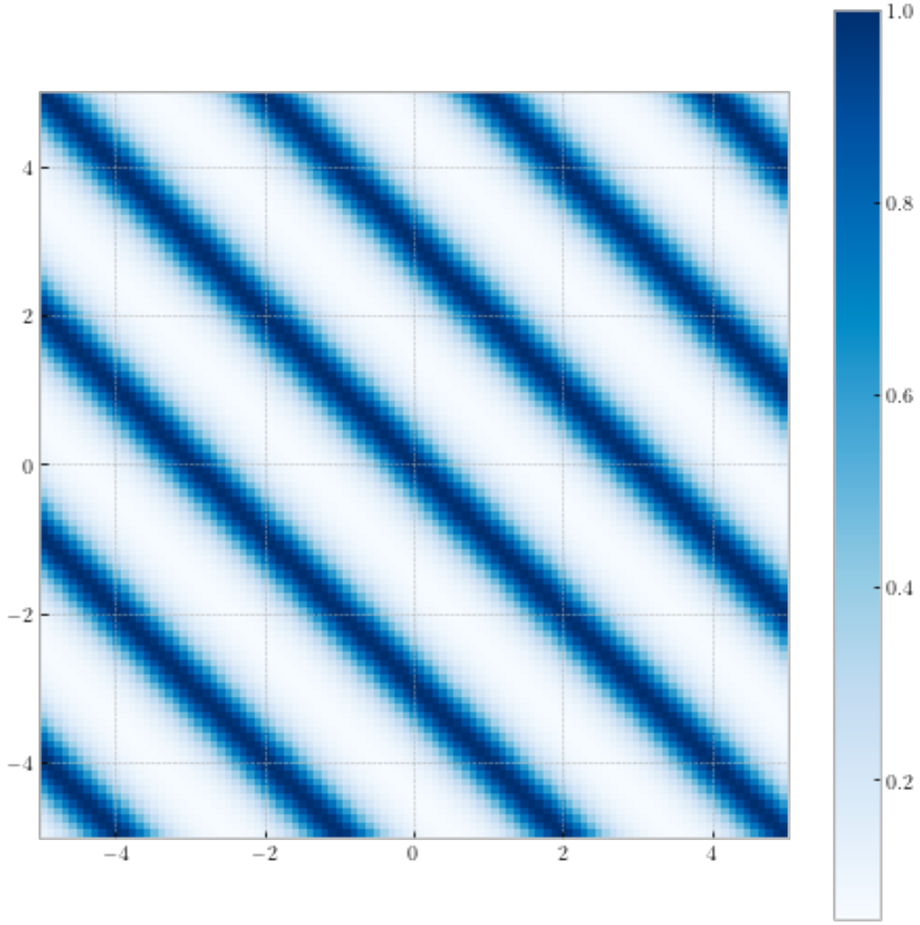


PERIODIC KERNEL

$$K(\mathrm{x}, \mathrm{x}') = \exp\left(-2 \sum_i \sin^2\left(\pi(x_i - x_i')\right) \right)$$

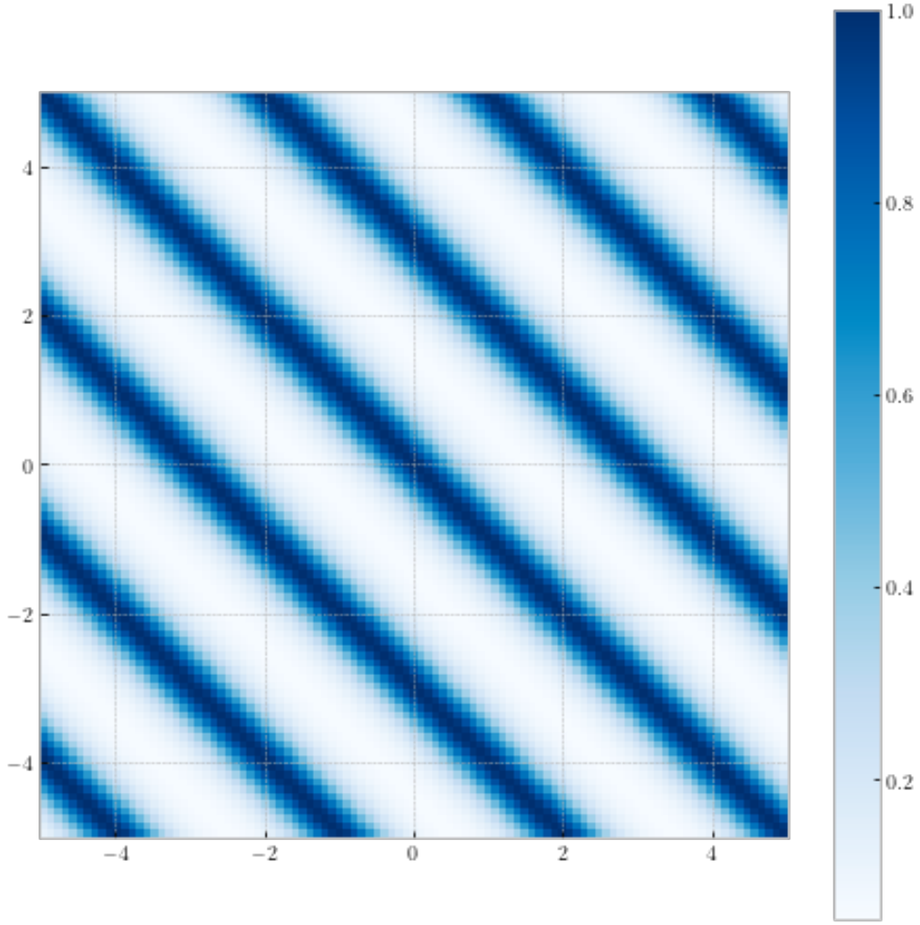
PERIODIC KERNEL

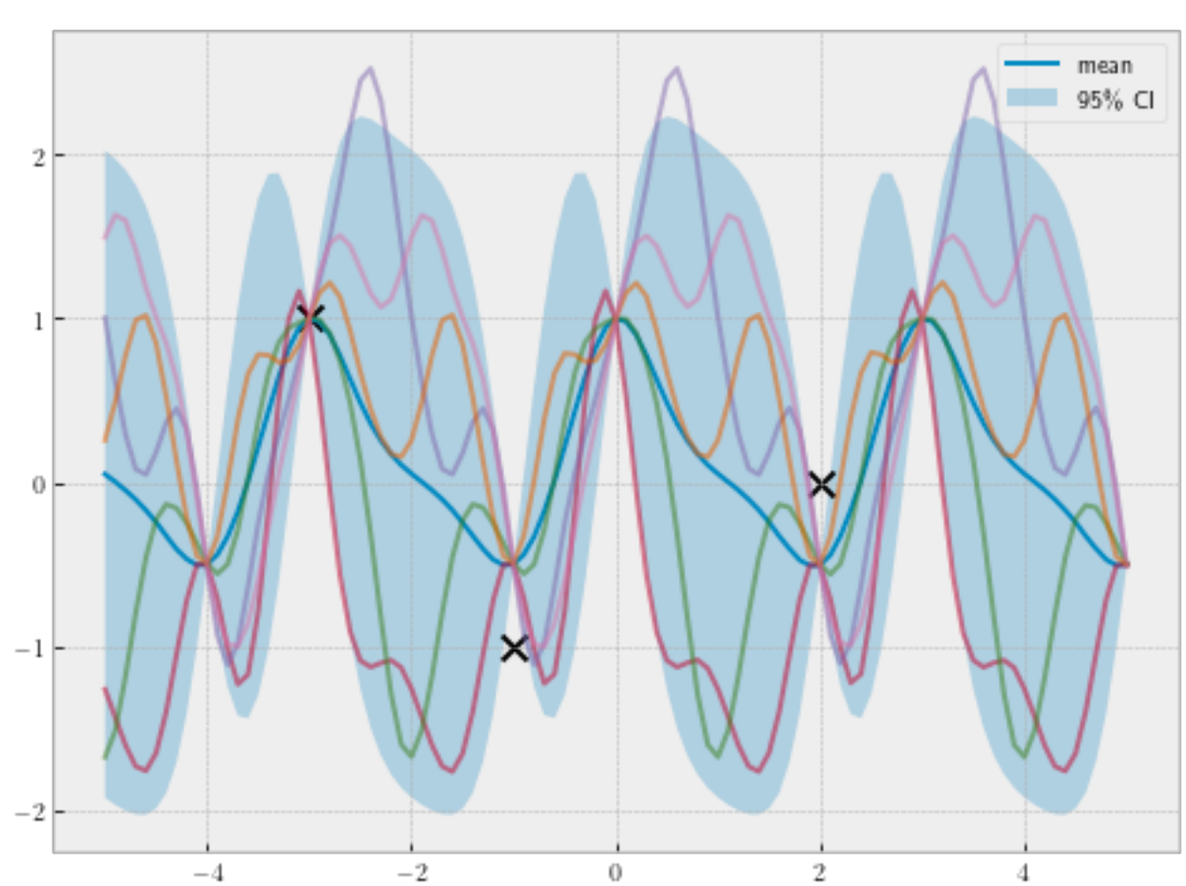
$$K(\mathrm{x}, \mathrm{x}') = \exp\left(-2\sum_i \sin^2\left(\pi(x_i - x_i')\right)\right)$$



PERIODIC KERNEL

$$K(\mathrm{x}, \mathrm{x}') = \exp\left(-2 \sum_i \sin^2\left(\pi(x_i - x_i')\right) \right)$$



