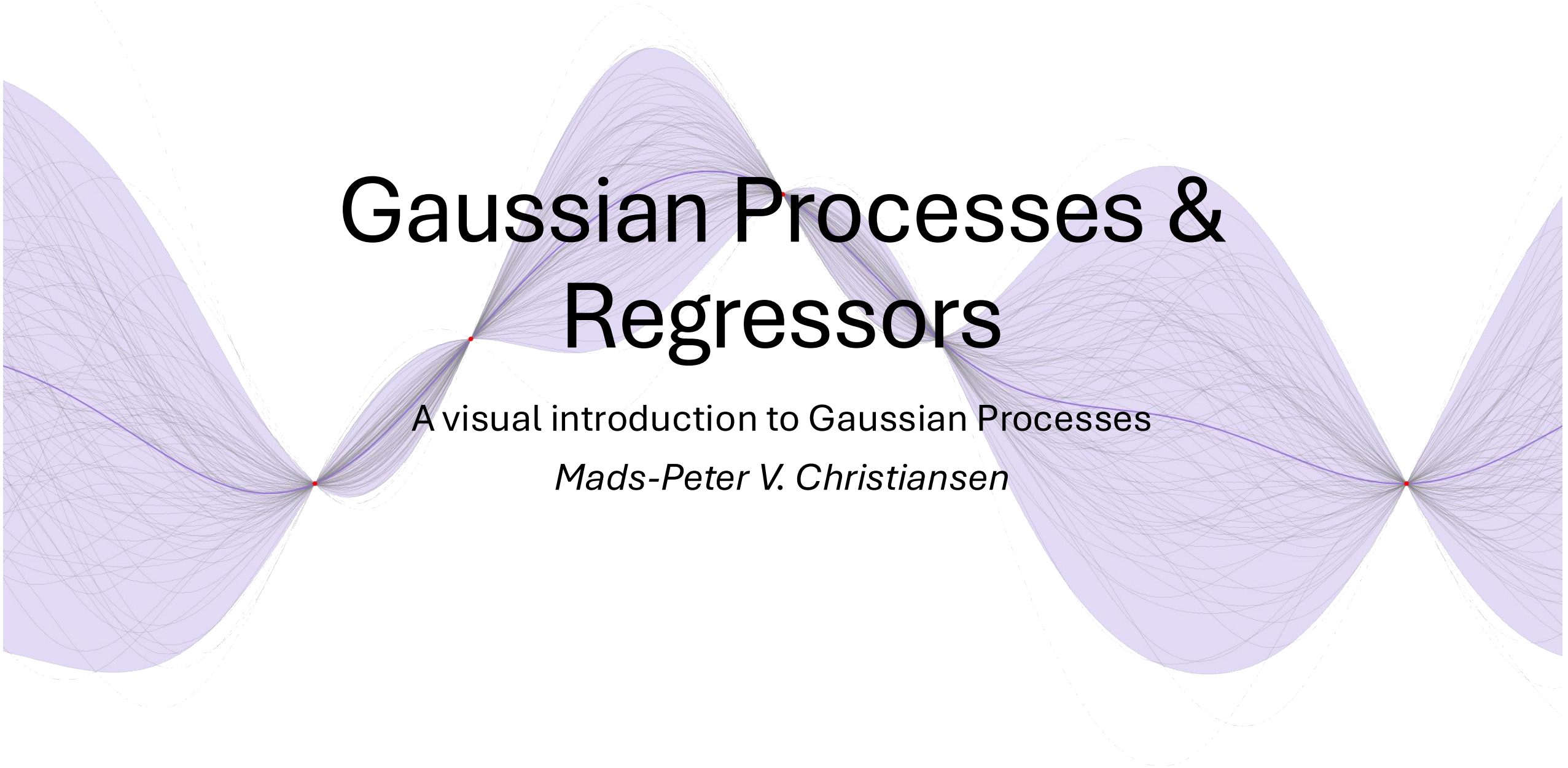


Gaussian Processes & Regressors

A visual introduction to Gaussian Processes

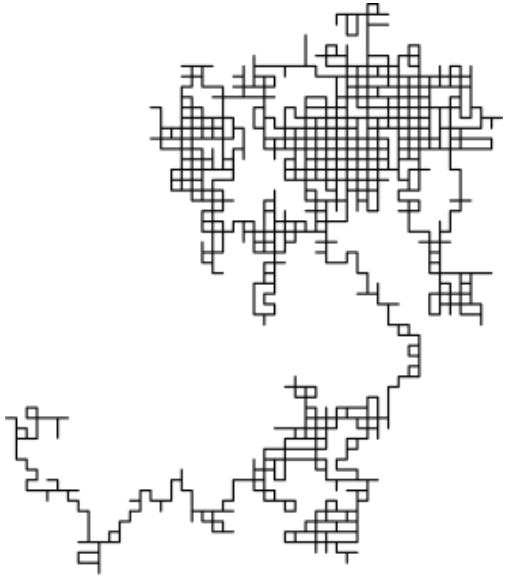
Mads-Peter V. Christiansen

Stochastic Process:

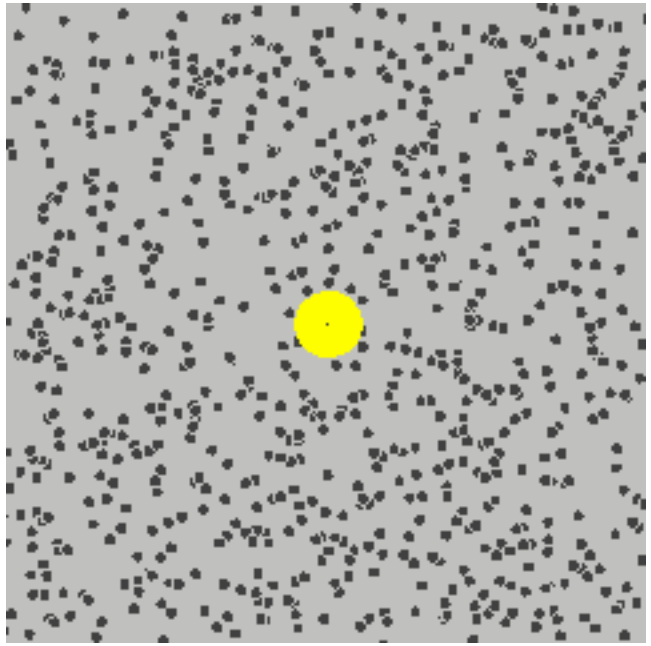
- "A collection of random variables indexed by time, space, or some other domain."

Think a **recipe** for generating sequences of random variables.

Random walk



Brownian Motion

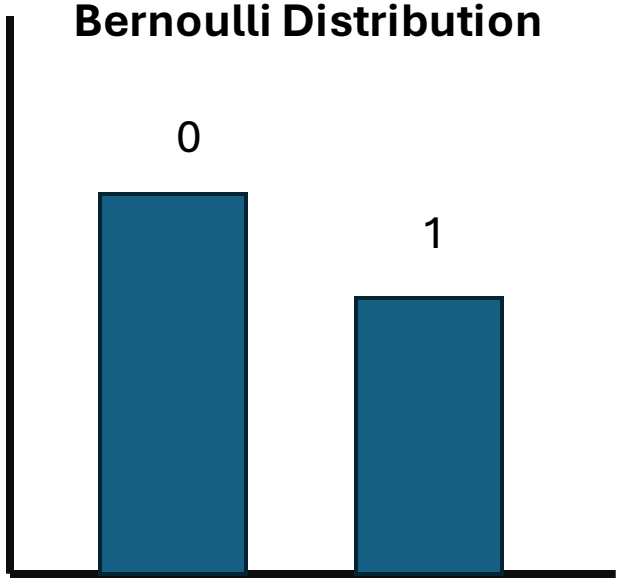


Bernoulli Process:

- Sequence of random variables each drawn iid. from a Bernoulli distribution.

This definition tells us what the **recipe** is.

Bernoulli Distribution



Tells you how to draw a single random variable

$\neq$

Bernoulli Process

Sample paths drawn from the process

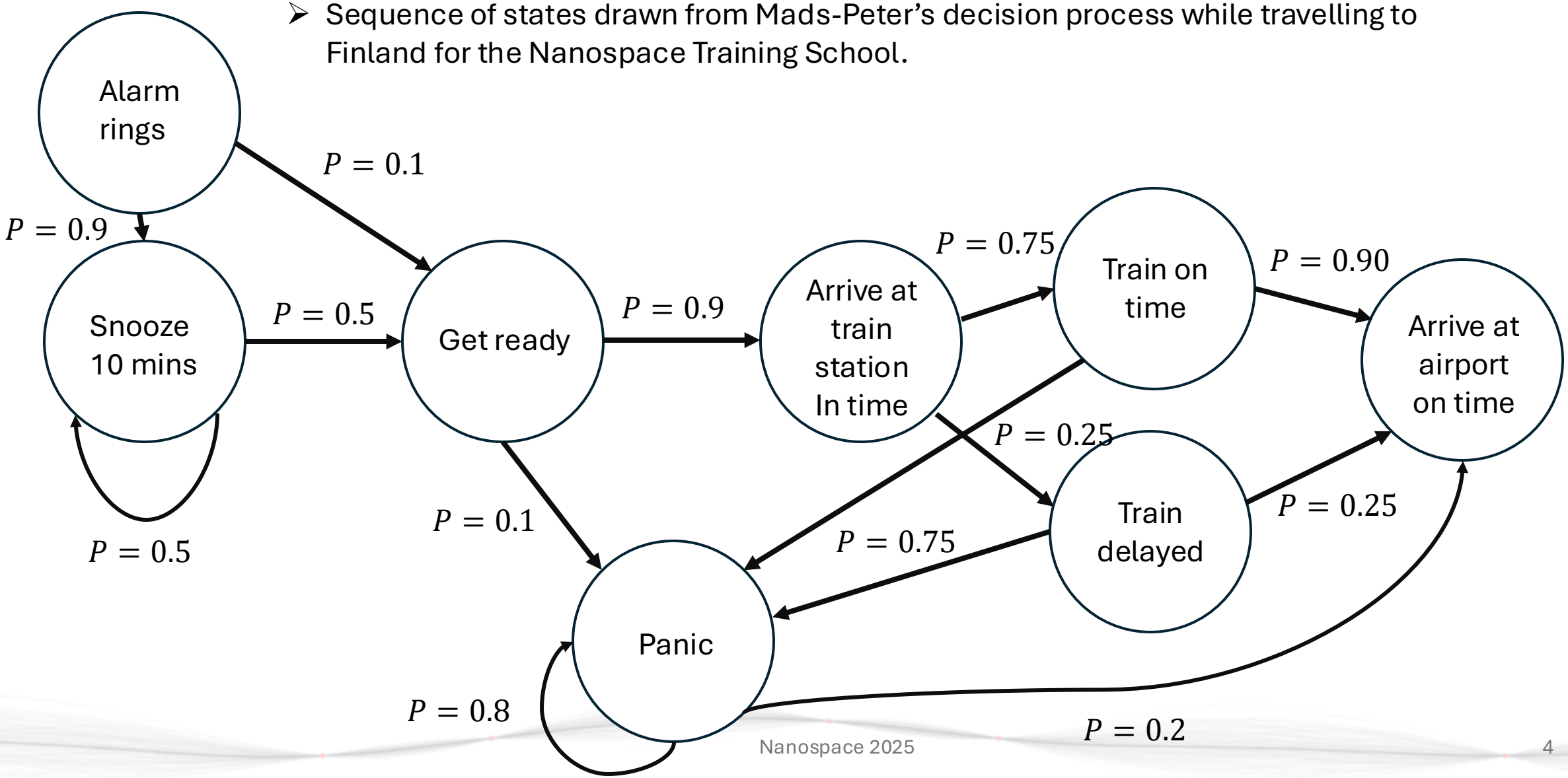
$X = \{0,0,1,0,1,1,0 \ldots\}$

$X = \{1,1,1,0,0,0,1 \ldots\}$

Tells you how to draw a collection of random variables

Mads-Peter Process

- Sequence of states drawn from Mads-Peter's decision process while travelling to Finland for the Nanospace Training School.



Mads-Peter Process

- Sequence of states drawn from Mads-Peter's decision process while travelling to Finland for the Nanospace Training School.

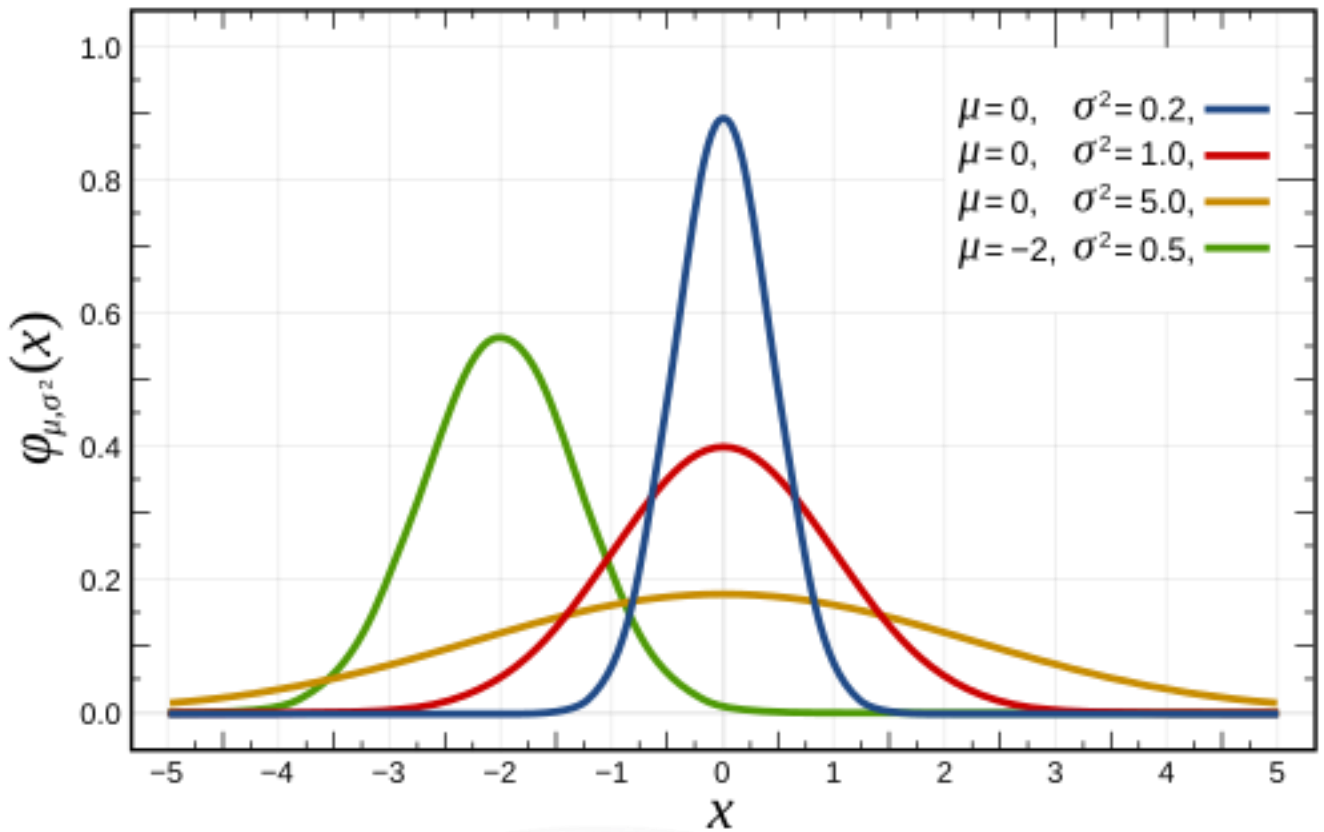
Samples from my decision process

Gaussian

- **Carl Friedrich Gauss(ian)**: German mathematician, astronomer, geodesist and physicst.
- **Gauss(ian)** (unit): Unit of magnetic flux
- **Gaussian function**: A function of the form e^{-x^2}
- **Gaussian distribution:** Continuous probability distribution.

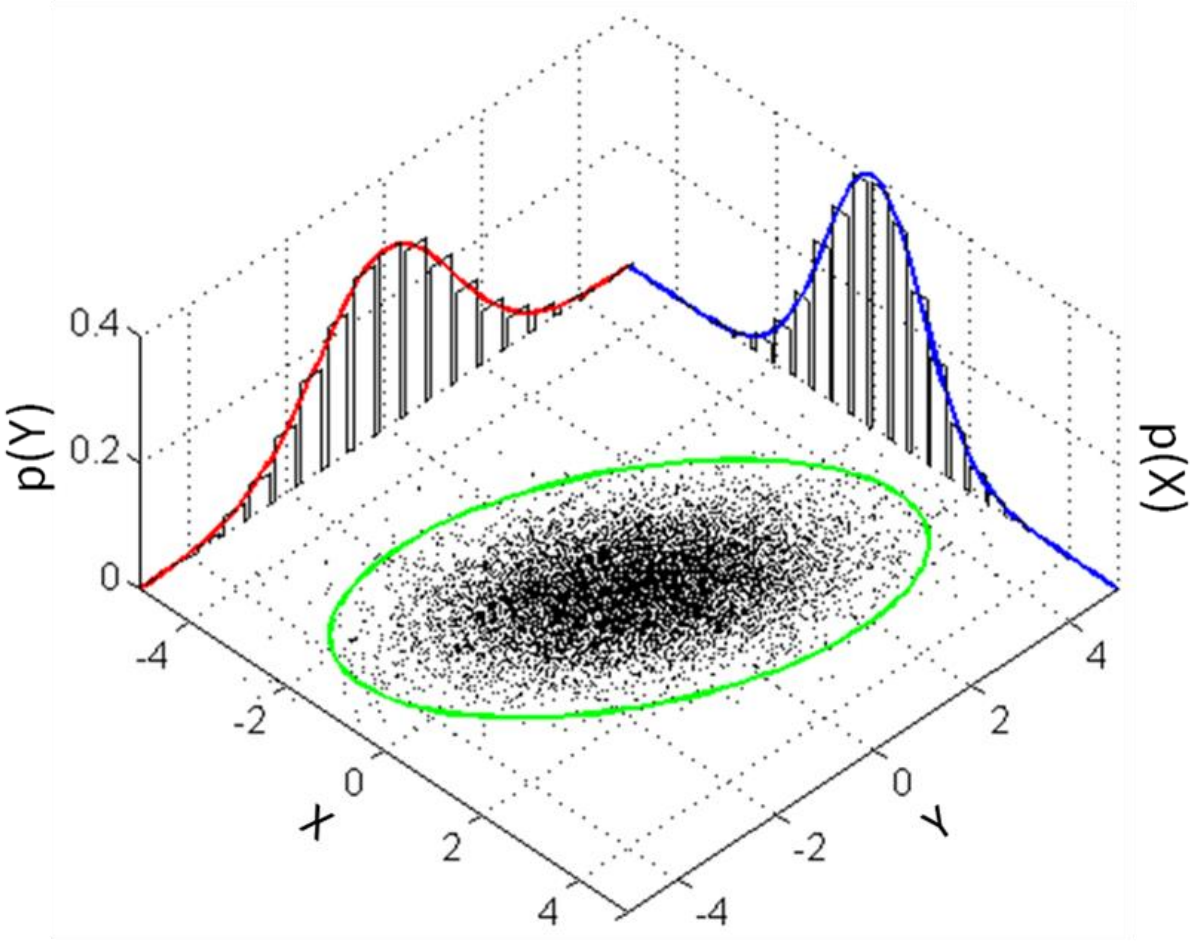
Gaussian distribution

- Normal distribution with the familiar bell shape parameterized by the mean μ and the variance σ.



Multivariate Gaussian distribution

- A distribution over vectors, parameterized by the mean $\boldsymbol{\mu}$ and the covariance $\boldsymbol{\Sigma}$.



Multivariate Gaussian distribution

- Multivariate normal distribution parameterized by the mean $\boldsymbol{\mu}$ and the covariance $\boldsymbol{\Sigma}$.

Stochastic Process:

- "A collection of random variables indexed by time, space, or some other domain."

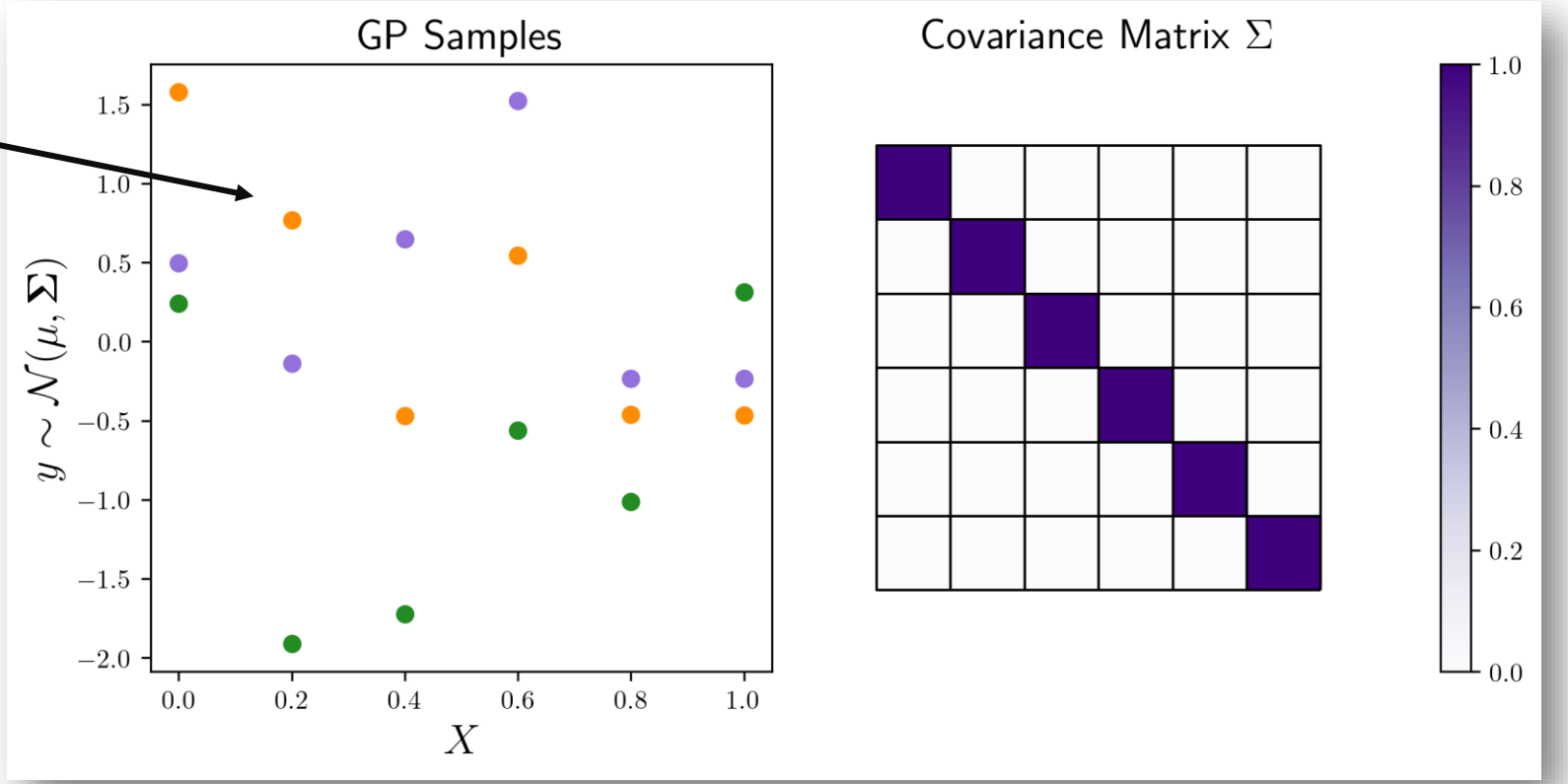
Gaussian Process

- A stochastic process where samples are generated by drawing from a multivariate Gaussian distribution indexed by time, space or some other domain.

Gaussian Process

- "A stochastic process where samples are generated by drawing from a multivariate Gaussian distribution indexed by time, space or some other domain."

Three samples from the GP.

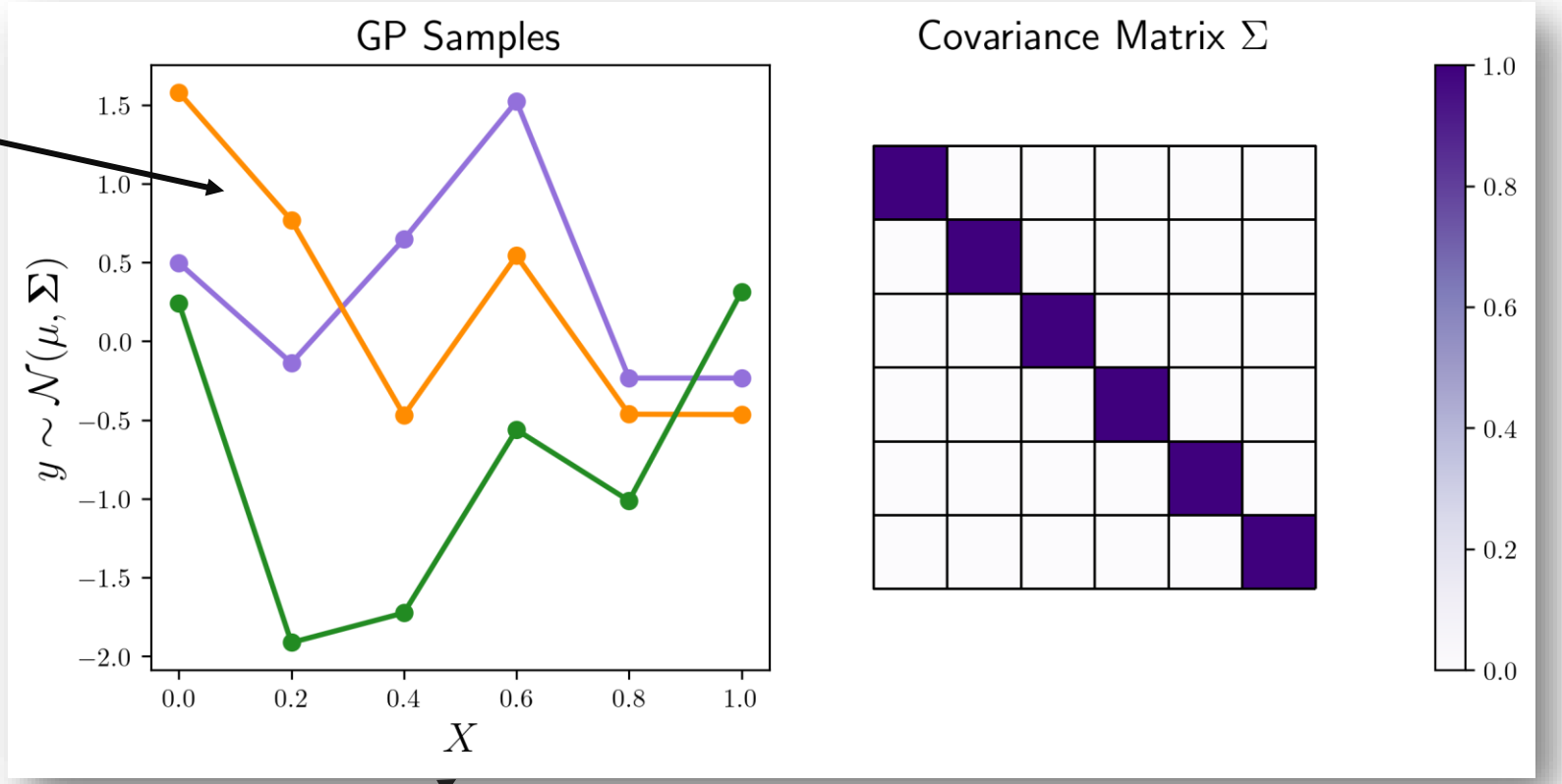


"... indexed by time, space or some other domain."

Gaussian Process

- "A stochastic process where samples are generated by drawing from a multivariate Gaussian distribution indexed by time, space or some other domain."

Three samples from the GP.



"... indexed by time, space or some other domain."

Gaussian Process

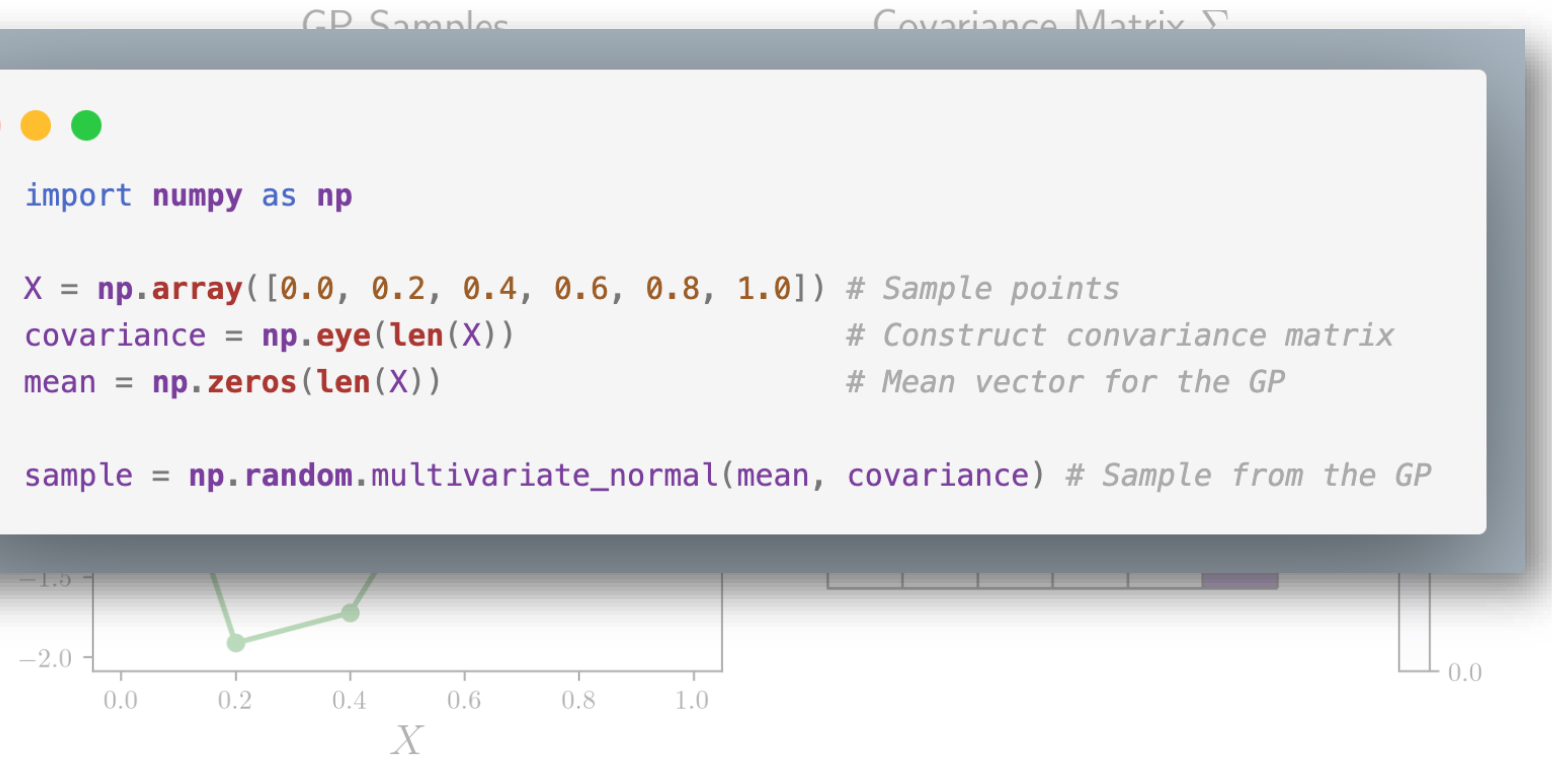
- "A stochastic process where samples are generated by drawing from a multivariate Gaussian distribution indexed by time, space or some other domain."

Three samples from the GP.

```python
import numpy as np

X = np.array([0.0, 0.2, 0.4, 0.6, 0.8, 1.0]) # Sample points
covariance = np.eye(len(X))                  # Construct convariance matrix
mean = np.zeros(len(X))                      # Mean vector for the GP

sample = np.random.multivariate_normal(mean, covariance) # Sample from the GP
```

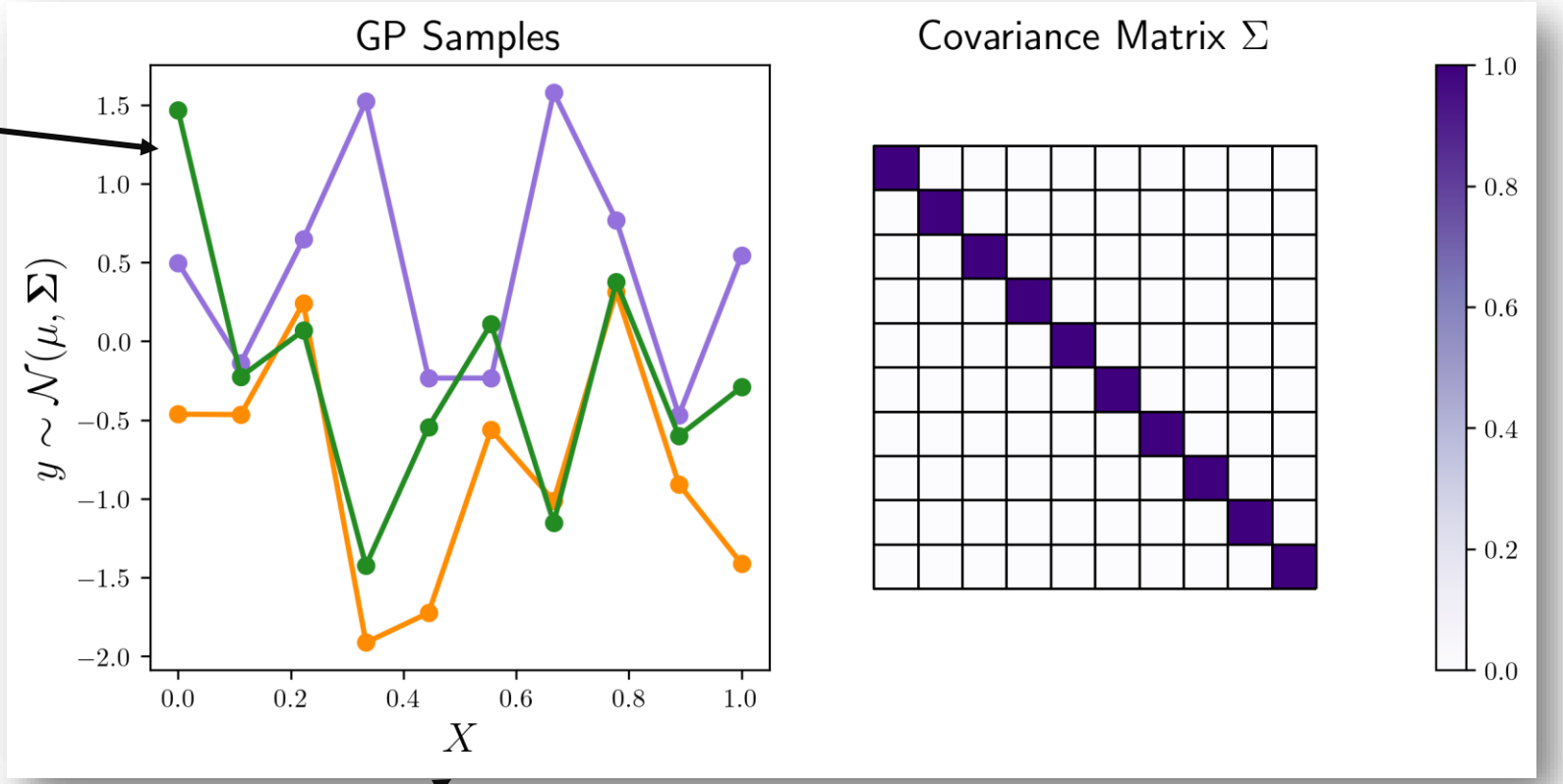


"... indexed by time, space or some other domain."