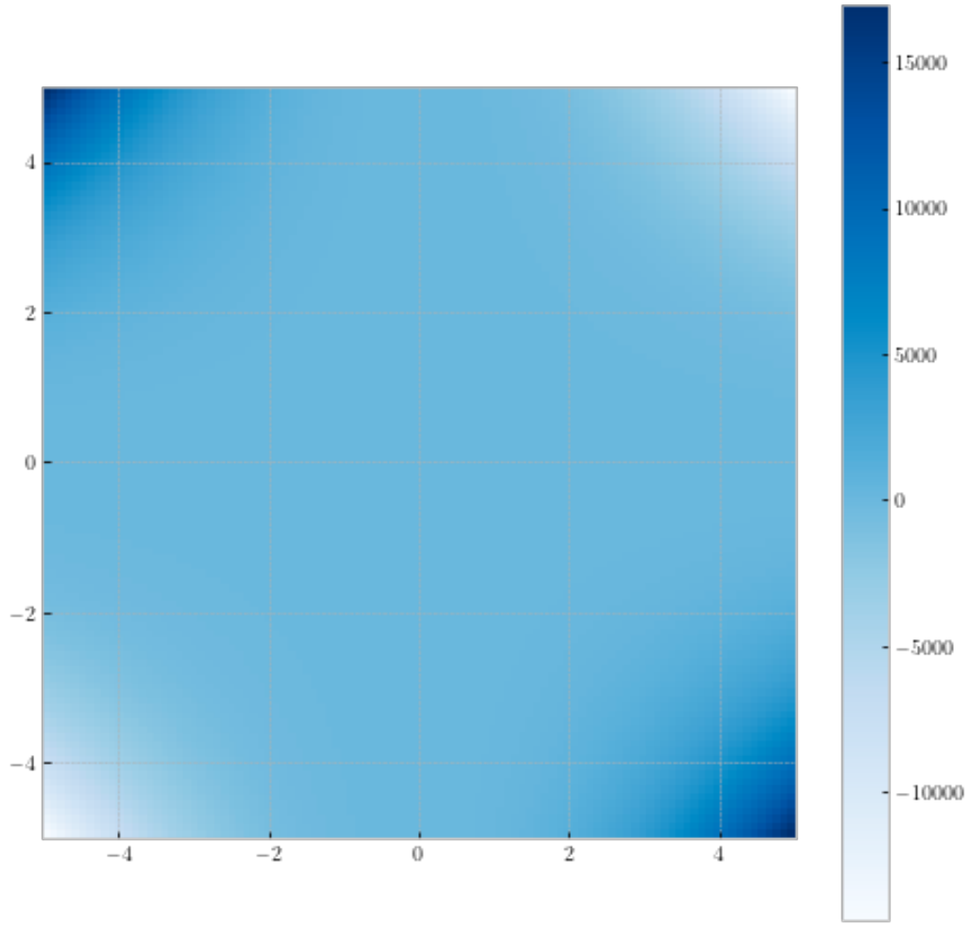


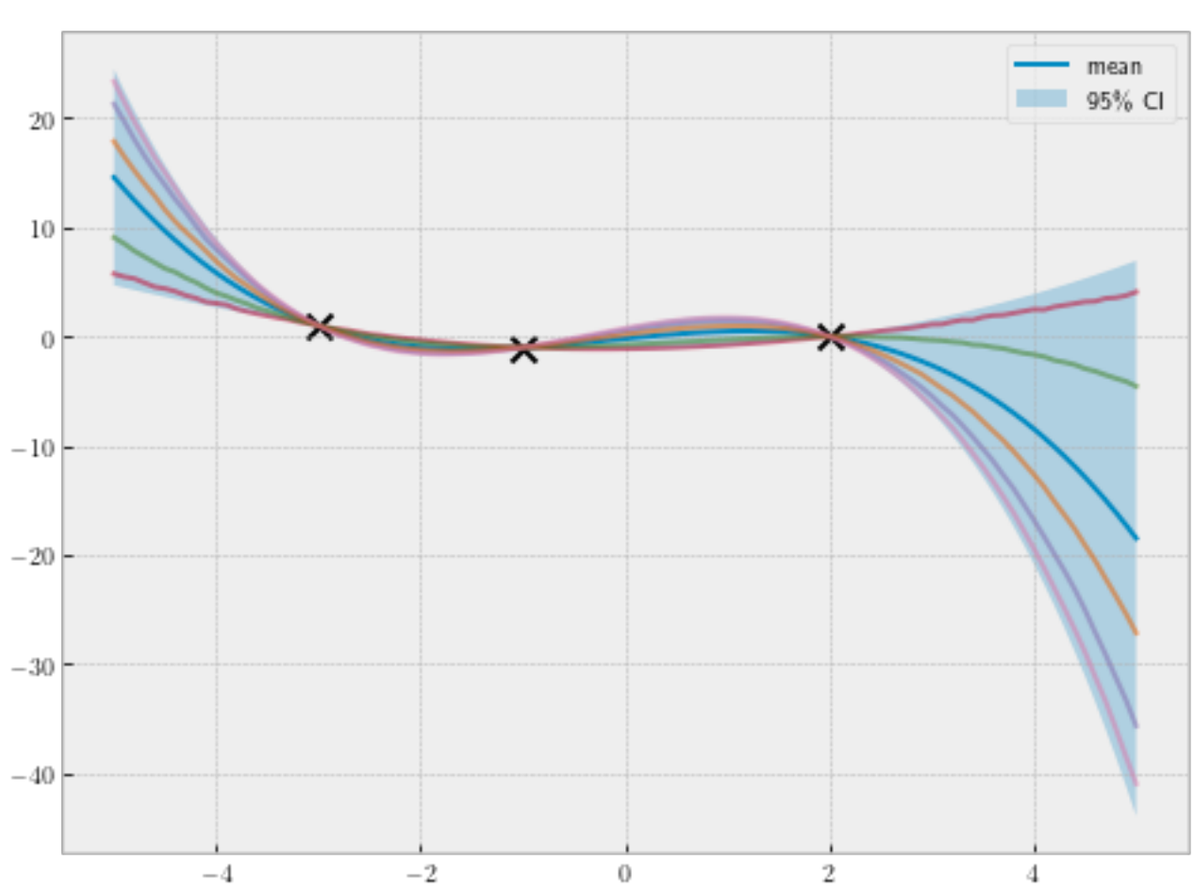
POLYNOMIAL KERNEL

$$K(\mathsf{x}, \mathsf{x}') = (\mathsf{x}^\top \mathsf{x}' + c)^d \text{ of degree } d$$

POLYNOMIAL KERNEL

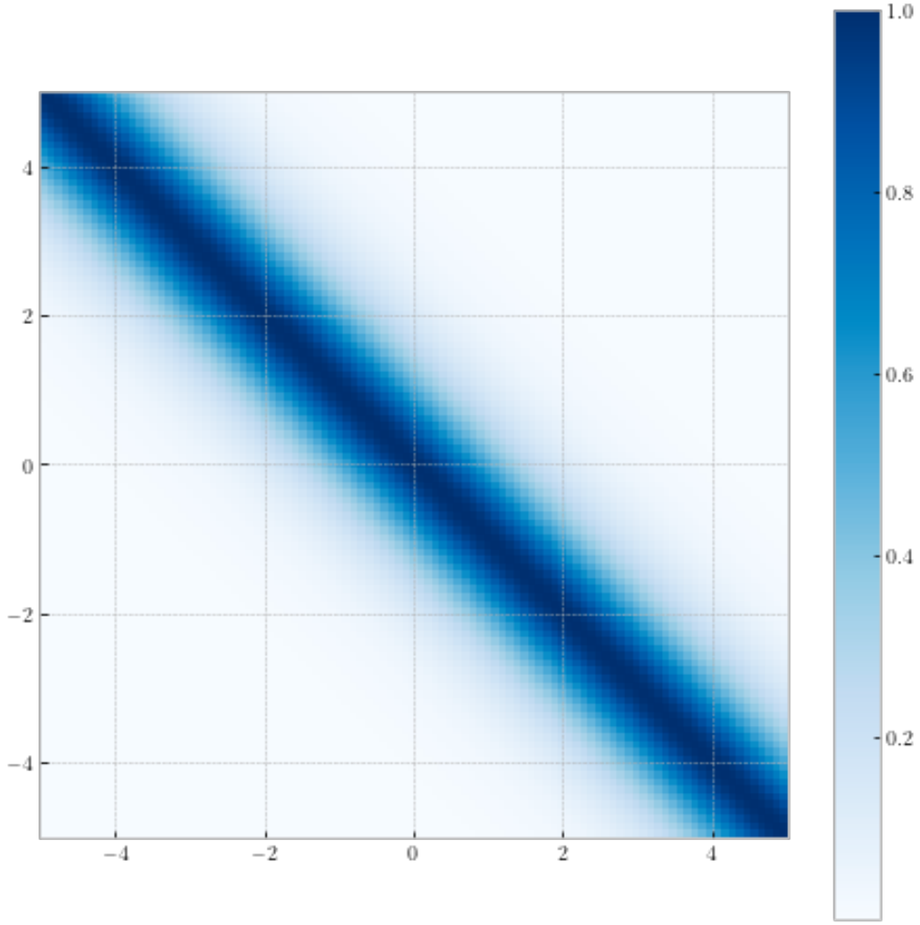
$$K(\mathsf{x}, \mathsf{x}') = (\mathsf{x}^\top \mathsf{x}' + c)^d \text{ of degree } d$$

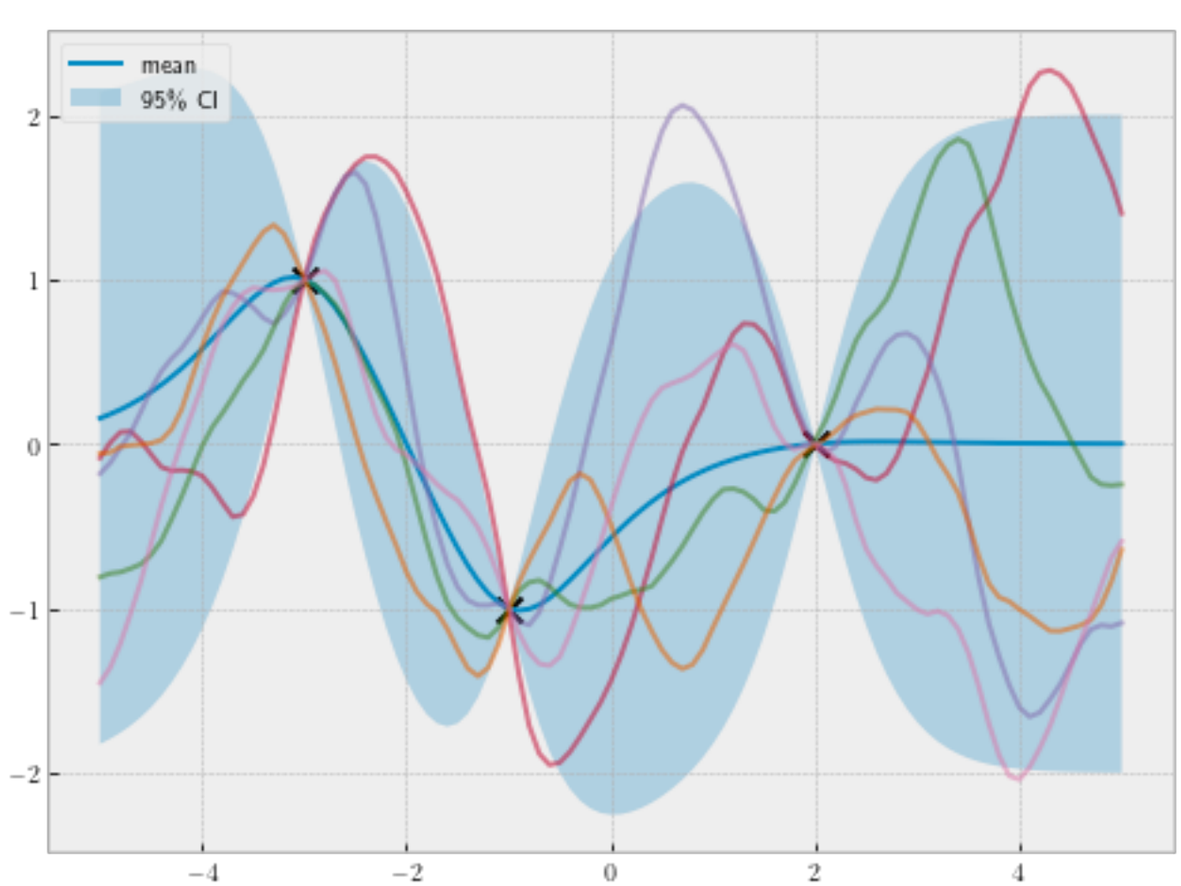




MATERN 5/2 KERNEL

$$K(\mathrm{x}, \mathrm{x}') = \left(1 + \sqrt{5}d + 5d^2\right) \exp\left(-\sqrt{5}d\right)$$





GP HYPERPARAMETERS

GP HYPERPARAMETERS

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

GP HYPERPARAMETERS

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

Length scale ℓ controls how "wiggly" the function is

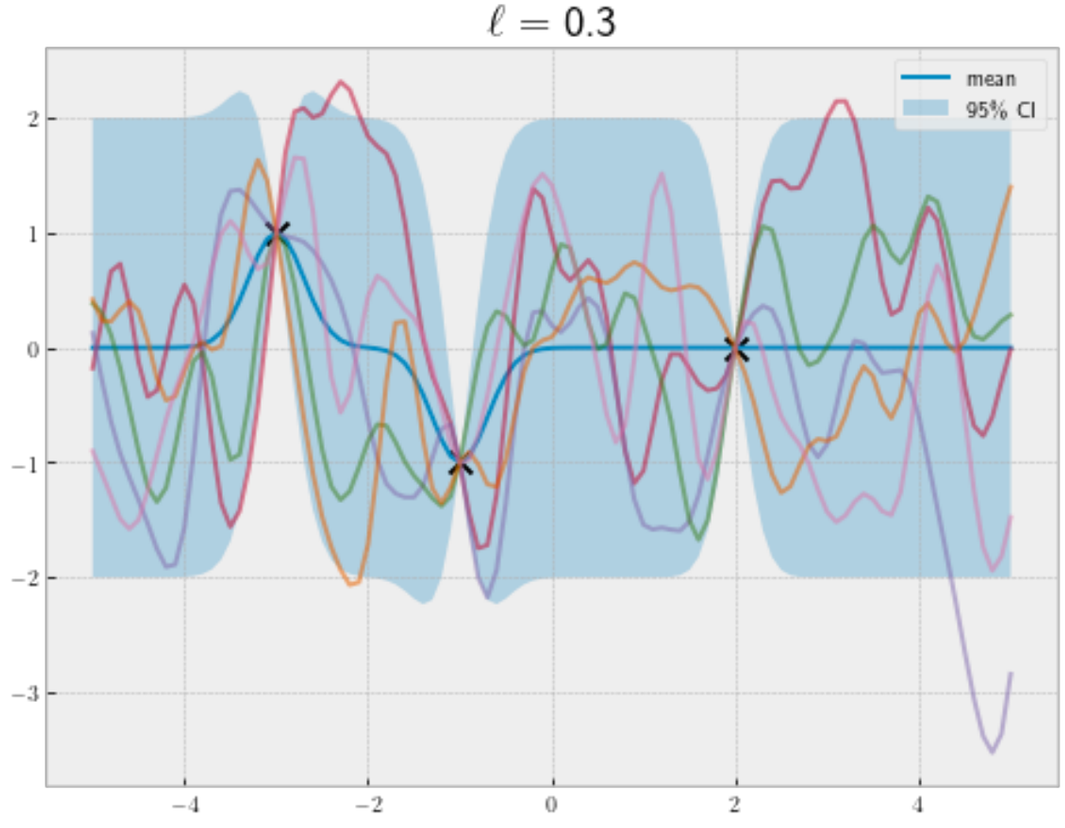
Output scale σ^2 controls the range of the function

LENGTH SCALE CONTROLS WIGGLINESS

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

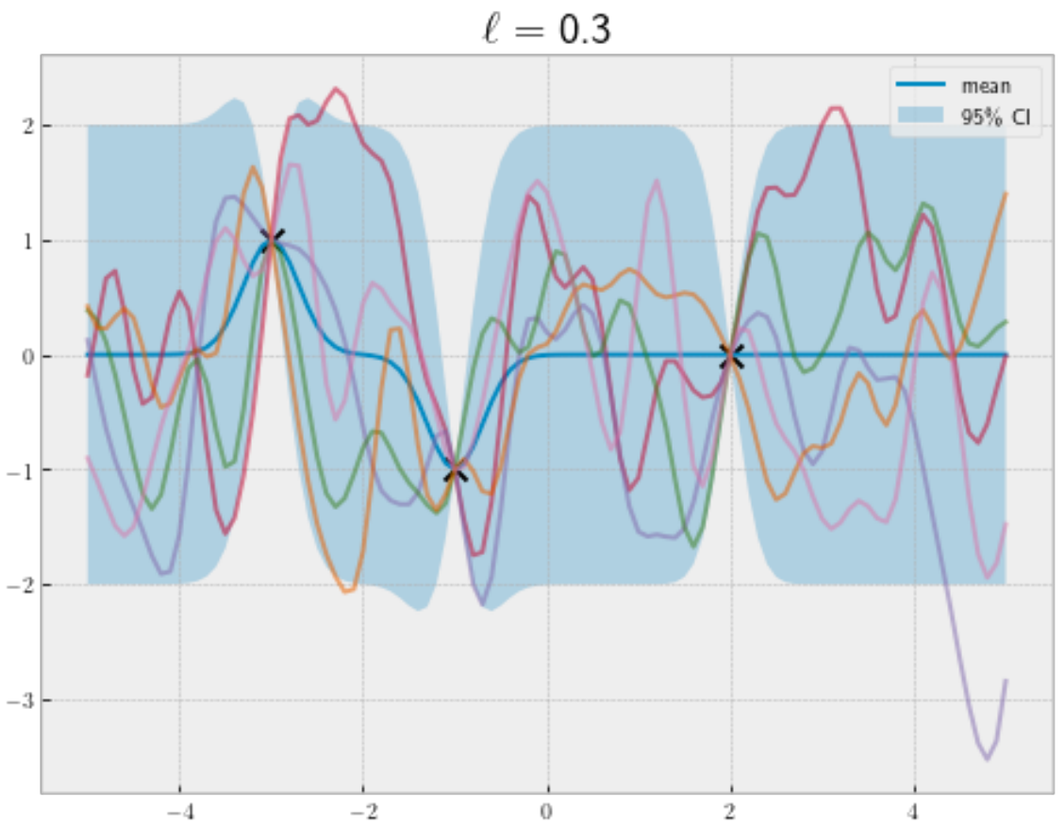
LENGTH SCALE CONTROLS WIGGLINESS

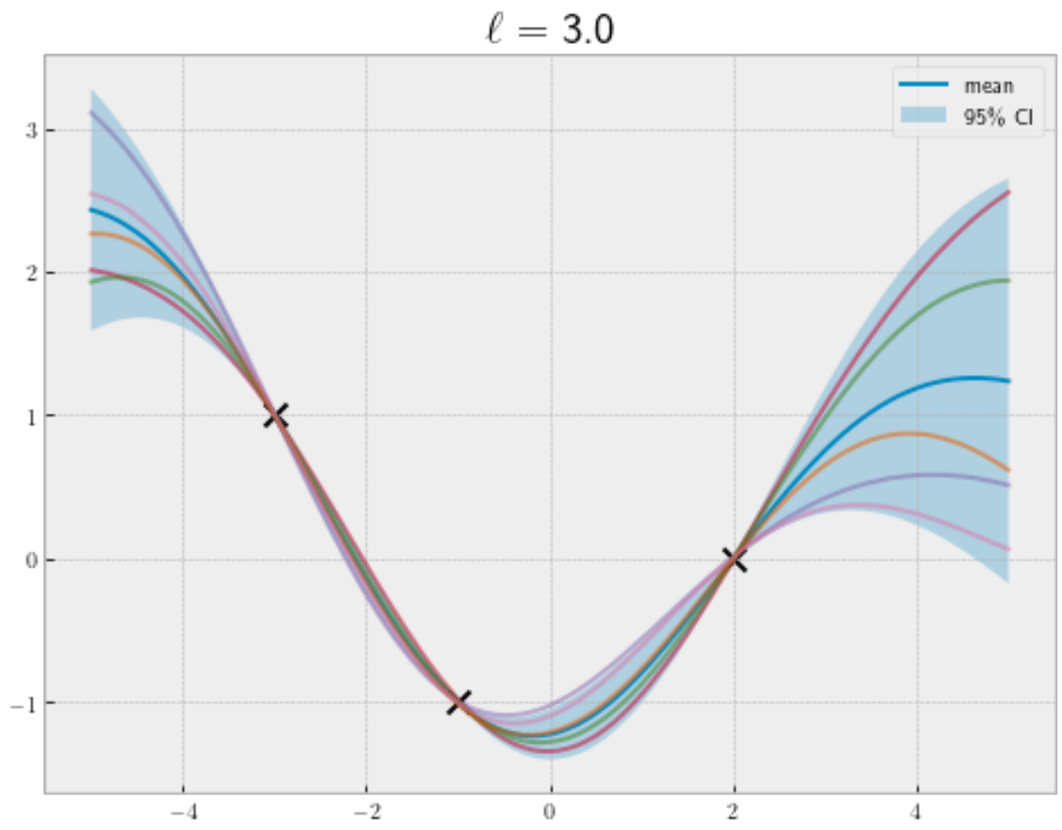
$$K(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\ell}\right)$$