Gaussian Process

- "A stochastic process where samples are generated by drawing from a multivariate Gaussian distribution indexed by time, space or some other domain."

Three samples from the GP.

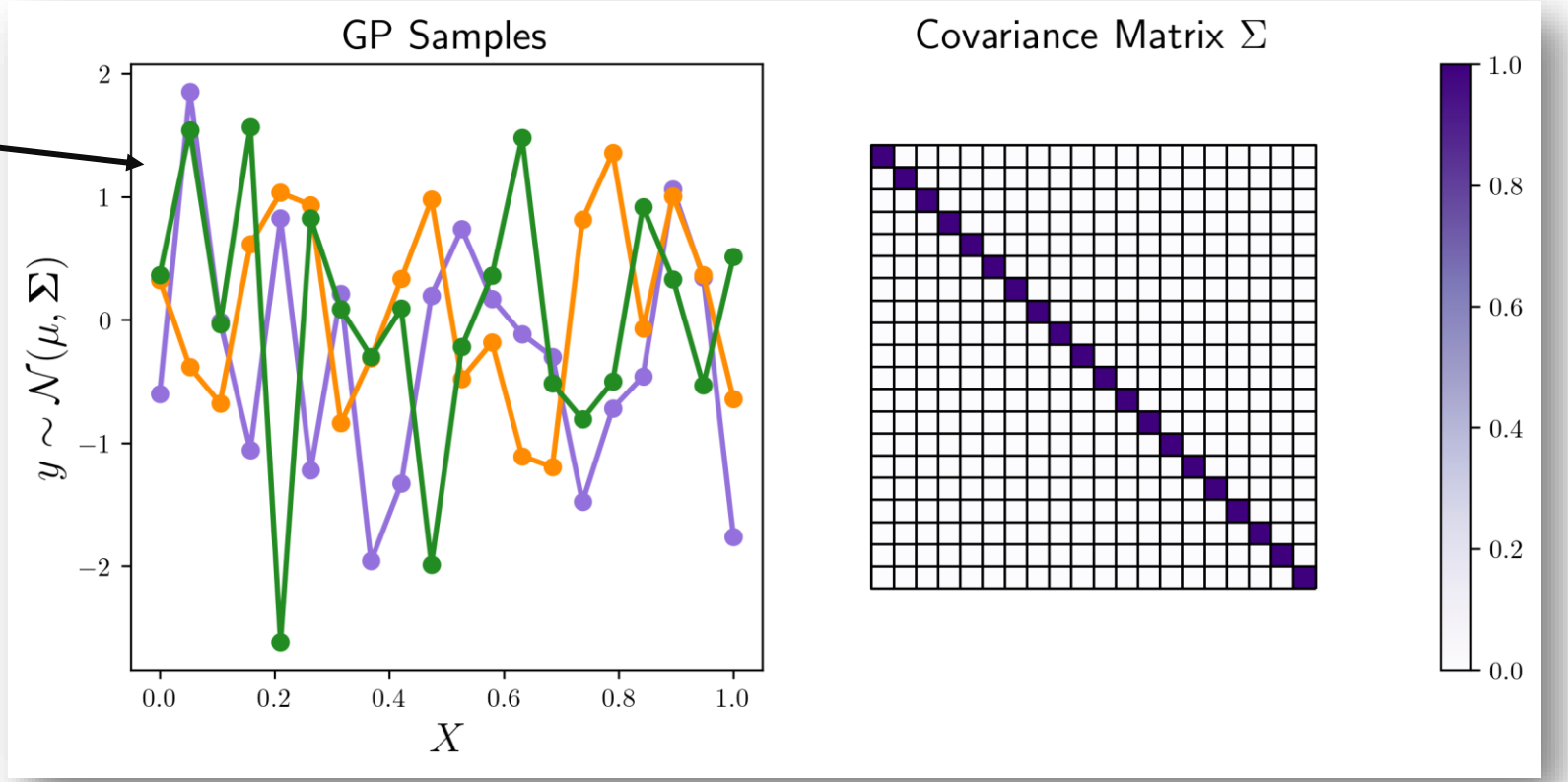


"... indexed by time, space or some other domain."

Gaussian Process

- "A stochastic process where samples are generated by drawing from a multivariate Gaussian distribution indexed by time, space or some other domain."

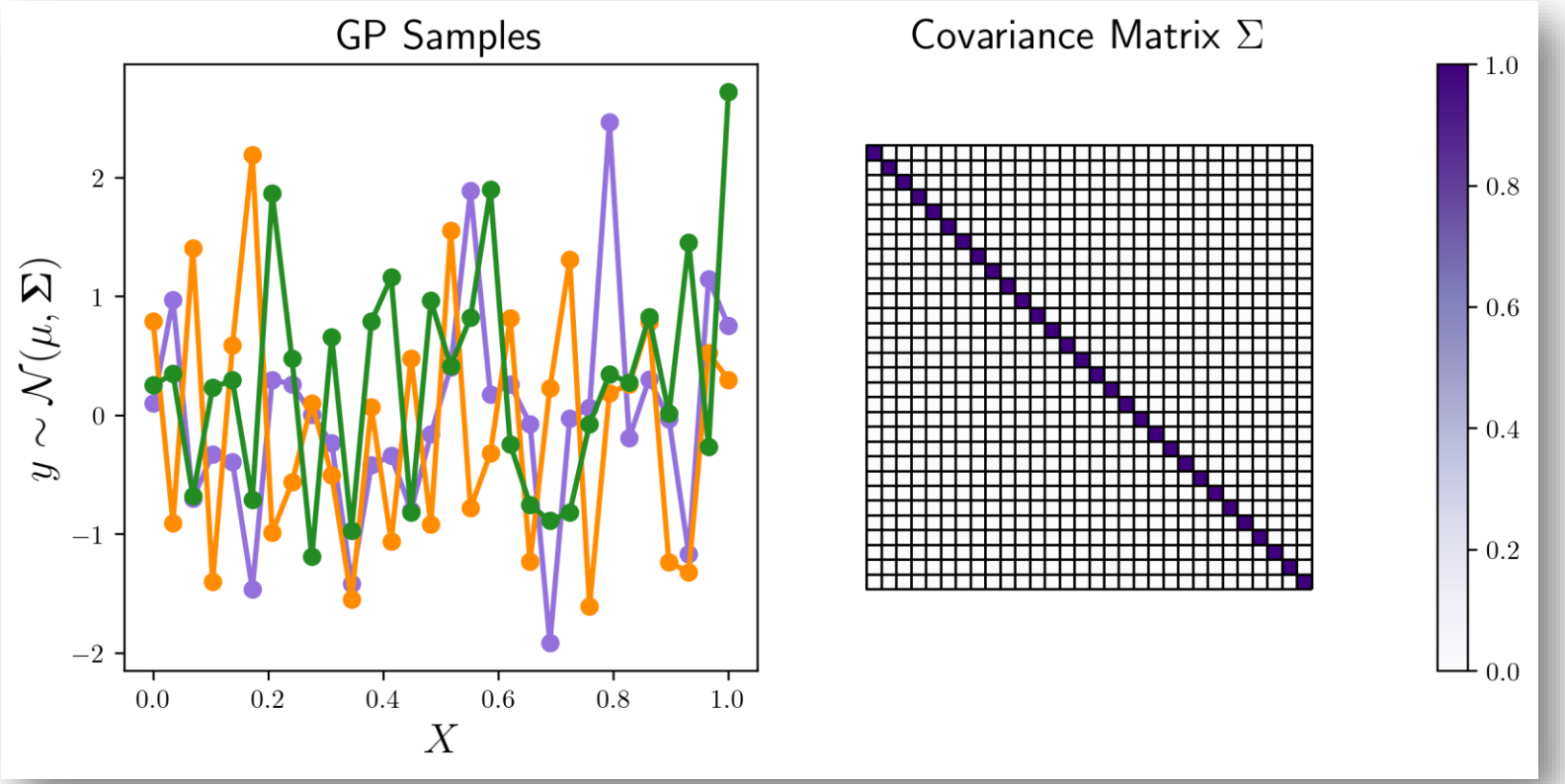
Three samples from the GP.



"… indexed by time, space or some other domain."

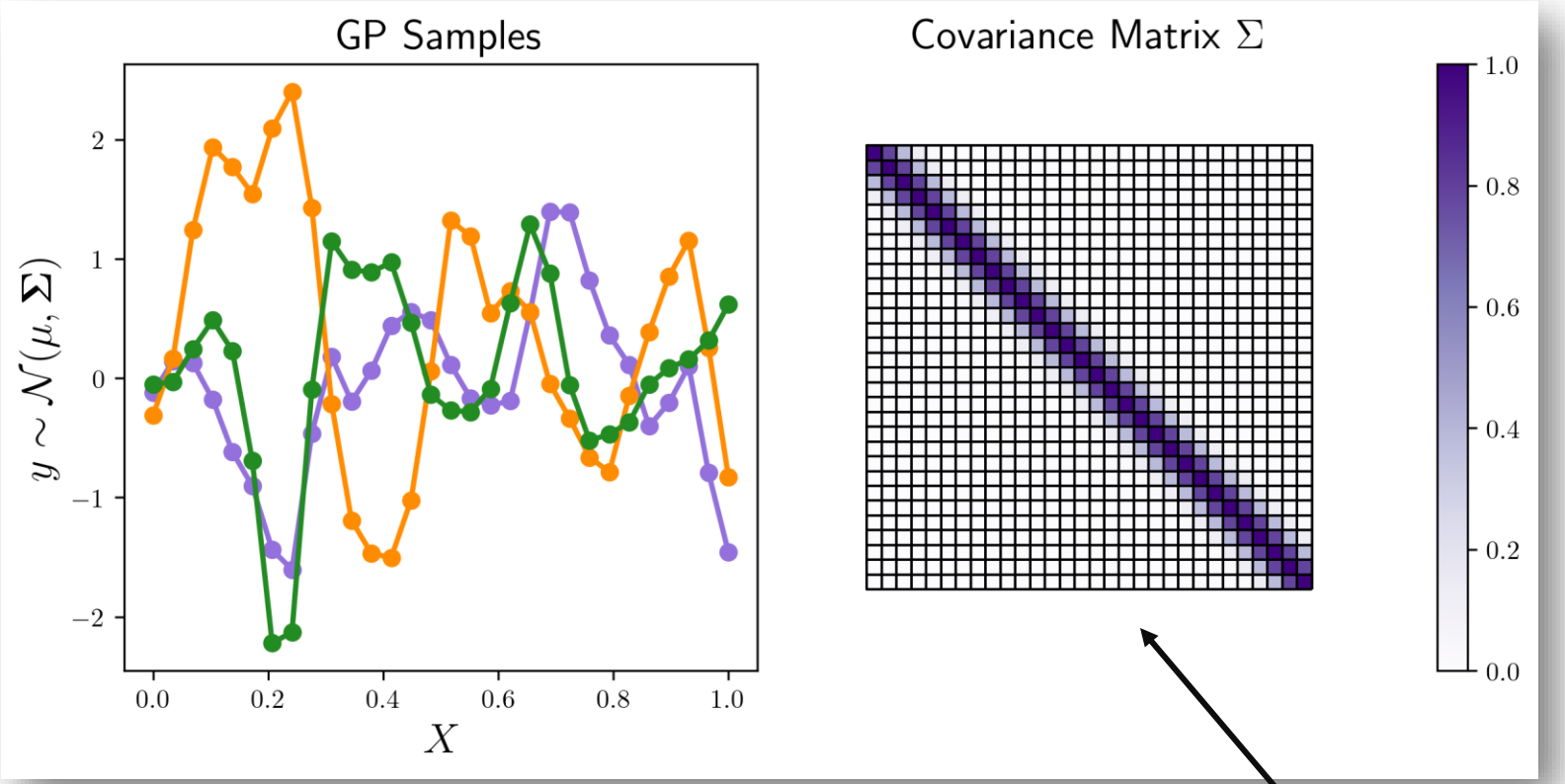
Gaussian Process

- "A stochastic process where samples are generated by drawing from a multivariate Gaussian distribution indexed by time, space or some other domain."



Gaussian Process

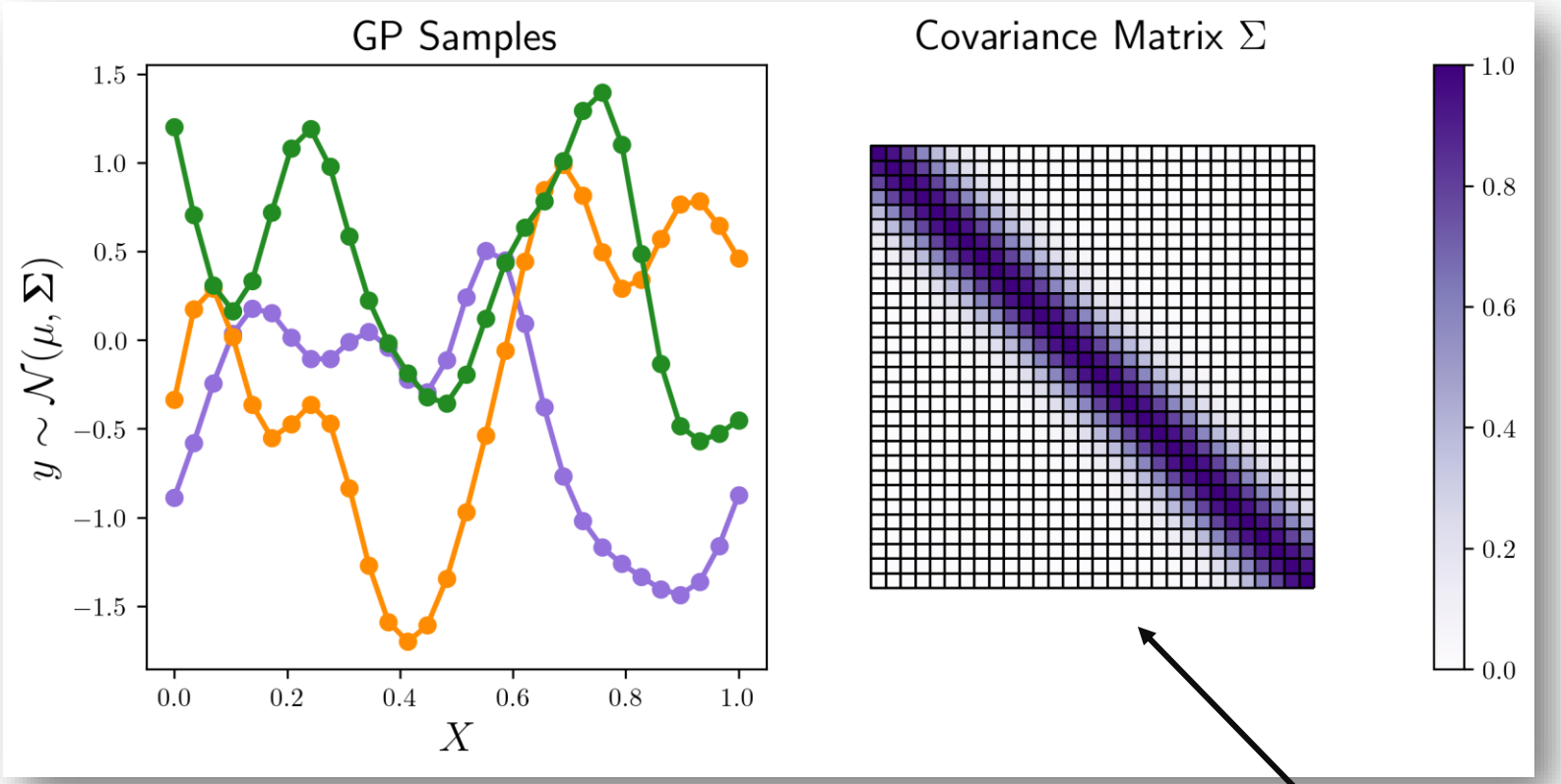
- "A stochastic process where samples are generated by drawing from a multivariate Gaussian distribution indexed by time, space or some other domain."