LENGTH SCALE CONTROLS WIGGLINESS

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



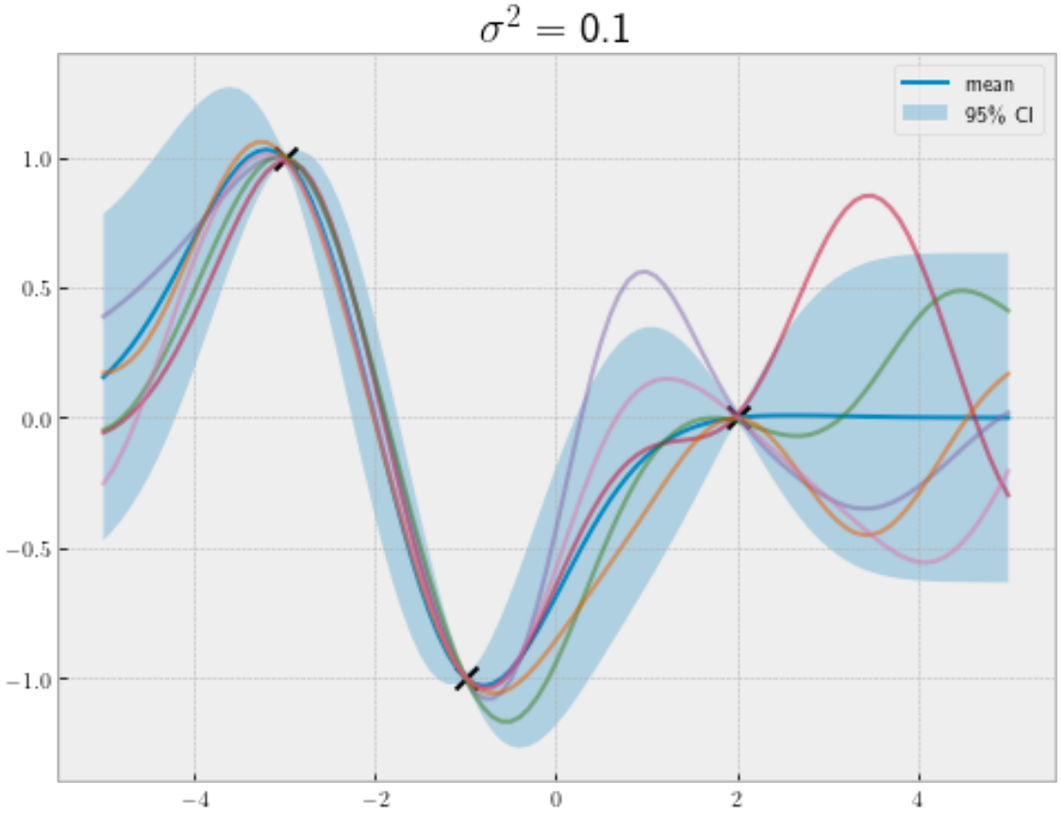


OUTPUT SCALE CONTROLS RANGE

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

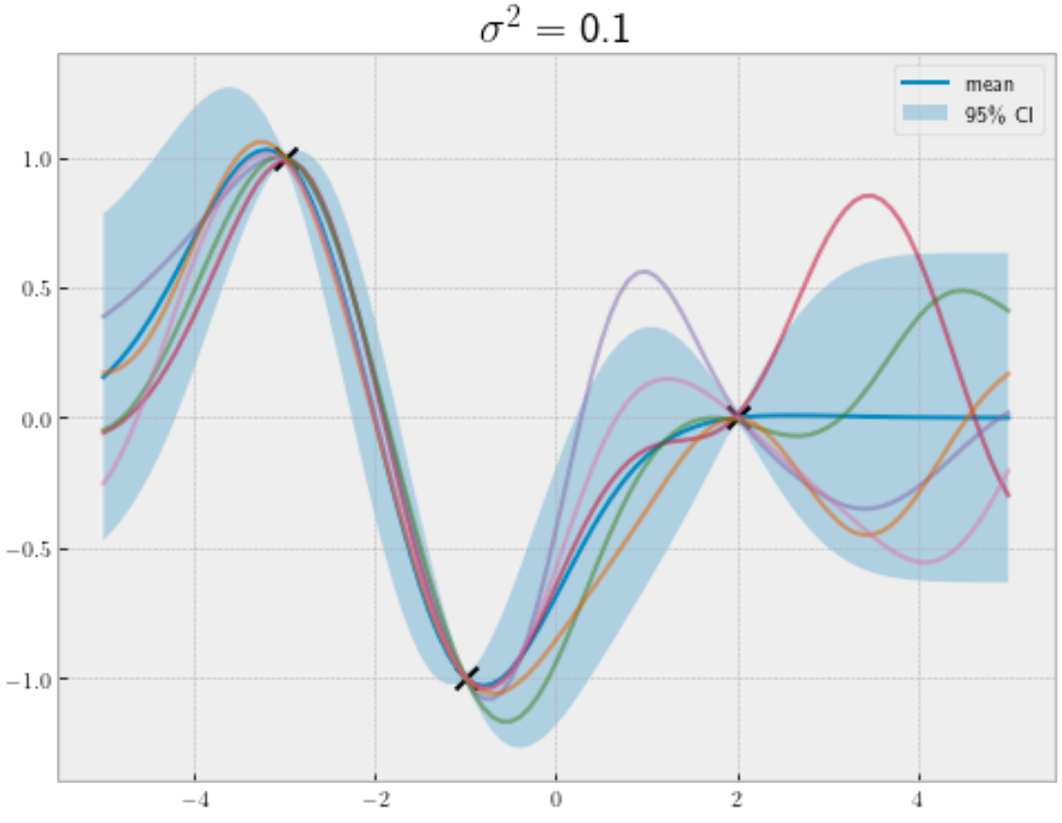
OUTPUT SCALE CONTROLS RANGE

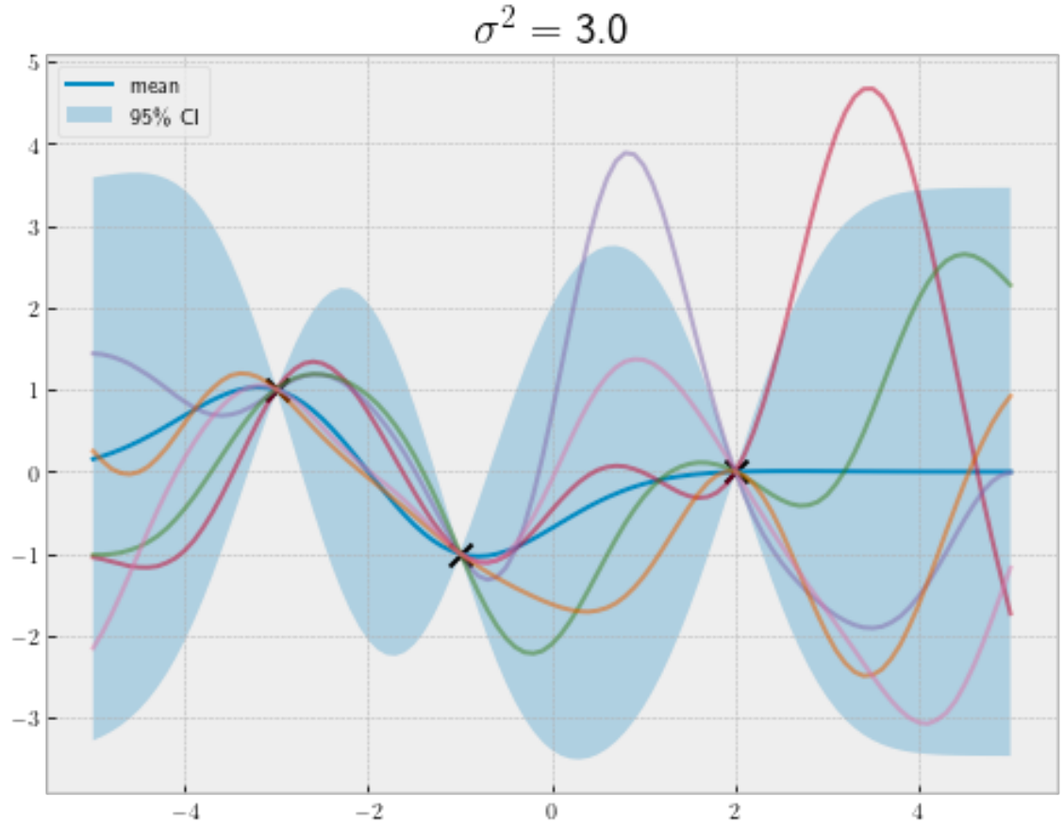
$$K(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\ell}\right)$$



OUTPUT SCALE CONTROLS RANGE

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





FINDING THE BEST HYPERPARAMETERS

FINDING THE BEST HYPERPARAMETERS

Given the hyperparameter vector θ, consider the negative

log marginal likelihood:

FINDING THE BEST HYPERPARAMETERS

Given the hyperparameter vector $\boldsymbol{\theta}$, consider the negative log marginal likelihood:

$$-\log p(\mathbf{y} \mid \boldsymbol{\theta}) = \frac{1}{2}\log\det K(X,X) + \frac{1}{2}\mathbf{y}^\top K^{-1}(X,X)\ \mathbf{y} + \frac{N}{2}\log(2\pi)$$

FINDING THE BEST HYPERPARAMETERS

Given the hyperparameter vector $\boldsymbol{\theta}$, consider the negative log marginal likelihood:

$$-\log p(\mathsf{y} \mid \boldsymbol{\theta}) = \frac{1}{2} \log \det K(X, X) + \frac{1}{2} \mathsf{y}^\top \, K^{-1}(X, X) \, \mathsf{y} + \frac{N}{2} \log(2\pi)$$

- Closed-form gradient w.r.t. elements in $\boldsymbol{\theta}$.

ONE LENGTH SCALE FOR EACH DIMENSION

ONE LENGTH SCALE FOR EACH DIMENSION

Automatic relevance determination (ARD) covariance function

ONE LENGTH SCALE FOR EACH DIMENSION

Automatic relevance determination (ARD) covariance function

$$K(\mathrm{x}, \mathrm{x}') = \sigma^2 \exp\left(-\frac{1}{2}\sum_d \frac{(x_d - x'_d)^2}{\ell_d}\right)$$

ONE LENGTH SCALE FOR EACH DIMENSION

Automatic relevance determination (ARD) covariance function

$$K(\mathrm{x}, \mathrm{x}') = \sigma^2 \exp\left(-\frac{1}{2}\sum_d \frac{(x_d - x'_d)^2}{\ell_d}\right)$$

Length scale ℓ_d for dimension d determines how relevant the dimension is

ONE LENGTH SCALE FOR EACH DIMENSION

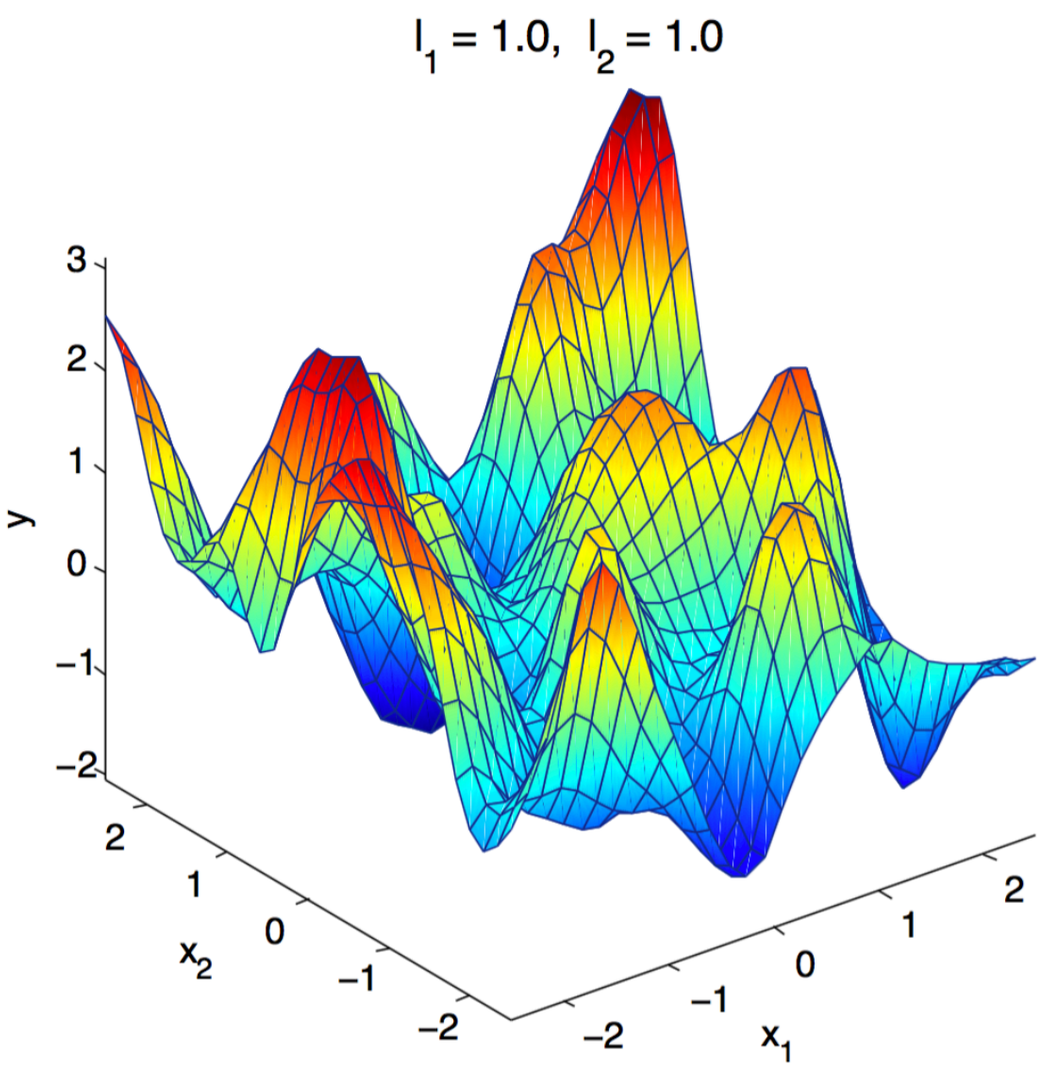
Automatic relevance determination (ARD) covariance function

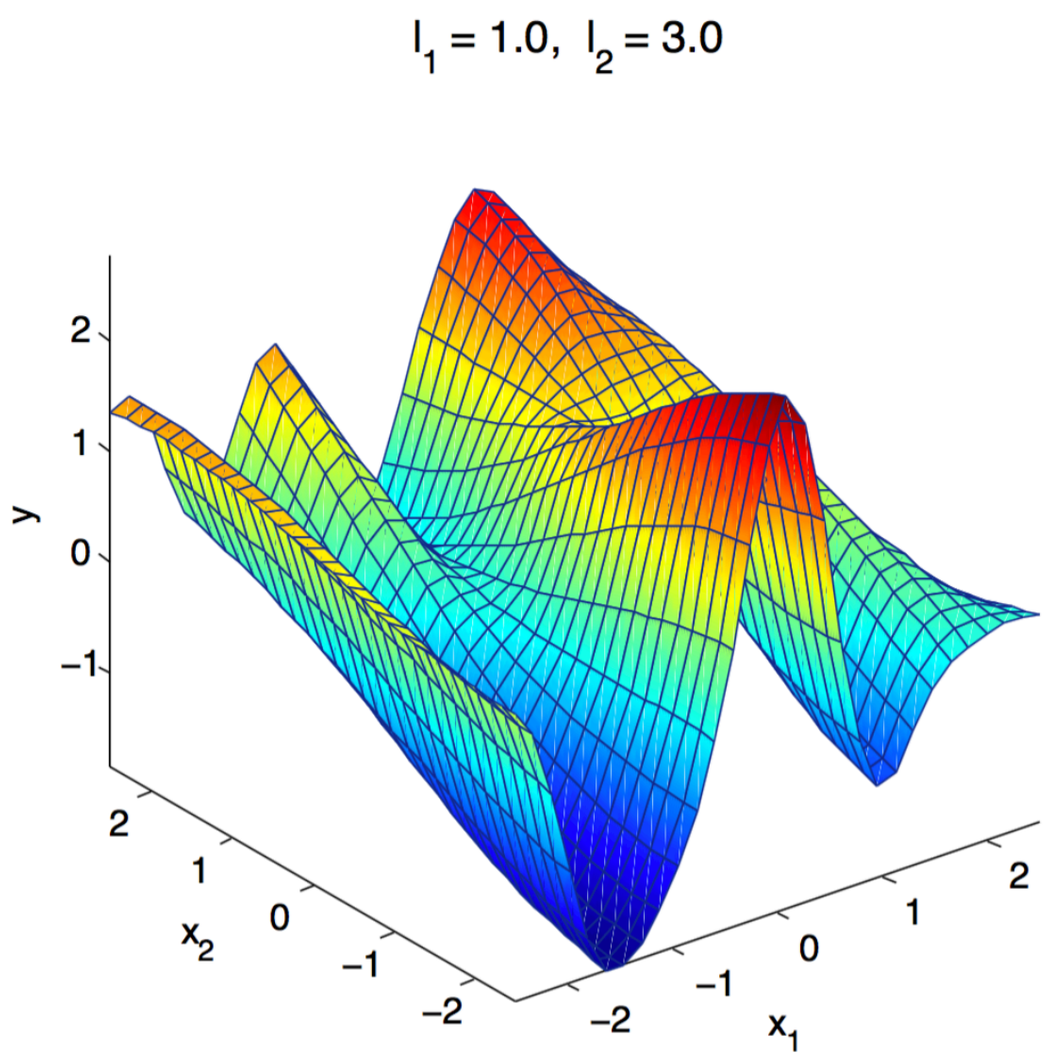
$$K(\mathrm{x}, \mathrm{x}') = \sigma^2 \exp\left(-\frac{1}{2}\sum_d \frac{(x_d - x'_d)^2}{\ell_d}\right)$$

Length scale ℓ_d for dimension d determines how relevant the dimension is

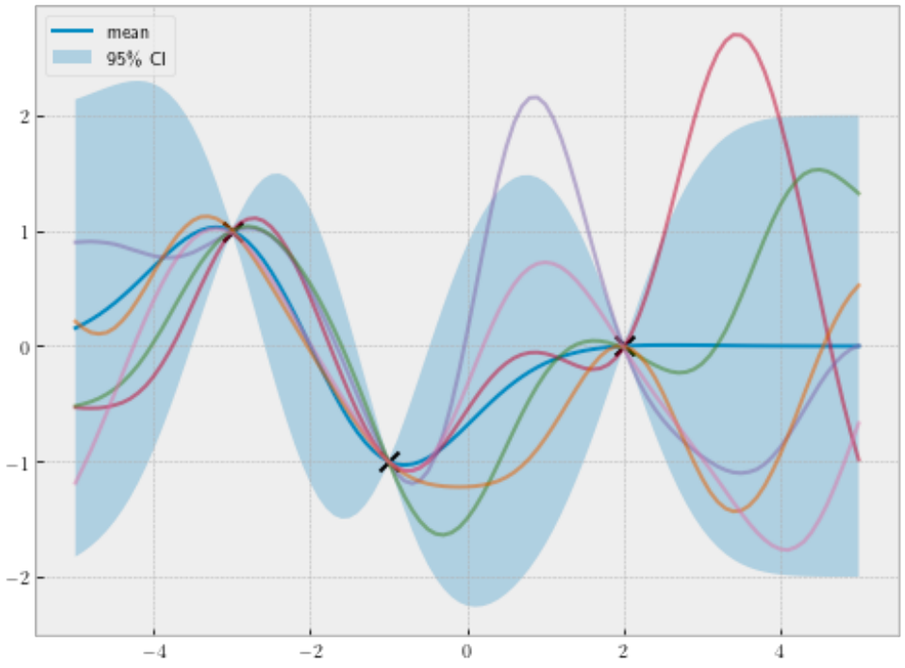
- As ℓ_d gets **larger**, dimension d becomes **less relevant**