More structure in the covariance matrix

Gaussian Process

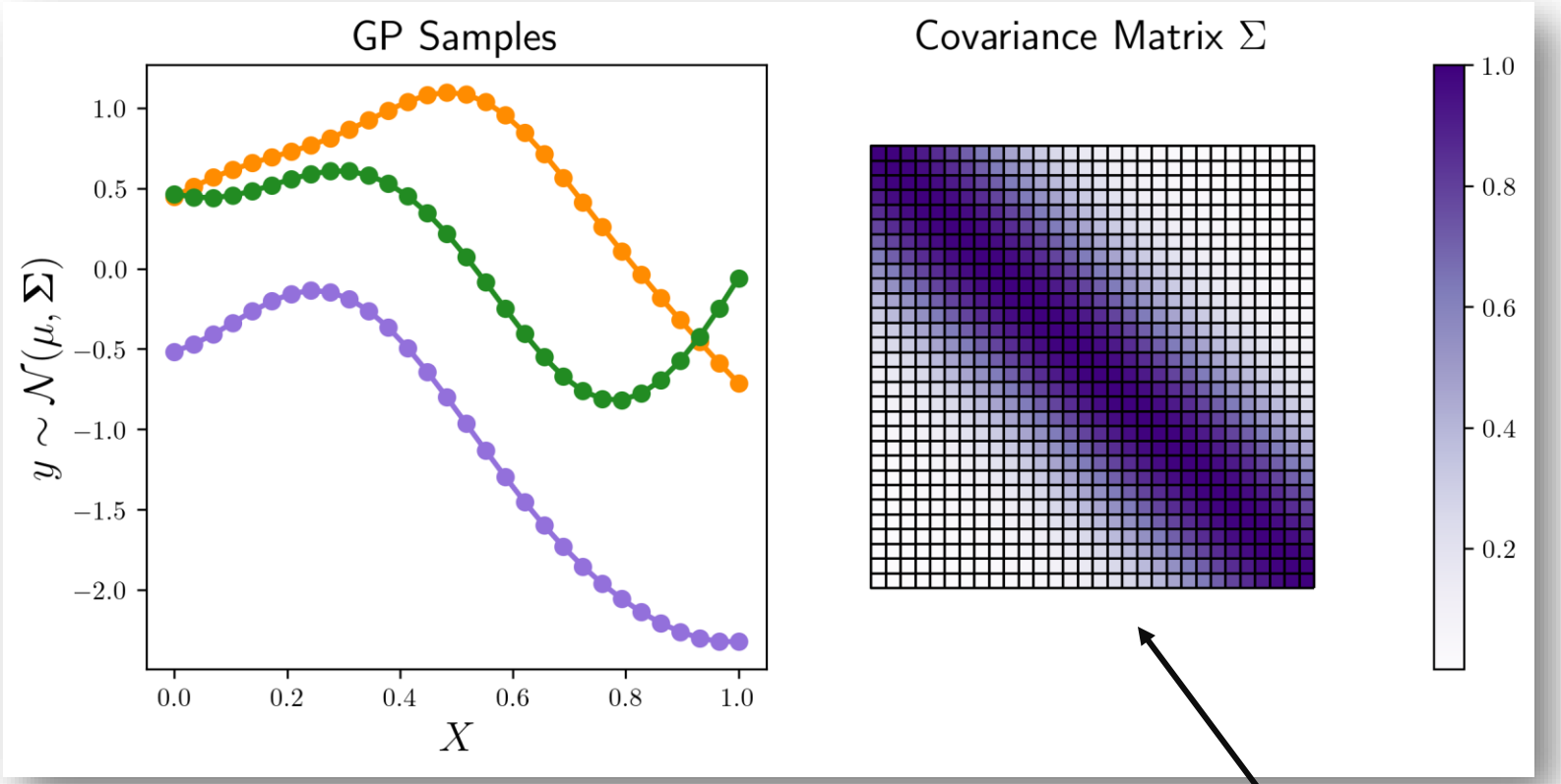
- "A stochastic process where samples are generated by drawing from a multivariate Gaussian distribution indexed by time, space or some other domain."



More structure in the covariance matrix

Gaussian Process

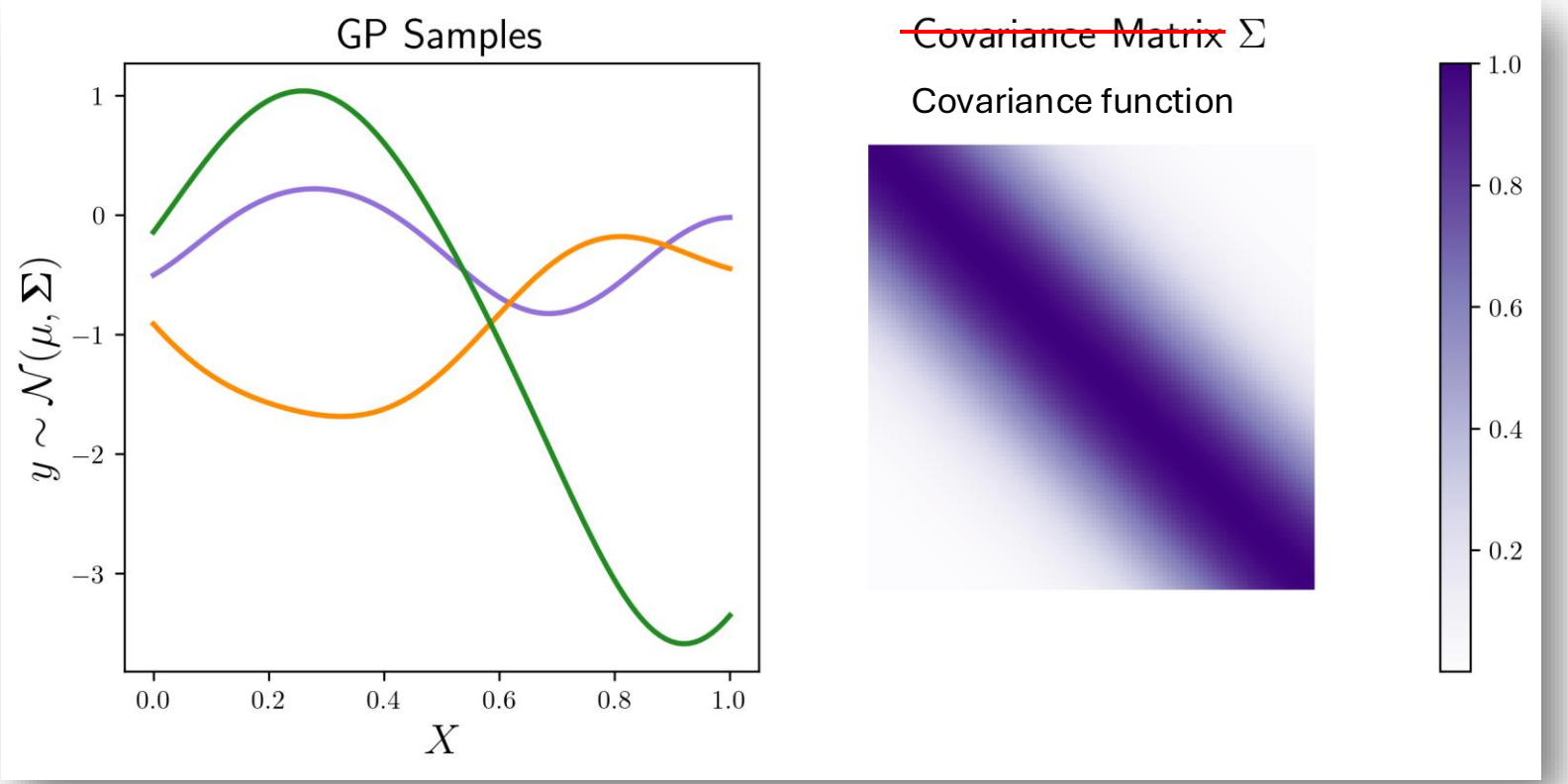
- "A stochastic process where samples are generated by drawing from a multivariate Gaussian distribution indexed by time, space or some other domain."



More structure in the covariance matrix

Gaussian Process

- "~~A stochastic process where samples are generated by drawing from a multivariate Gaussian distribution indexed by time, space or some other domain.~~"



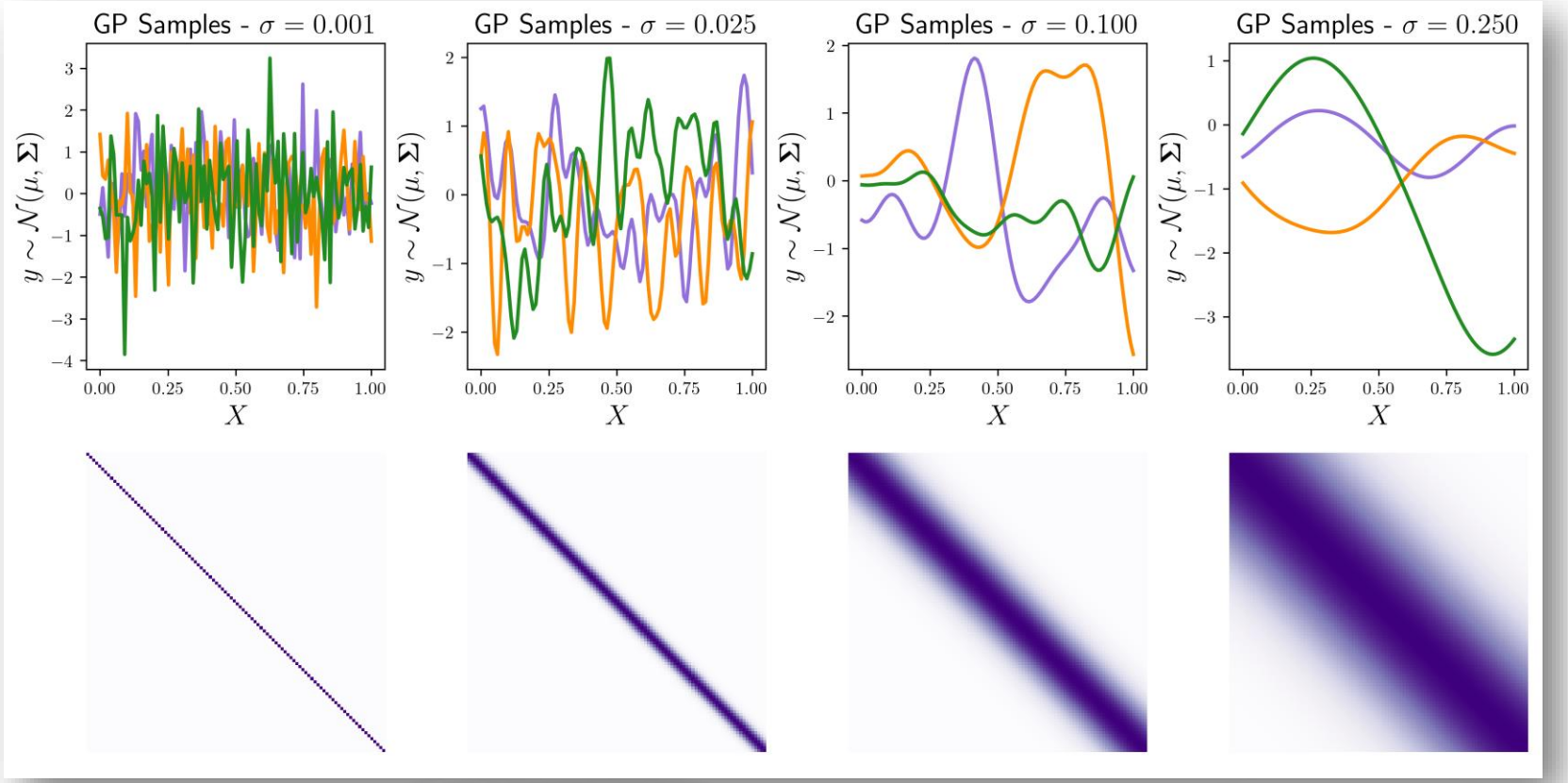
Gaussian Process

- "*A probability distribution over functions*, where any finite collection of function values is jointly *Gaussian distributed*."

Covariance matrix → Covariance function = Kernel function

- "A kernel is a function that defines the covariance between function values at any pair of input locations."

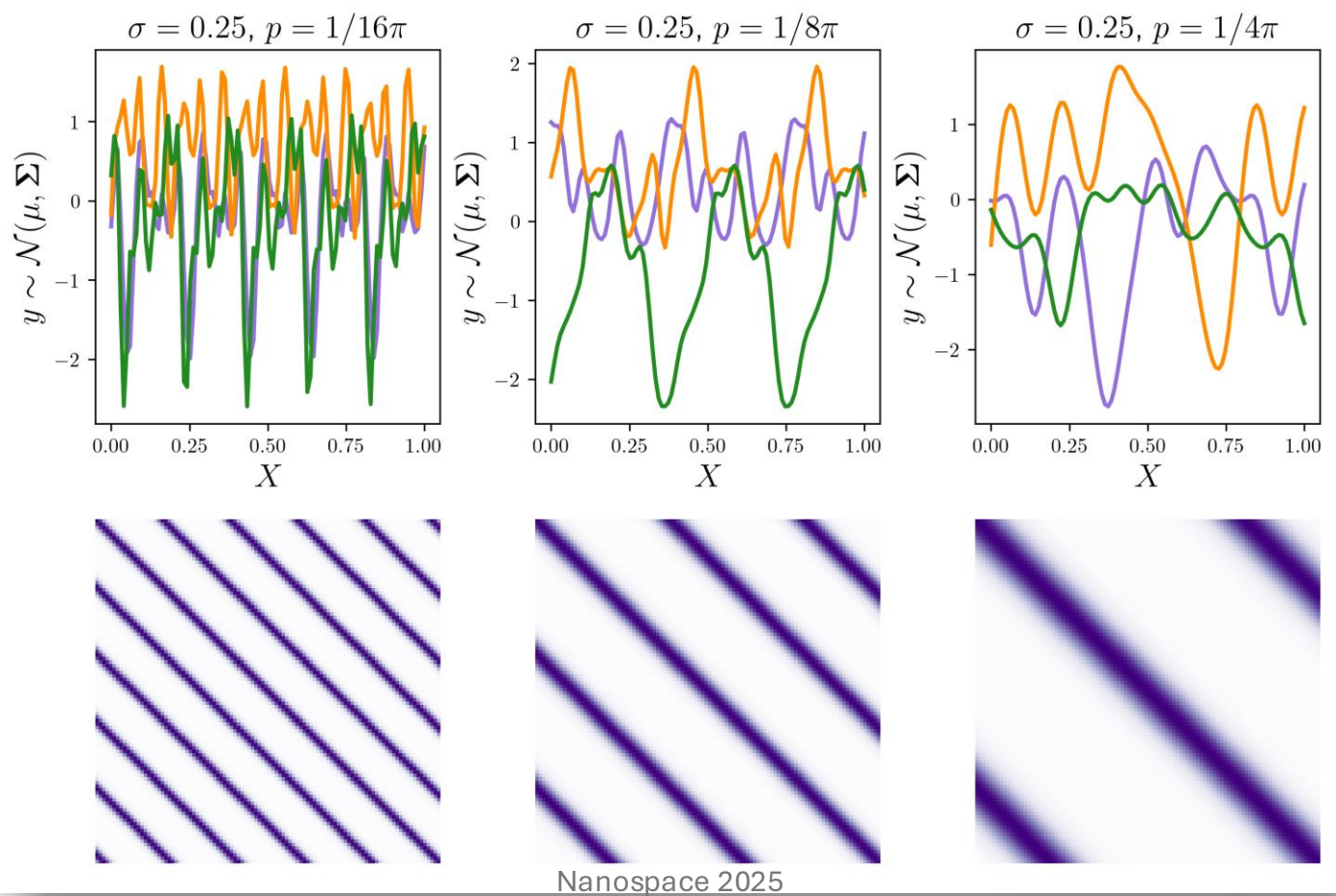
$$K(x_1, x_2) = \exp\left(-\frac{||x_2 - x_1||^2}{2\sigma^2}\right)$$ RBF or Gaussian kernel.



Covariance matrix → Covariance function = Kernel function

- "A kernel is a function that defines the covariance between function values at any pair of input locations."

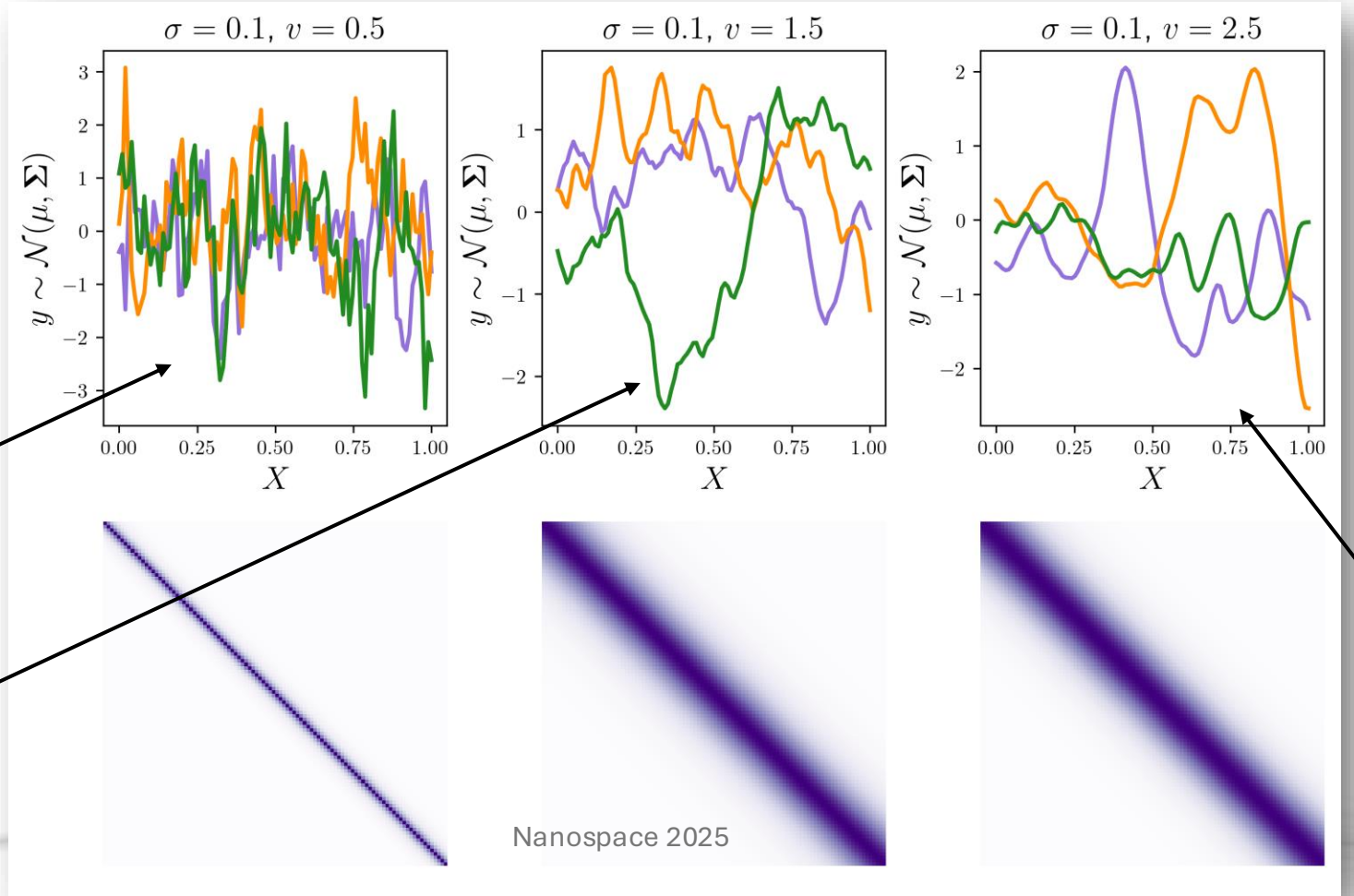
$$K(x_1, x_2) = \exp\left(-\frac{2}{\sigma^2}\sin^2\left(\frac{\pi||x_1 - x_2||}{p}\right)\right)$$ Periodic kernel



Covariance matrix → Covariance function = Kernel function

- "A kernel is a function that defines the covariance between function values at any pair of input locations."

$$k(x_i, x_j) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d(x_i, x_j)\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}}{l}d(x_i, x_j)\right), \quad \text{Matern kernel}$$



Discontinuous

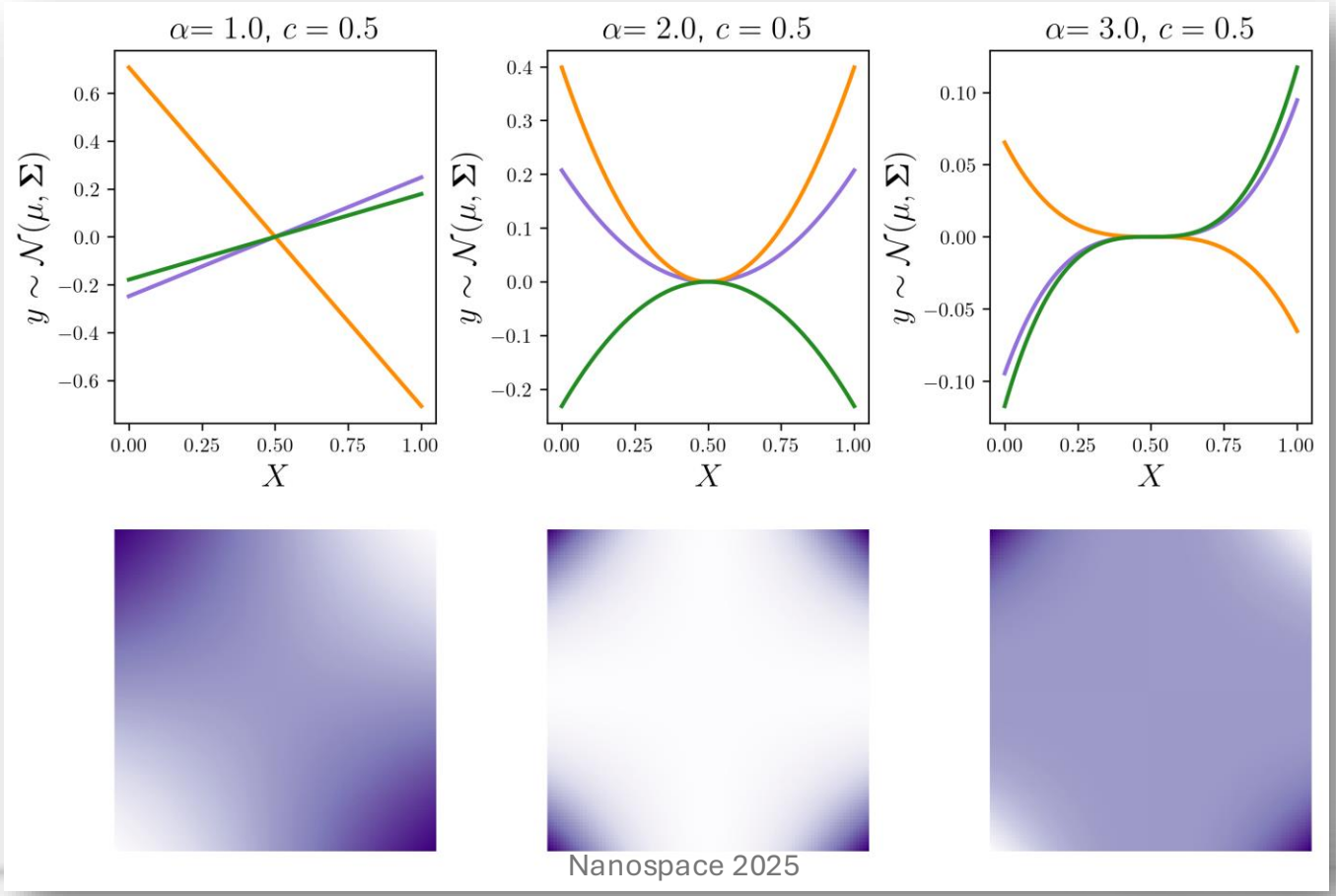
Once differentiable

Twice differentiable

Covariance matrix → Covariance function = Kernel function

- "A kernel is a function that defines the covariance between function values at any pair of input locations."

$$K(x_1, x_2) = ((x_1 - c) * (x_2 - c))^{\alpha}$$ Polynomial kernel



Multivariate Gaussian distribution

- Multivariate normal distribution parameterized by the mean $\boldsymbol{\mu}$ and the covariance $\boldsymbol{\Sigma}$.

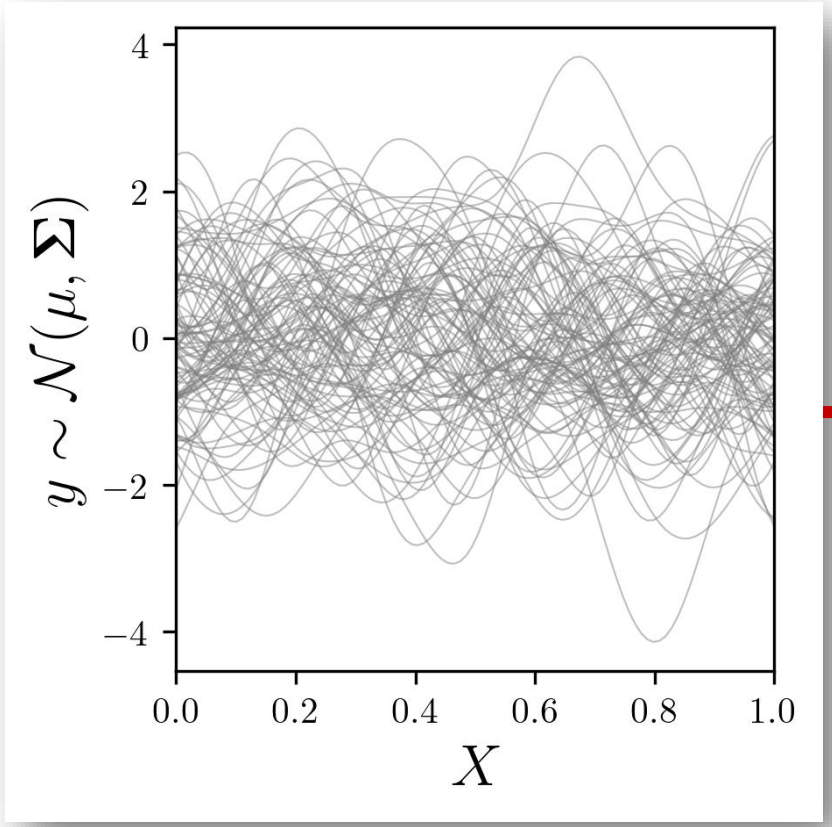
Stochastic Process:

- "A collection of random variables indexed by time, space, or some other domain."

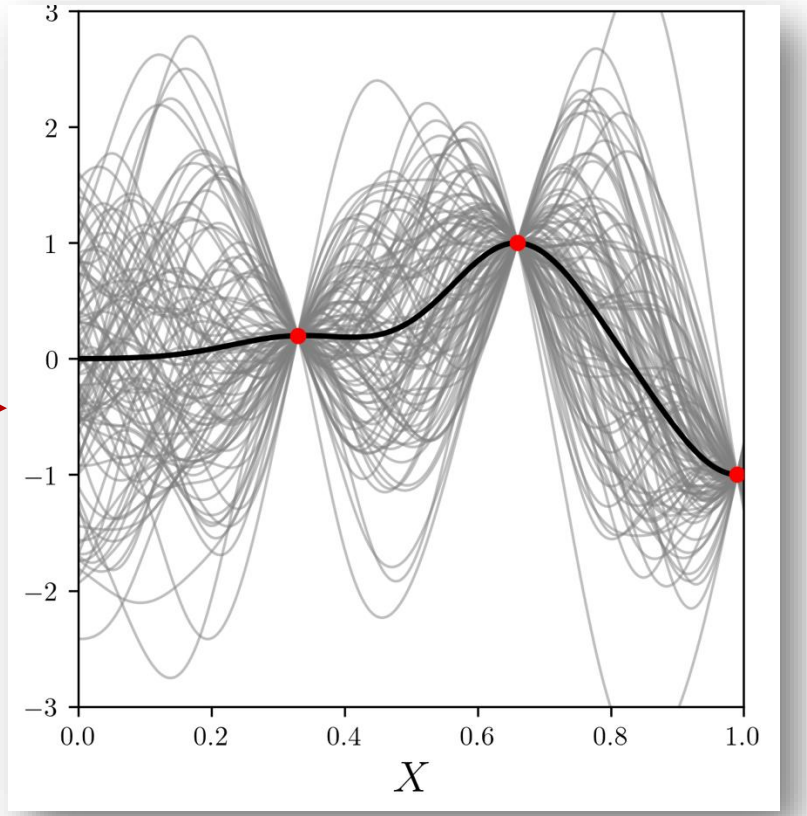
Gaussian Process

- "*A probability distribution over functions*, where any finite collection of function values is jointly *Gaussian distributed*."

Gaussian Processes & Regressors



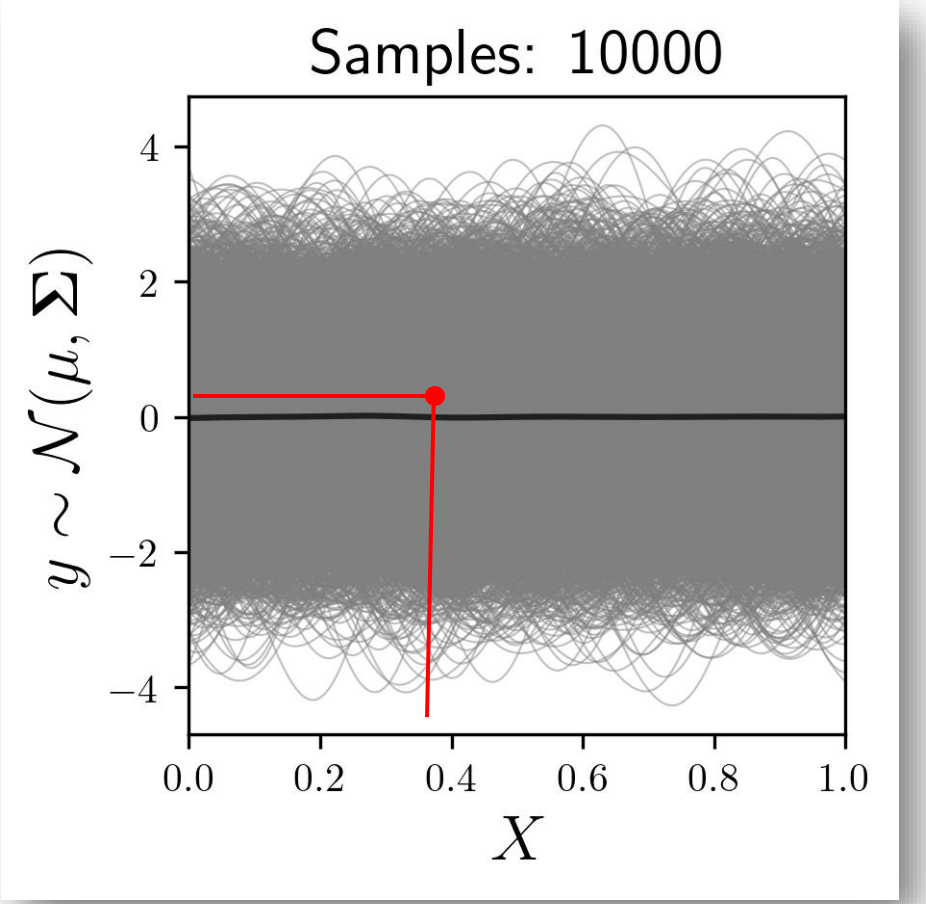
How to use for regression?