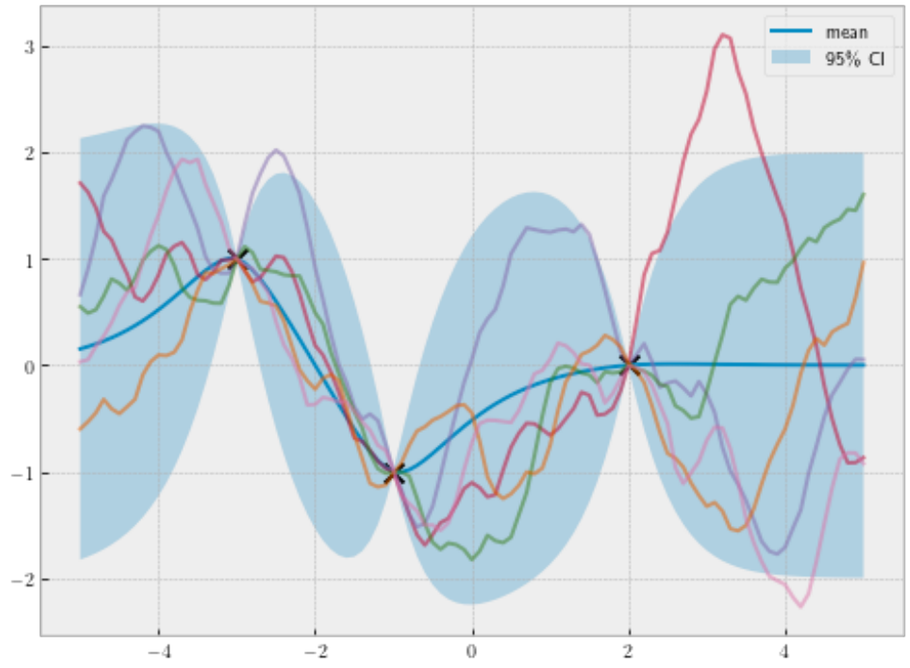
ARD IN ACTION

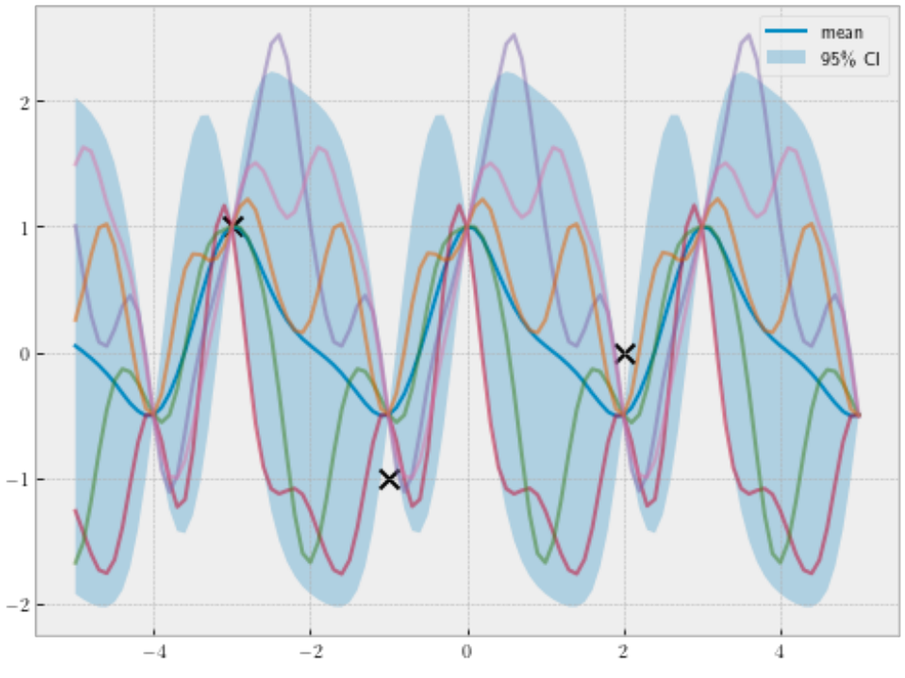


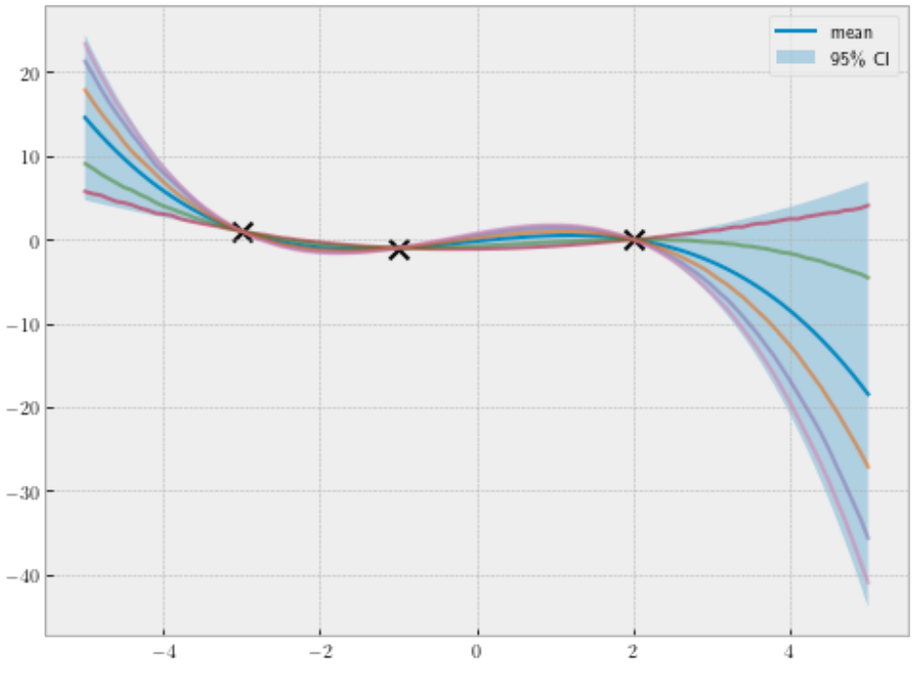


GP KERNELS









MODELING MULTIPLE TRENDS SIMULTANEOUSLY

CO_2 concentration

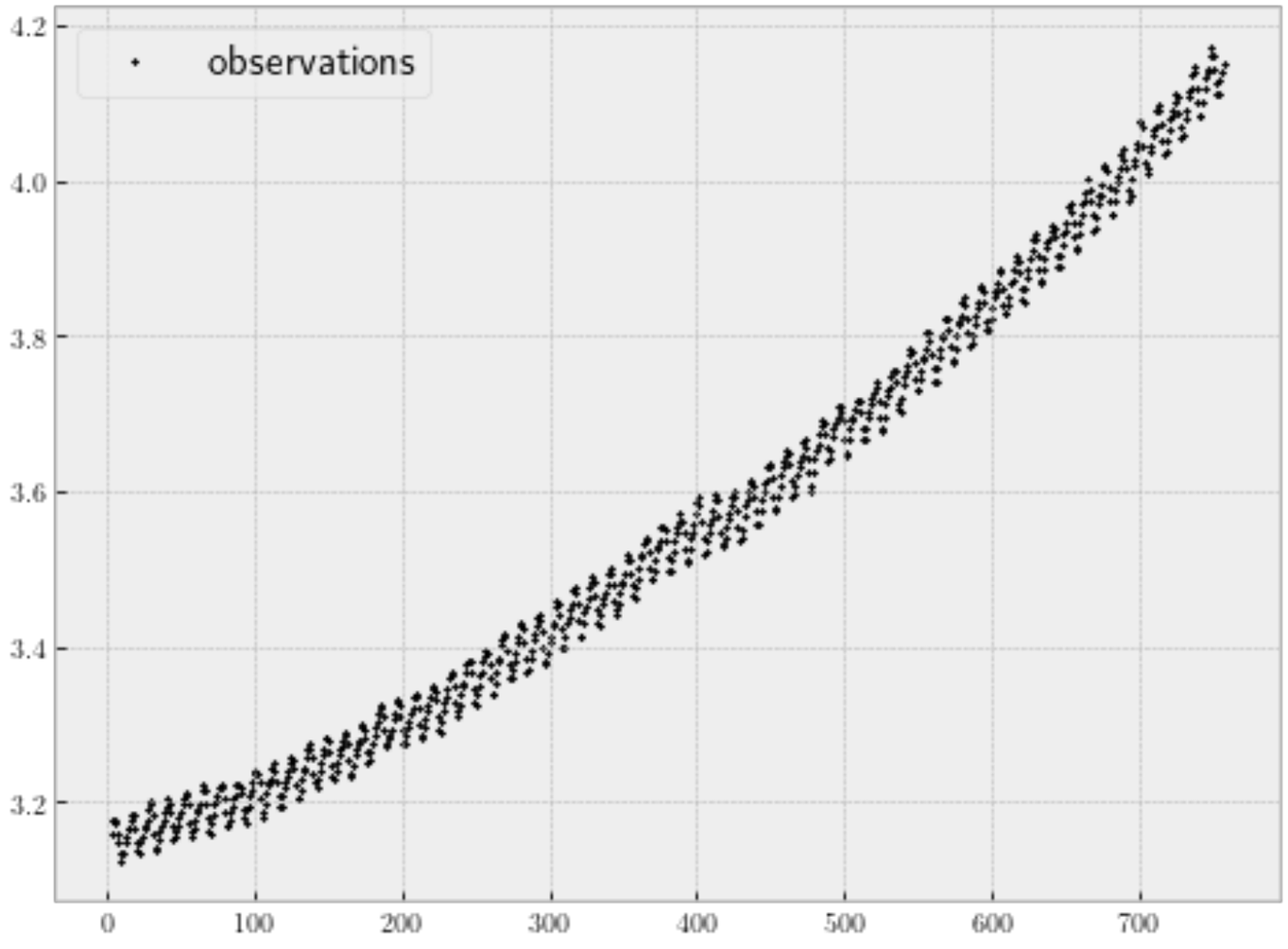
by **month**:

- Long-term rising trend
- Seasonal changes

MODELING MULTIPLE TRENDS SIMULTANEOUSLY

CO_2 concentration by **month**:

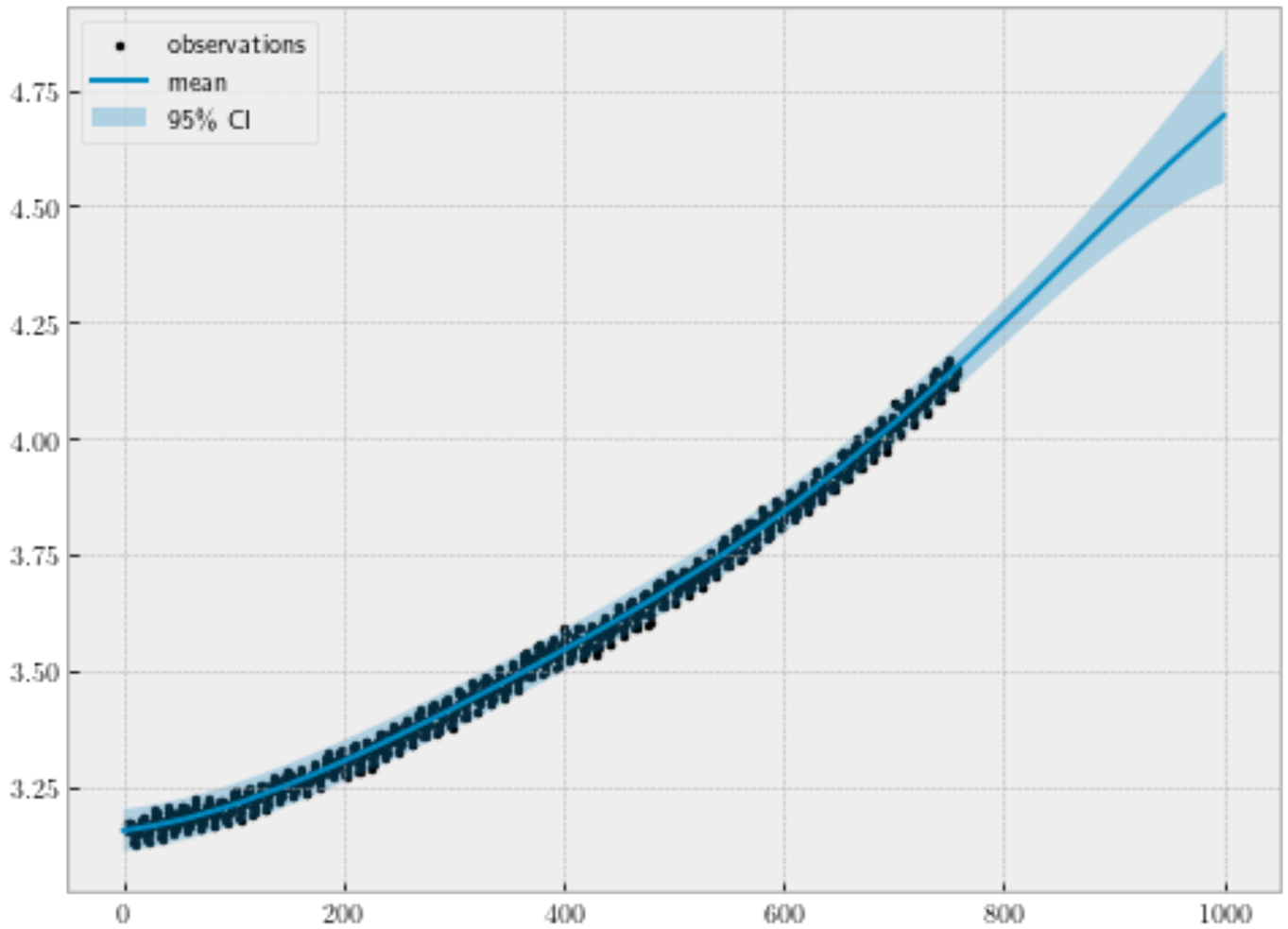
- Long-term rising trend
- Seasonal changes



MODELING MULTIPLE TRENDS SIMULTANEOUSLY

CO_2 concentration by **month**:

- Long-term rising trend
- Seasonal changes

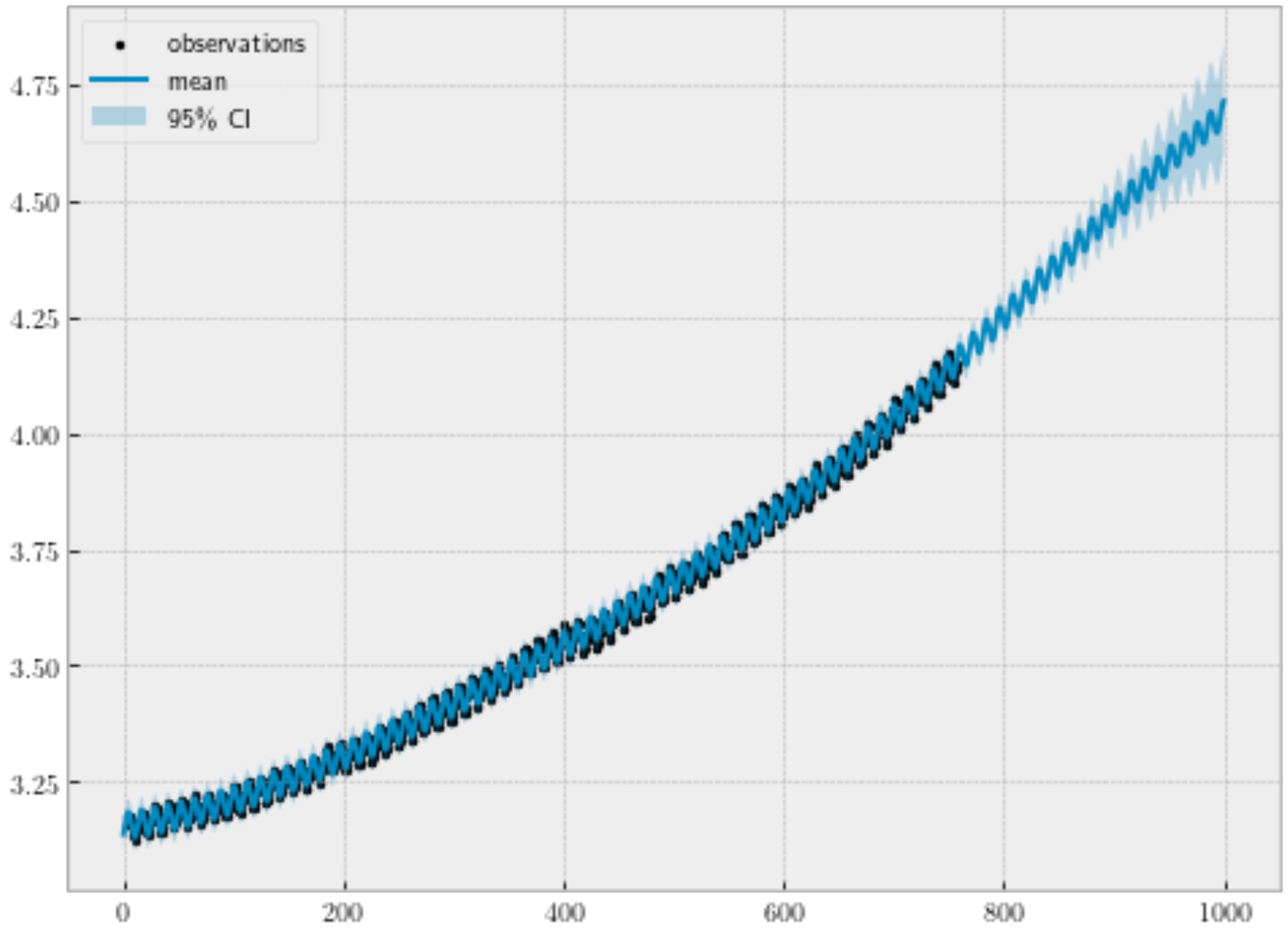


$$K(\mathrm{x}, \mathrm{x}') = \sigma^2 \exp\left(-\frac{\|\mathrm{x} - \mathrm{x}'\|^2}{2\ell}\right) + \sigma_n^2$$

MODELING MULTIPLE TRENDS SIMULTANEOUSLY

CO_2 concentration by **month**:

- Long-term rising trend
- Seasonal changes



$$K(\mathrm{x}, \mathrm{x}') = \sigma_1^2 \exp\left(-\frac{\|\mathrm{x}-\mathrm{x}'\|^2}{2\ell}\right) + \sigma_2^2 \cos\left(\pi\frac{\|\mathrm{x}-\mathrm{x}'\|^2}{p}\right) + \sigma_n^2$$

DON'T TRY THIS IN NC!

KERNEL CONSTRUCTION

KERNEL CONSTRUCTION

- Kernel grammar

KERNEL CONSTRUCTION

- Kernel grammar
  - Adding, multiplying, exponentiating, etc.

KERNEL CONSTRUCTION

- Kernel grammar
  - Adding, multiplying, exponentiating, etc.
- How to find the **best** “formula”?: AutoML

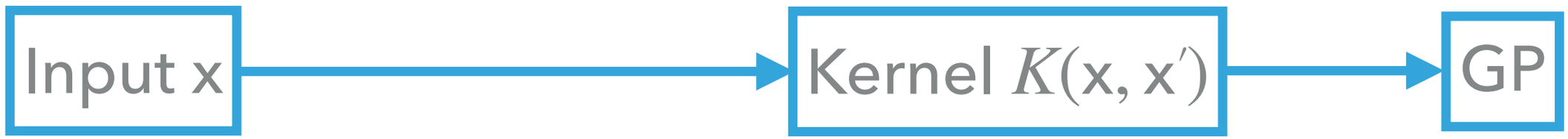
KERNEL CONSTRUCTION

- Kernel grammar
  - Adding, multiplying, exponentiating, etc.
    - How to find the **best** “formula”?: AutoML
- Deep kernel learning

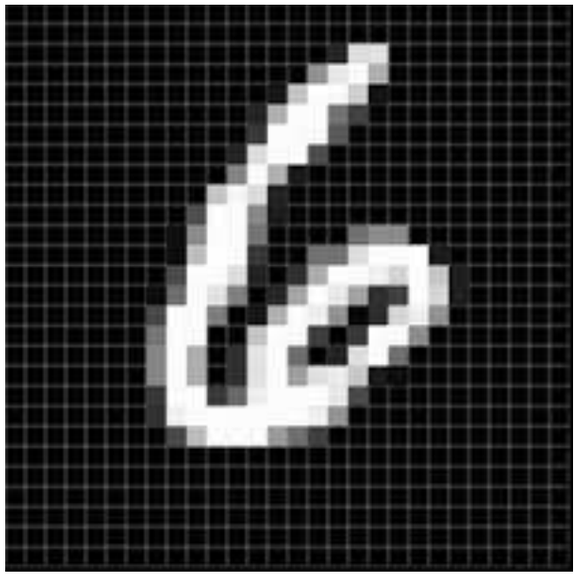
KERNEL CONSTRUCTION

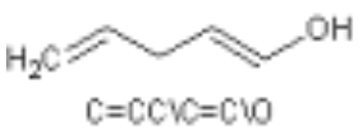
- Kernel grammar
  - Adding, multiplying, exponentiating, etc.
    - How to find the **best** “formula”?: AutoML
- Deep kernel learning
  - Flexible, good for **structured** data

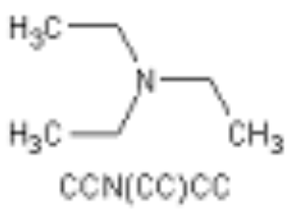
DEEP KERNEL LEARNING

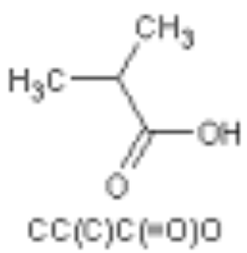


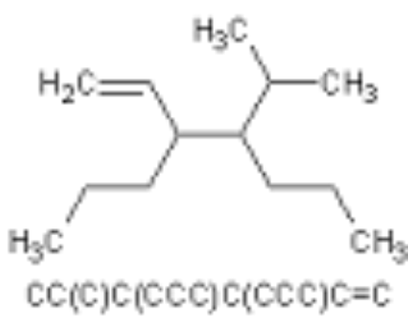
DEEP KERNEL LEARNING

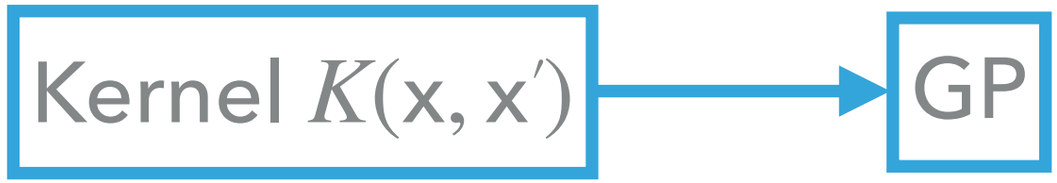




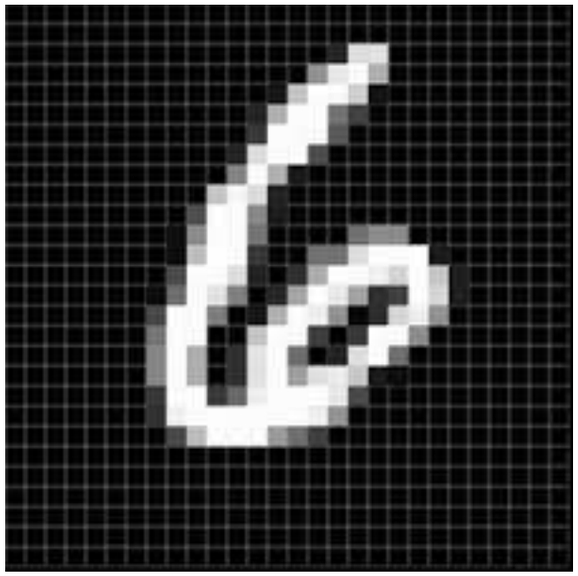


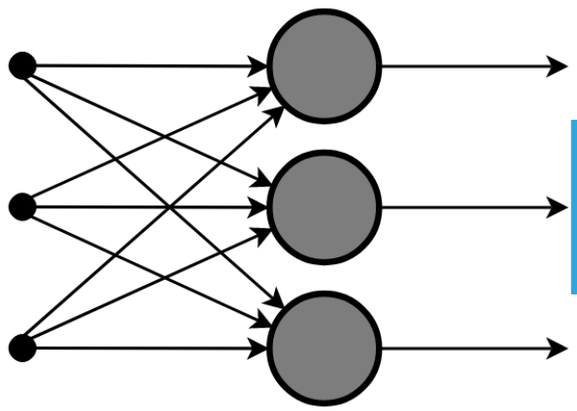


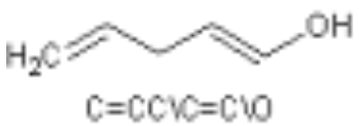


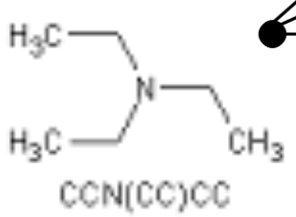


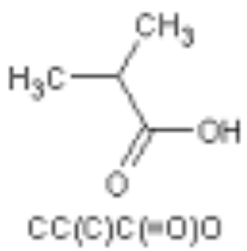
DEEP KERNEL LEARNING

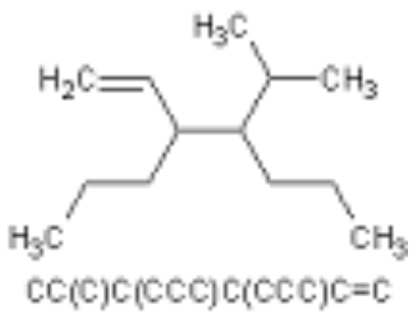












BAYESIAN OPTIMIZATION

BAYESIAN OPTIMIZATION

Expensive, **blackbox** optimization problems are common

BAYESIAN OPTIMIZATION

Expensive, **blackbox** optimization problems are common

- No known functional form

BAYESIAN OPTIMIZATION

Expensive, **blackbox** optimization problems are common

- No known functional form
- Cost associated with querying the function

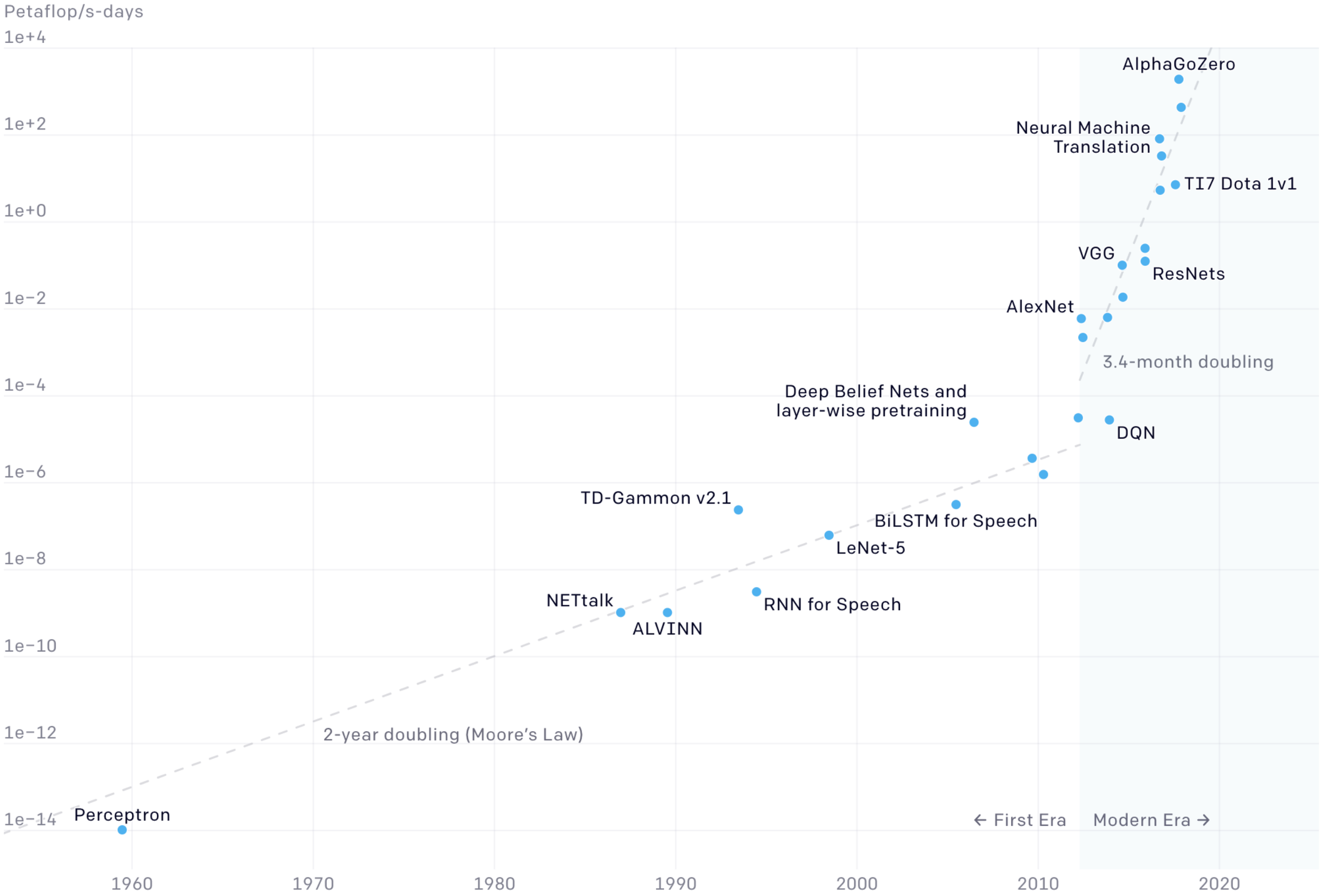
BAYESIAN OPTIMIZATION

Expensive, **blackbox** optimization problems are common

- No known functional form
- Cost associated with querying the function
- No derivative information

BAYESIAN OPTIMIZATION EXAMPLES

Hyperparameter tuning, user's preference optimization, drug discovery, etc.


Petaflop/s-days
1e+4
1e+2
1e+0
1e−2
1e−4
1e−6
1e−8
1e−10
1e−12
1e−14
AlphaGoZero
Neural Machine Translation
TI7 Dota 1v1
VGG
ResNets
AlexNet
3.4-month doubling
Deep Belief Nets and layer-wise pretraining
DQN
TD-Gammon v2.1
BiLSTM for Speech
LeNet-5
NETtalk
ALVINN
RNN for Speech
2-year doubling (Moore's Law)
Perceptron
← First Era
Modern Era →
1960
1970
1980
1990
2000
2010
2020

Bayesian Optimization for a Better Dessert

Greg Kochanski, Daniel Golovin, John Karro, Benjamin Solnik,
Subhodeep Moitra, and D. Sculley
{gpk, dgg, karro, bsolnik, smoitra, dsculley}@google.com; Google Brain Team

Abstract

We present a case study on applying Bayesian Optimization to a complex real-world system; our challenge was to optimize chocolate chip cookies. The process was a mixed-initiative system where both human chefs, human raters, and a machine optimizer participated in 144 experiments. This process resulted in highly rated cookies that deviated from expectations in some surprising ways – much less sugar in California, and cayenne in Pittsburgh. Our experience highlights the importance of incorporating domain expertise and the value of transfer learning approaches.

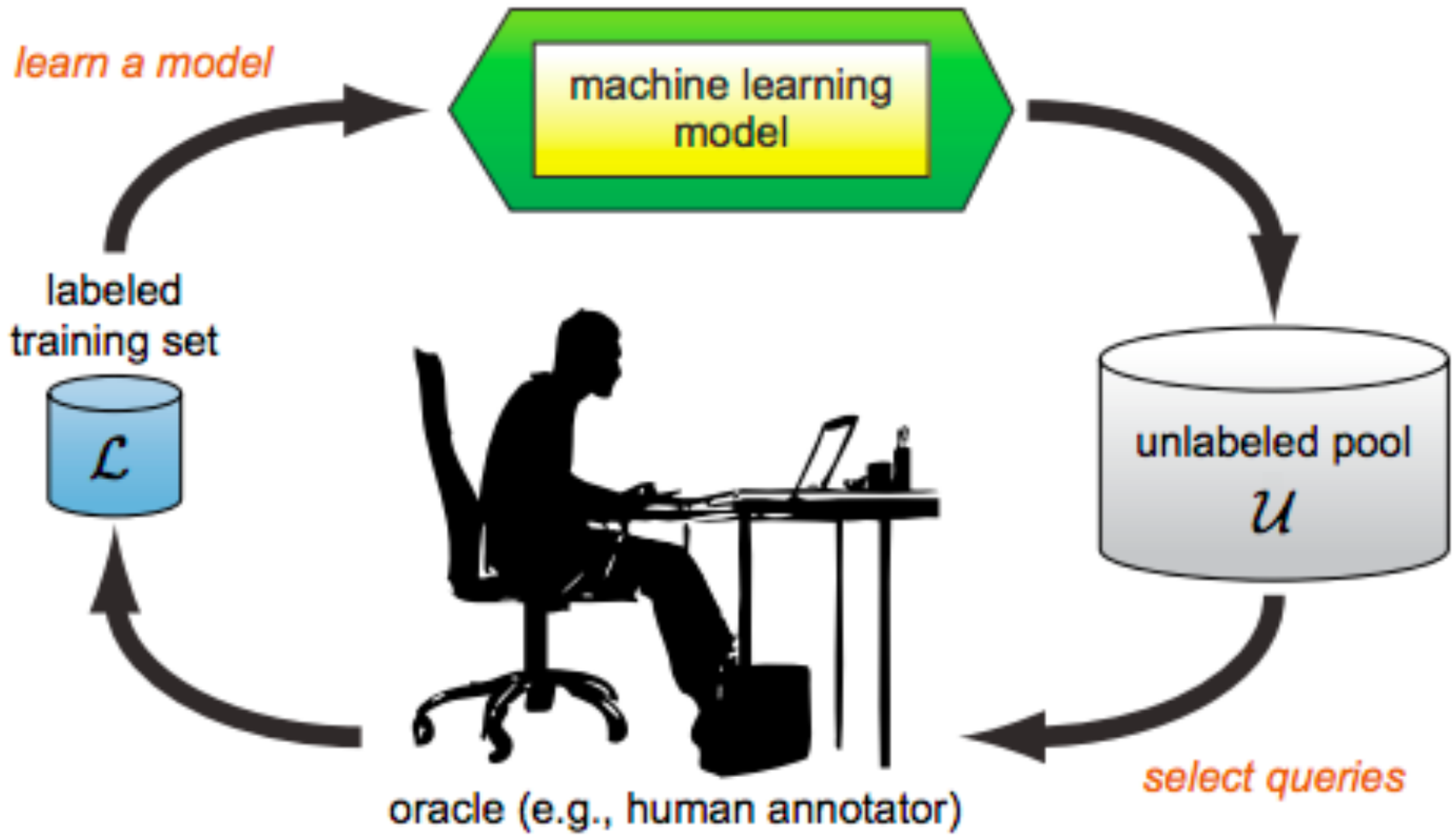
1 Introduction: The Challenge of Chocolate-Chip Cookies

Bayesian Optimization and black-box optimization are used extensively to optimize hyperparameters in machine learning (e.g. [13, 11, 3]) but less so outside that area, and even less so in fields like the culinary arts. We conjecture that the primary barrier to adoption is not technical, but rather cultural and educational. Just as it took years for application communities to frame tasks as supervised learning problems, it likewise takes time to recognize when black–box optimization can provide value. We seek to accelerate this process of cross-disciplinary adoption by creating a challenge that would help practitioners across disciplines recognize problems suitable for black-box optimization in their own settings.

The challenge was to optimize the recipe for chocolate chip cookies. This task highlights key qualities of problems well suited to Bayesian Optimization. The number of tunable parameters is relatively small (e.g. amounts of flour, sugar, etc; see e.g. Table 2). The cost of each experimental iteration is relatively high, requiring manual labor to mix and bake, then taste and report scores. And the problem is familiar enough so that people outside the optimization community can savor the complexities of the challenge and the resulting solutions.

In the remainder of this paper we will lay out a case study of our experience, including methods used, complications that arose in practice, and a summation of lessons learned.

THE ACTIVE LEARNING LOOP



TWENTY QUESTIONS