Naïve **Gaussian Process Regression:**

- Given targets $f(x_{\text{target}}) = y_{\text{target}}$
- Sample a large number of functions from the GP
- Keep only those functions that are sufficiently close to y_{target} at x_{target}

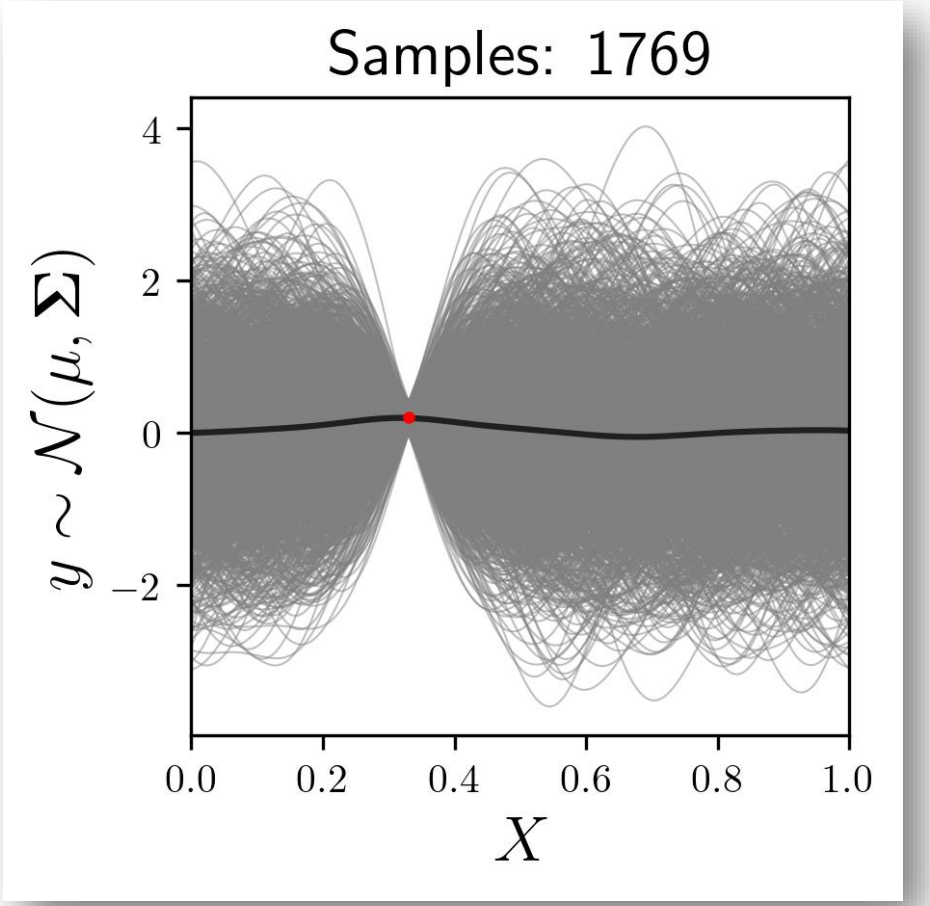
Say that our fit is the mean of the qualified samples



Naïve **Gaussian Process Regression:**

- Given targets $f(x_{\text{target}}) = y_{\text{target}}$
- Sample a large number of functions from the GP
- Keep only those functions that are sufficiently close to y_{target} at x_{target}

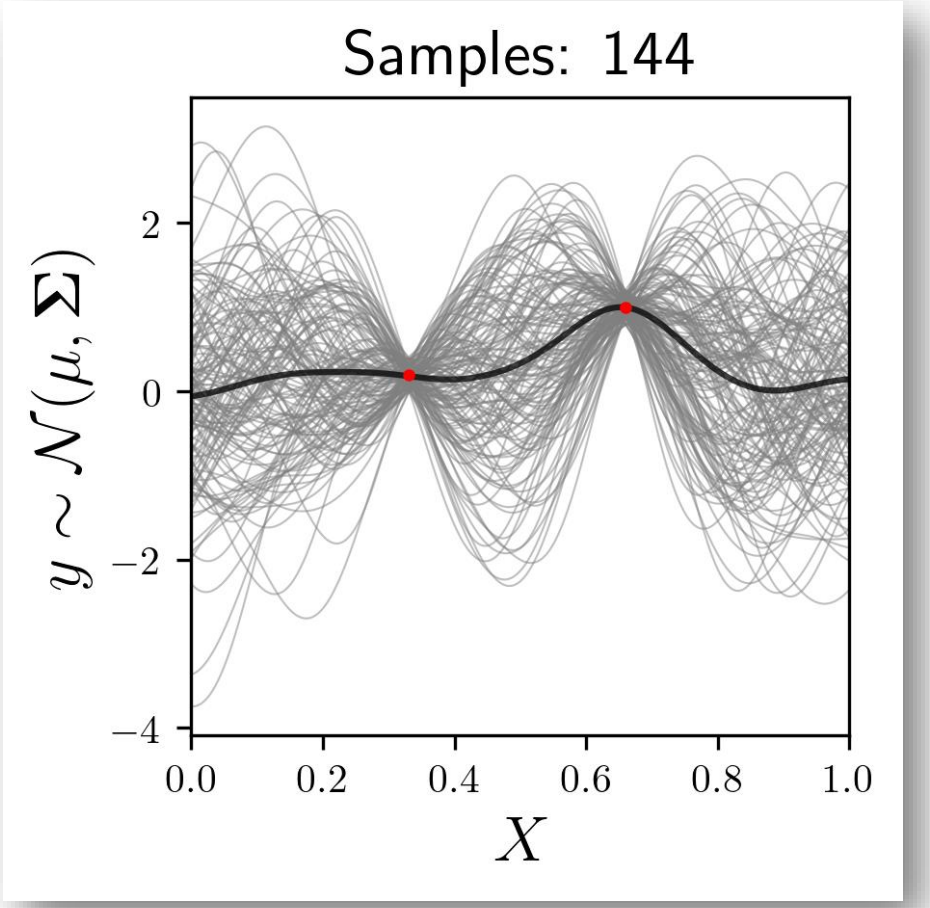
Say that our fit is the mean of the qualified samples



Naïve **Gaussian Process Regression:**

- Given targets $f(x_{\text{target}}) = y_{\text{target}}$
- Sample a large number of functions from the GP
- Keep only those functions that are sufficiently close to y_{target} at x_{target}

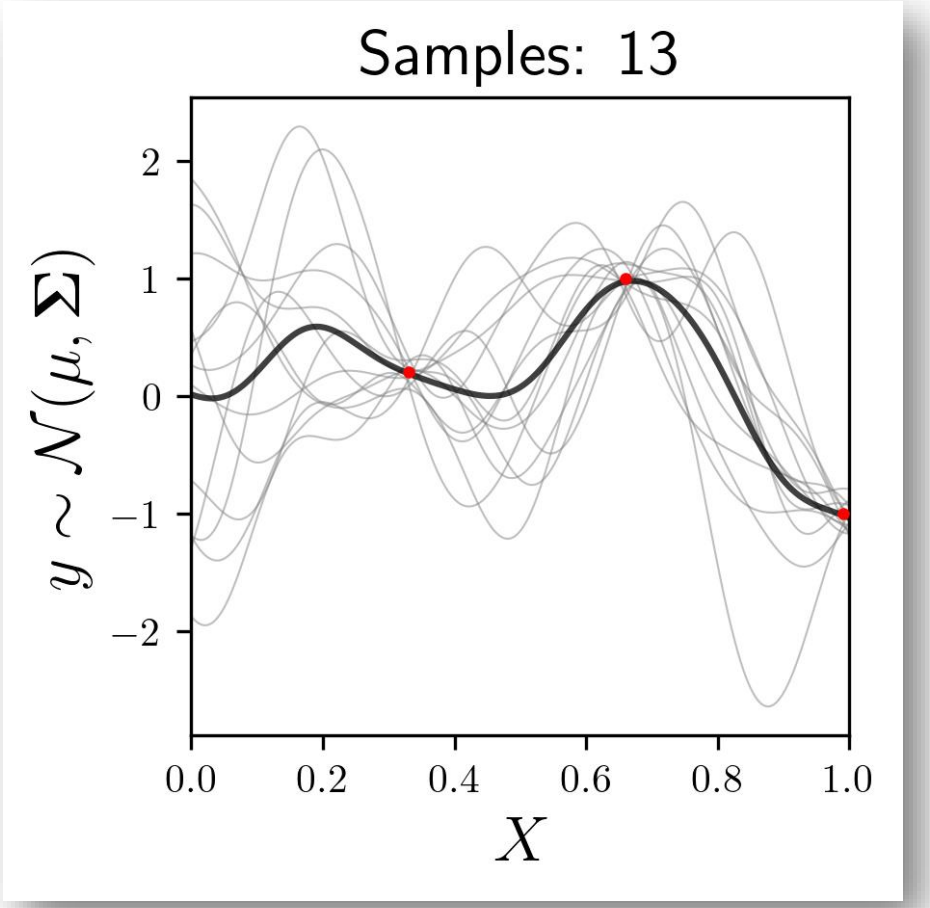
Say that our fit is the mean of the qualified samples



Naïve **Gaussian Process Regression:**

- Given targets $f(x_{\text{target}}) = y_{\text{target}}$
- Sample a large number of functions from the GP
- Keep only those functions that are sufficiently close to y_{target} at x_{target}

Say that our fit is the mean of the qualified samples



Naïve **Gaussian Process Regression:**

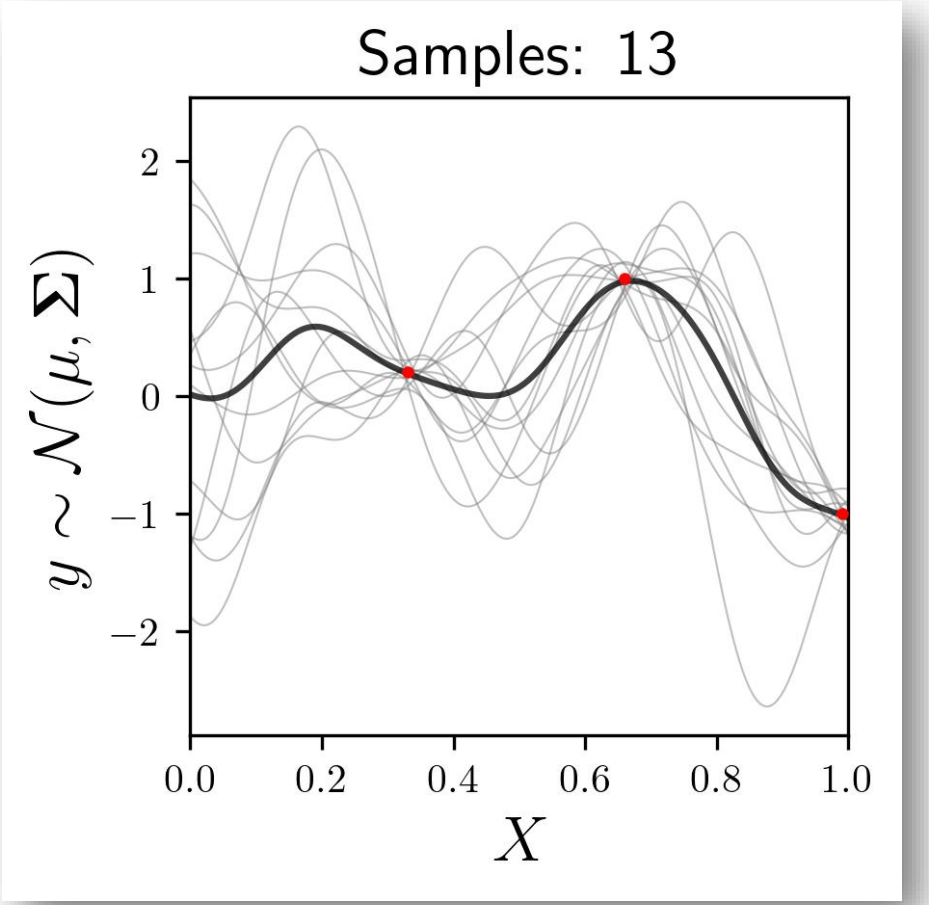
- Given targets $f(x_{\text{target}}) = y_{\text{target}}$
- Sample a large number of functions from the GP
- Keep only those functions that are sufficiently close to y_{target} at x_{target}

Say that our fit is the mean of the qualified samples

Problem: The number of samples that fit falls by about a factor of 10 pr. Added data point.

$$10000 \rightarrow 1769 \rightarrow 144 \rightarrow 13$$

Among other problems.

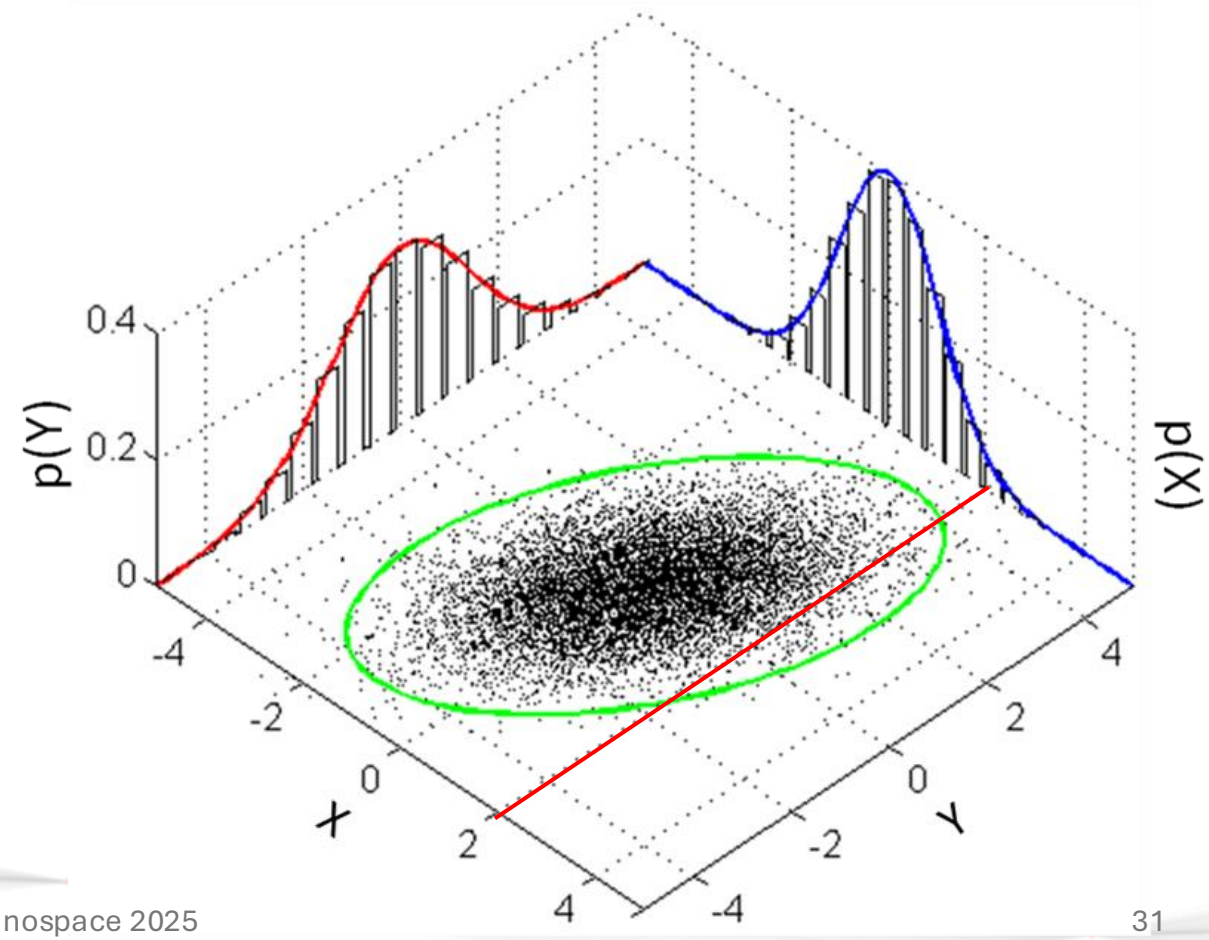


Can something smarter be done?

Bayes Theorem

- Rule to compute a conditional distribution from a joint distribution.

$$p(y|X) = \frac{p(X|y)p(y)}{p(X)}$$

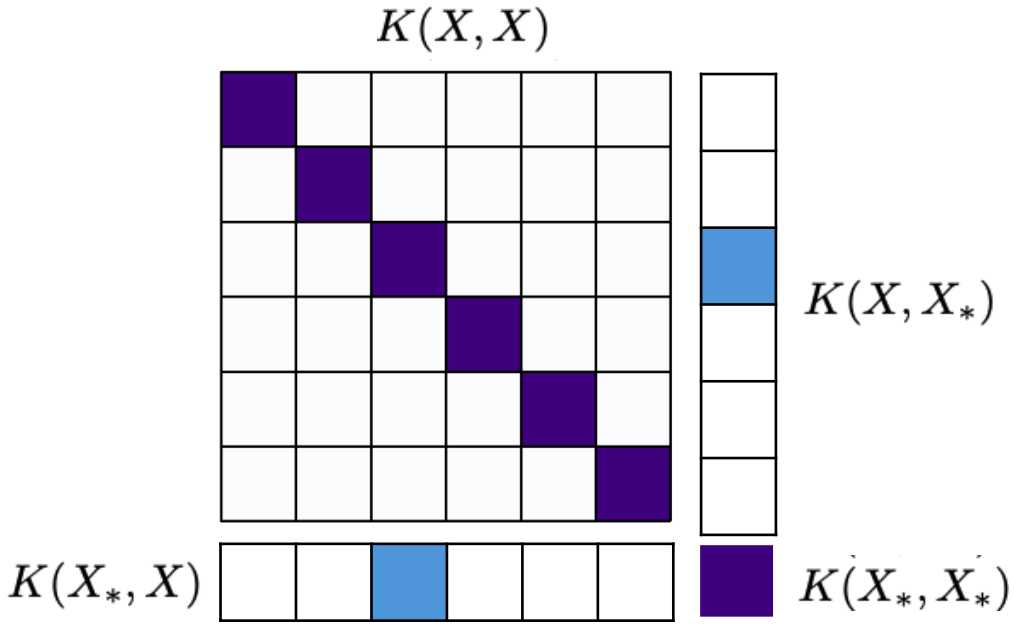


Bayes Theorem

- Rule to compute a conditional distribution from a joint distribution.

Joint prior distribution

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right).$$



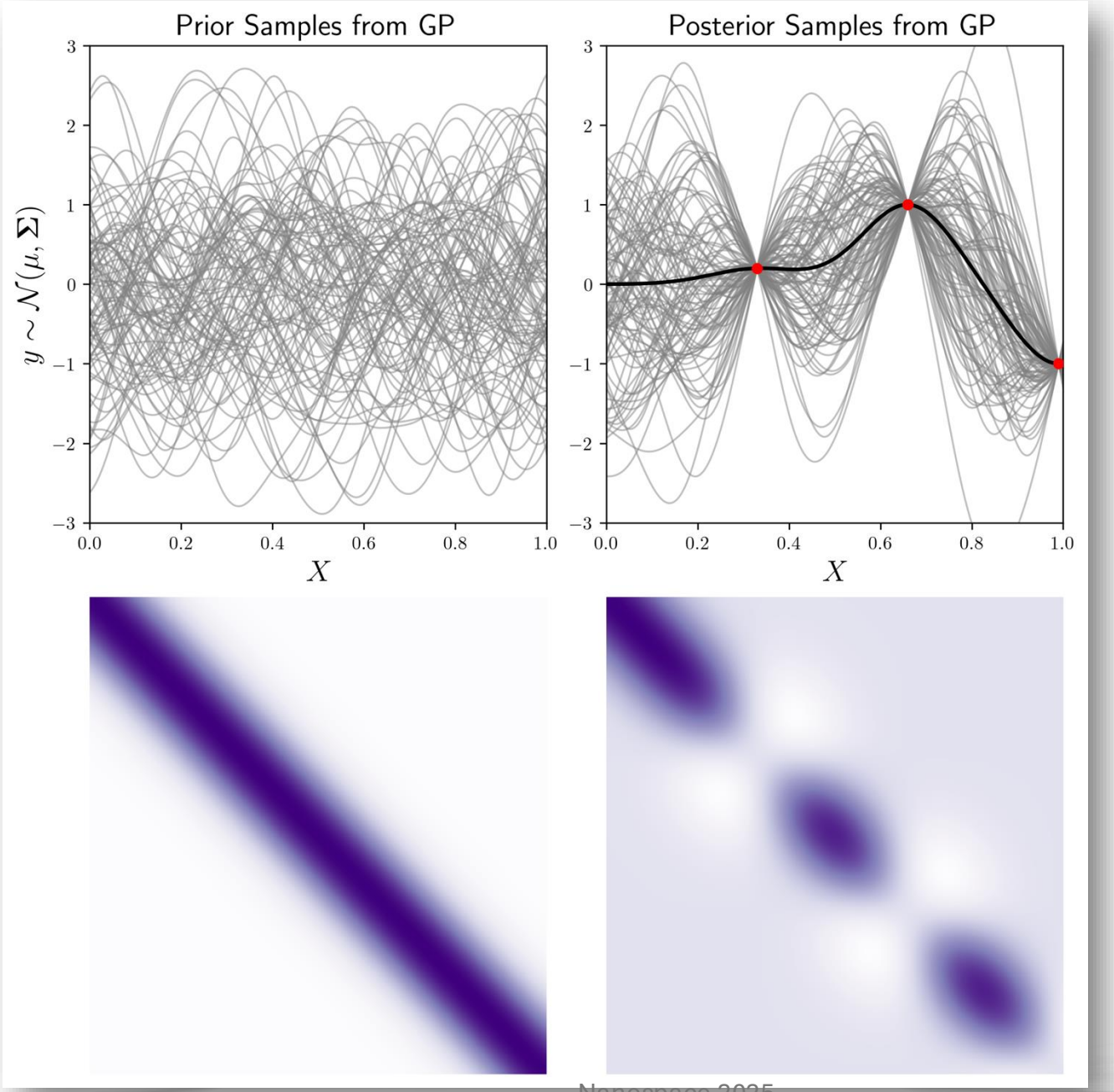
Due to the ~~magical~~ properties of Gaussians the mean and covariance of the conditional distribution are

$$\boldsymbol{\mu}^* = K(X^*,X)K(X,X)^{-1}\boldsymbol{f}$$

$$\boldsymbol{\Sigma}^* = K(X^*,X^*) - K(X^*,X)K(X,X)^{-1}K(X,X^*)$$

In GP terminology these are typically called the **posterior** mean and covariance.

$\boldsymbol{\mu} = 0$

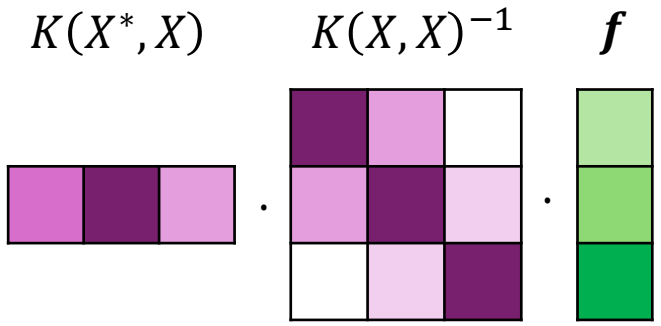


$\boldsymbol{\mu}^* = K(X^*, X)K(X, X)^{-1}\boldsymbol{f}$

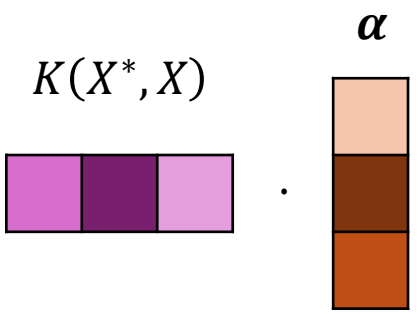
$\boldsymbol{\Sigma} = K(X^*, X^*)$

$\boldsymbol{\Sigma}^*$
$= K(X^*, X^*)$
$- K(X^*, X)K(X, X)^{-1}K(X, X^*)$

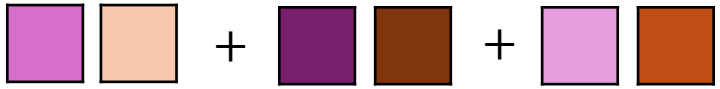
$$\boldsymbol{\mu}^* = K(X^*, X)K(X, X)^{-1}\boldsymbol{f}$$

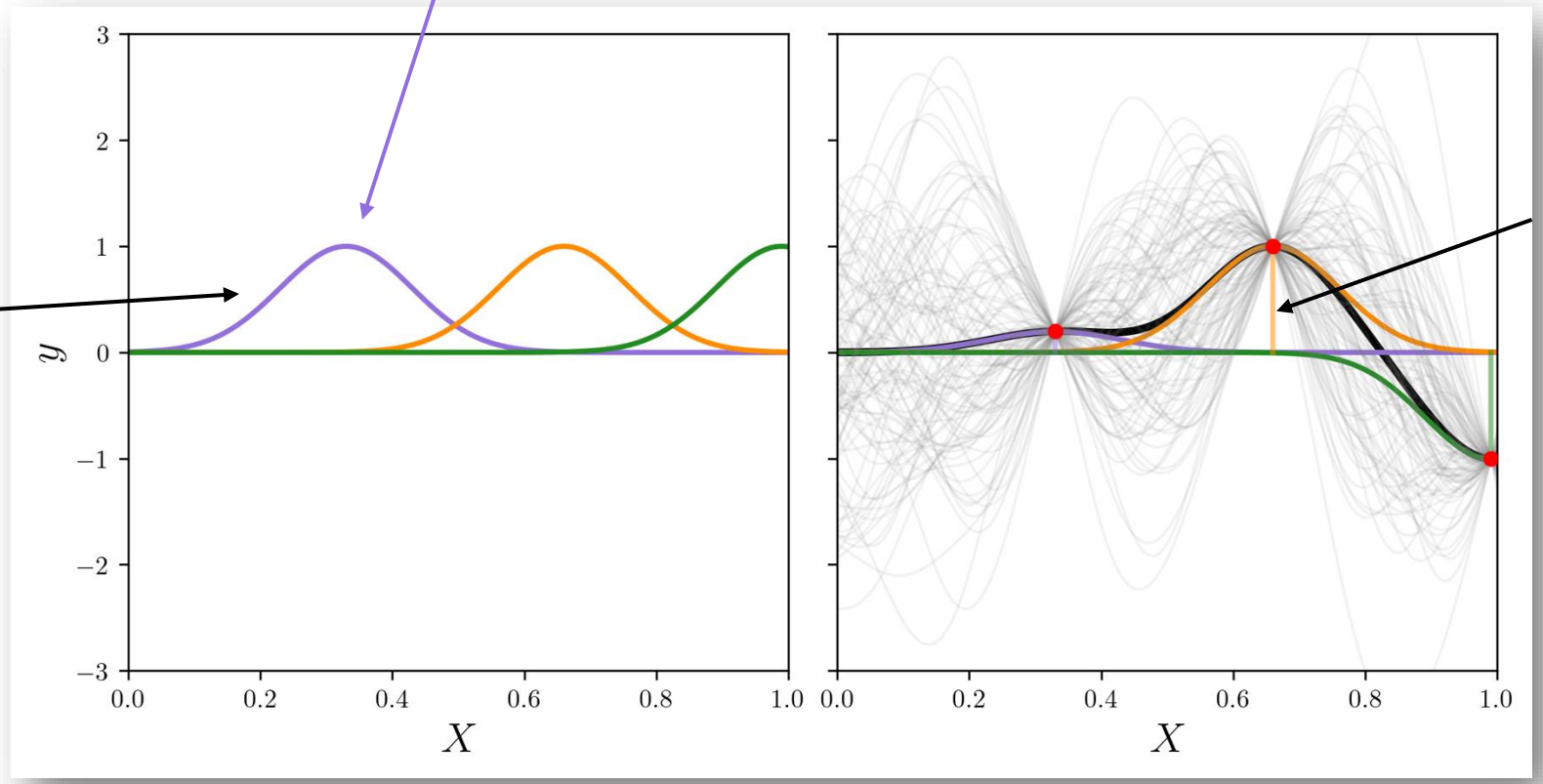


$$= K(X^*, X)\boldsymbol{\alpha}$$



$$= \sum_{\boldsymbol{i}} \alpha_{\boldsymbol{i}} K(X^*, X_i)$$



$$\boldsymbol{\mu}^* = \sum_{\boldsymbol{i}} \alpha_{\boldsymbol{i}} K(X^*, X_i) =$$  

Kernels placed at observations as basis functions

Weights α_i come from the conditioning

Bayes Theorem

- Rule to compute a conditional distribution from a joint distribution.

Joint prior distribution with noisy observations

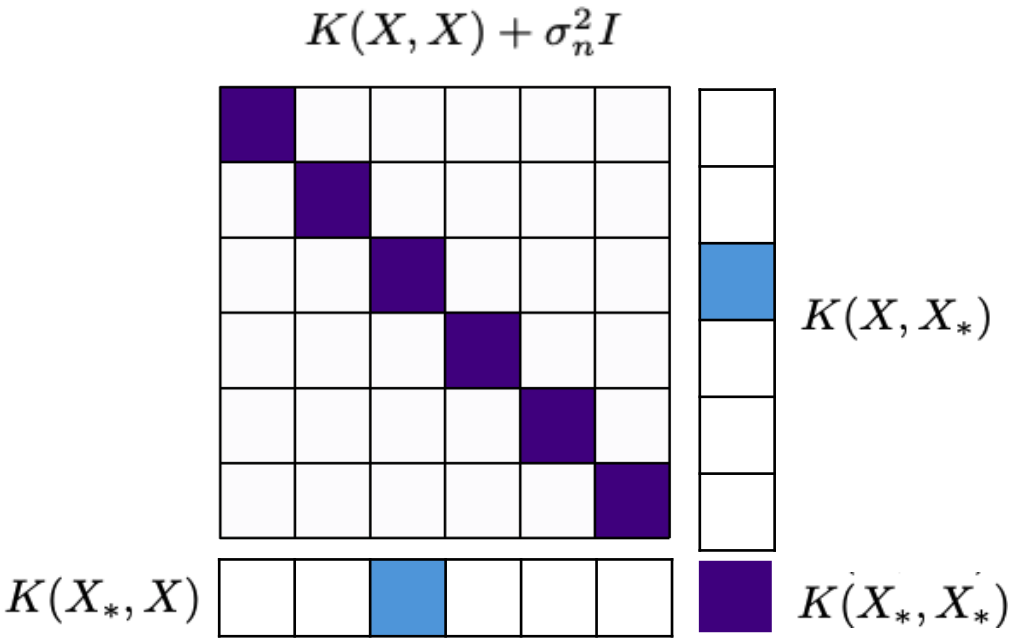
$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X)+\sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right).$$

Due to the ~~magical~~ properties of Gaussians the mean and covariance of the conditional distribution are

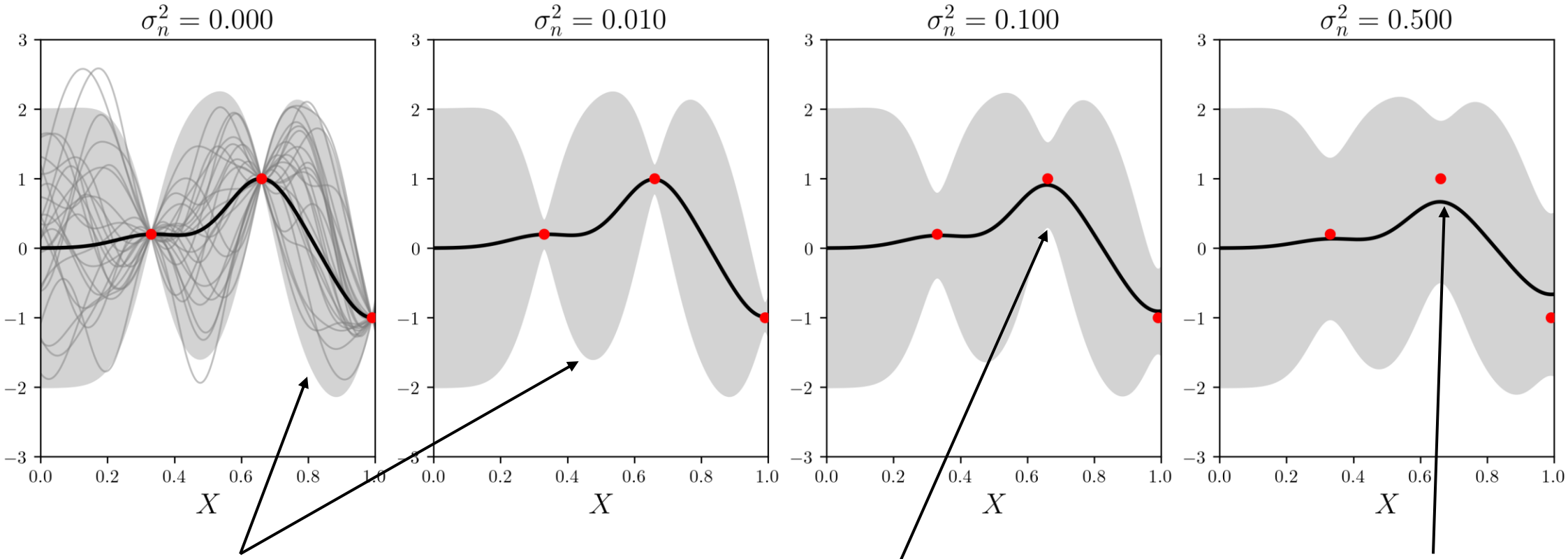
$$\boldsymbol{\mu}^* = K(X^*,X)[K(X,X)+\sigma_n^2 I]^{-1}\boldsymbol{f}$$

$$\boldsymbol{\Sigma}^* = K(X^*,X^*) - K(X^*,X)[K(X,X)+\sigma_n^2 I]^{-1}K(X,X^*)$$

In GP terminology these are typically called the **posterior** mean and covariance.



$$\boldsymbol{\mu}^* = K(X^*, X)[K(X, X)^{-1} + \sigma_n^2 I]\boldsymbol{f}$$



Rather than plotting samples we can just plot the **2 std confidence interval** resulting from the covariance.

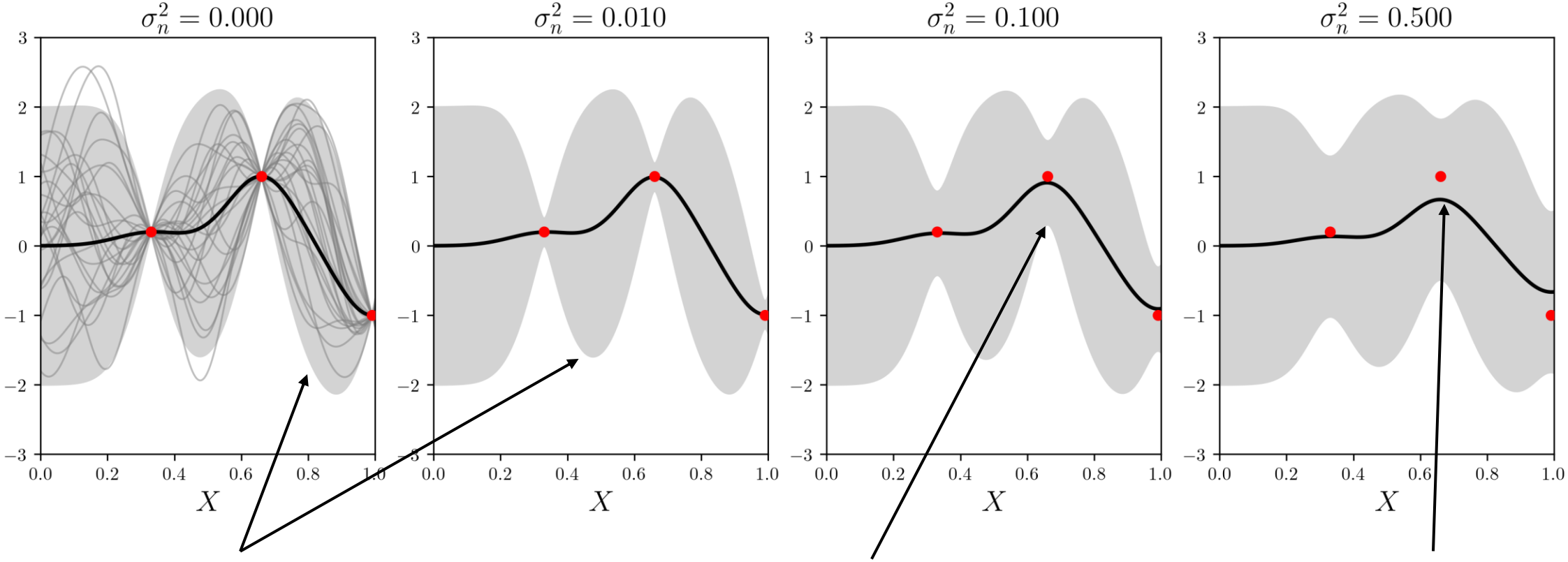
More noisy observations → Less confident

Posterior mean less obliged to hit the data → Reverts to prior mean = 0

$$\boldsymbol{\mu}^* = K(X^*, X)[K(X, X)^{-1} + \sigma_n^2 I]\boldsymbol{f}$$
$$= \sum_i \alpha_i K(X^*, X_i)$$

Kernel ridge regression

$$\longleftrightarrow \quad L(\alpha) = \left\| \sum_i \alpha_i K(X^*, X_i) - y_i \right\|^2 + \sigma_n^2 ||\boldsymbol{\alpha}||$$



Rather than plotting samples we can just plot the **2 std confidence interval** resulting from the covariance.

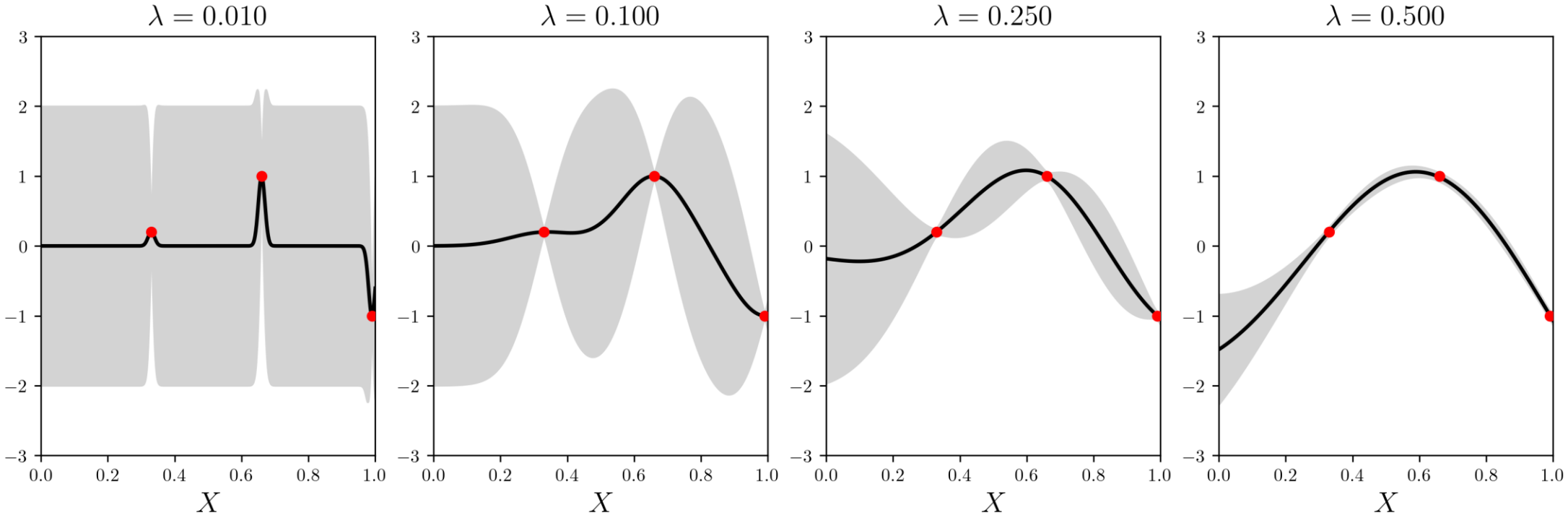
More noisy observations → Less confident

Posterior mean less obliged to hit the data → Reverts to prior mean = 0

Covariance matrix → Covariance function = Kernel function

- "A kernel is a function that defines the covariance between function values at any pair of input locations."

$$K(x_1, x_2) = \exp\left(-\frac{||x_2 - x_1||^2}{2\lambda^2}\right)$$

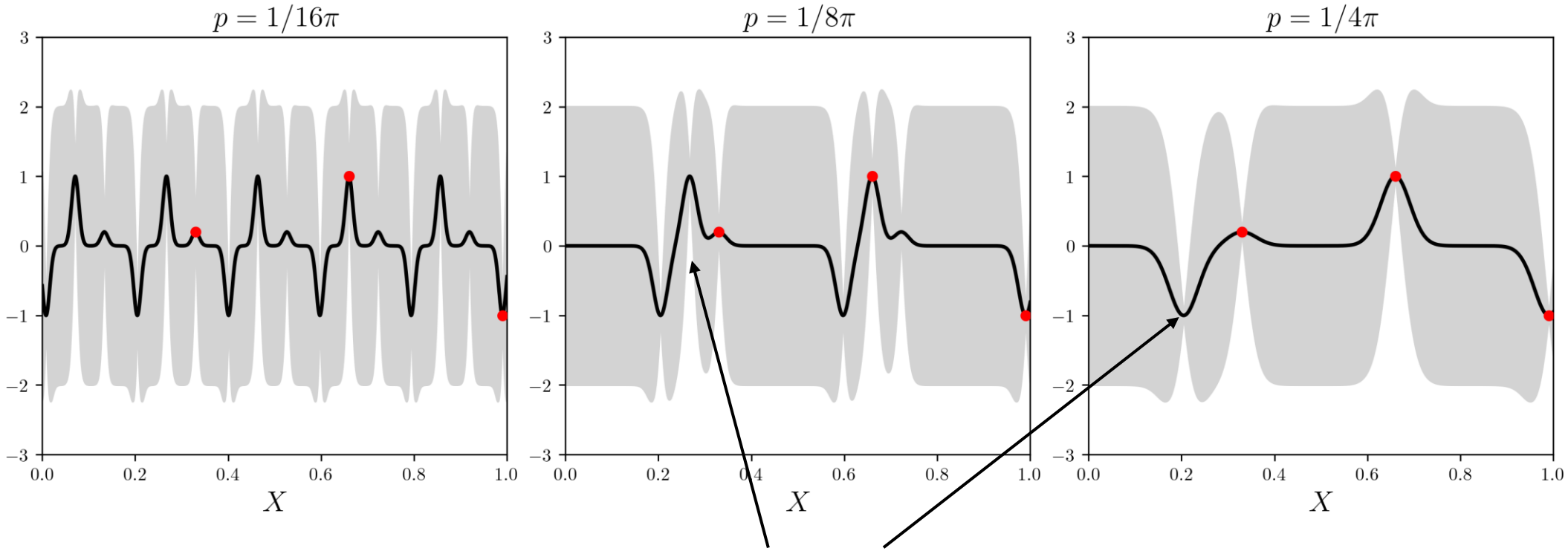


Longer length scale λ gives a mean that changes slowly and the "relative distance" to observation is smaller so less variance → Tighter confidence intervals.

Covariance matrix → Covariance function = Kernel function

- "A kernel is a function that defines the covariance between function values at any pair of input locations."

$$K(x_1, x_2) = \exp\left(-\frac{2}{\sigma^2}\sin^2\left(\frac{\pi||x_1 - x_2||}{p}\right)\right)$$ Periodic kernel

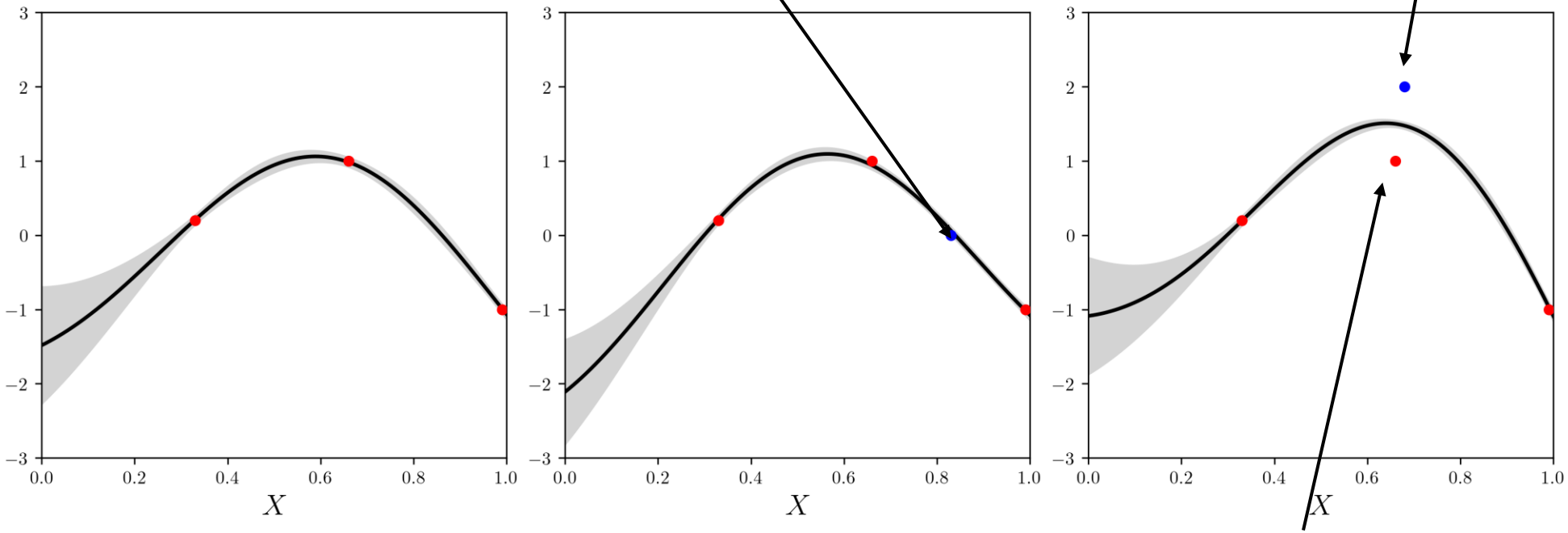


The periodic kernel introdues low variance regions where we don't have direct obsevations

$$K(x_1, x_2) = \exp\left(-\frac{||x_2 - x_1||^2}{2\lambda^2}\right)$$

Additional datapoint fits well

Additional datapoint fits poorly

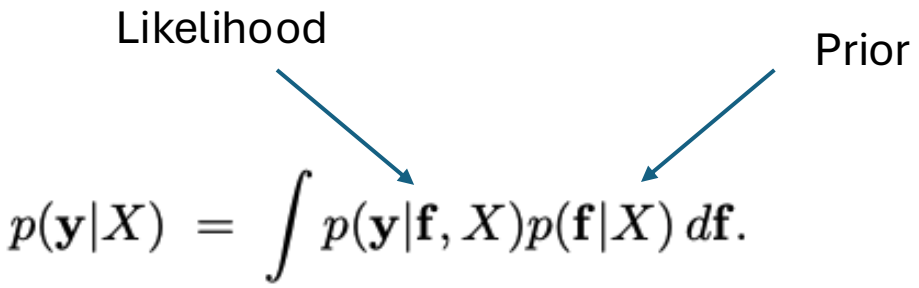


Distribution does not contain functions that can fit all the points

The kernel function has poor parameters!

Log marginal likelihood

- "Log-probability of observing the data under the GP model, integrating over all possible latent functions"



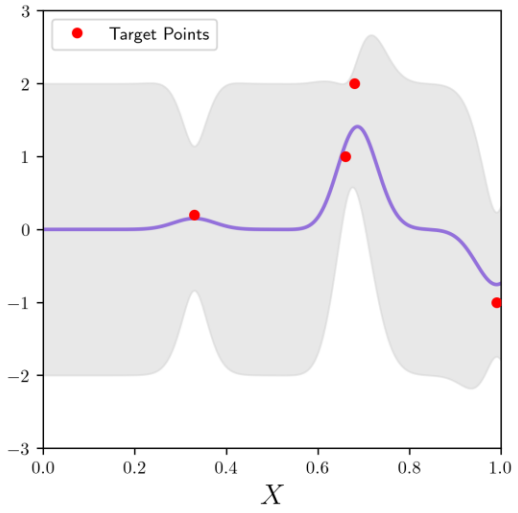
Again Gaussians have ~~magical~~ nice properties

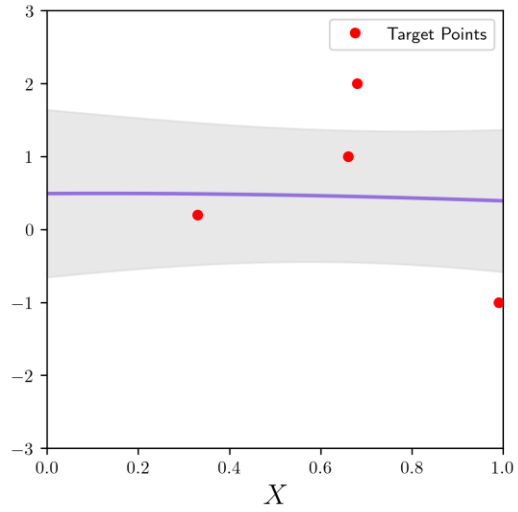
$$\log p(\boldsymbol{y}|X) = -\frac{1}{2}\boldsymbol{y}^T(K_\theta(X,X) + \sigma_n^2 I)^{-1}\,\boldsymbol{y}$$ Data fit

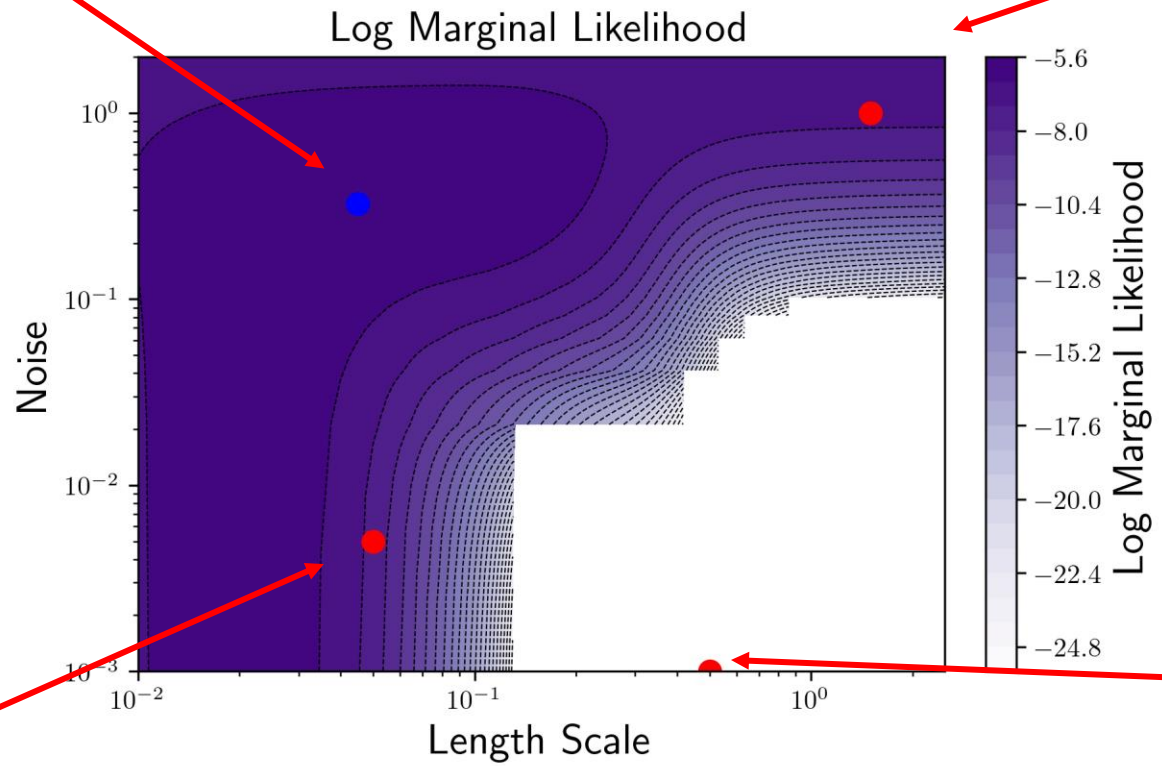
$$-\frac{1}{2}\log|K_\theta(X,X) + \sigma^2 I|$$ Model complexity

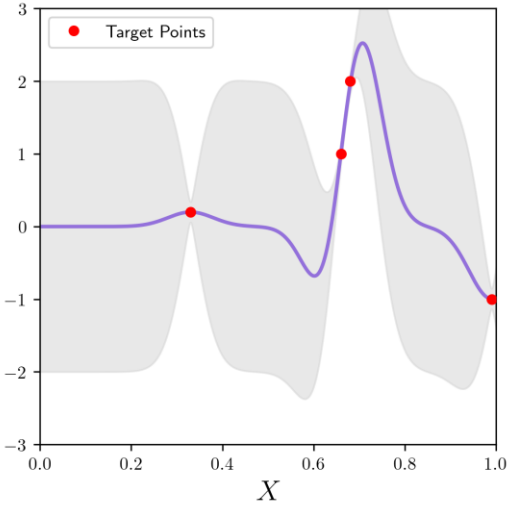
$$-\frac{n}{2}\log 2\pi$$ Normalization

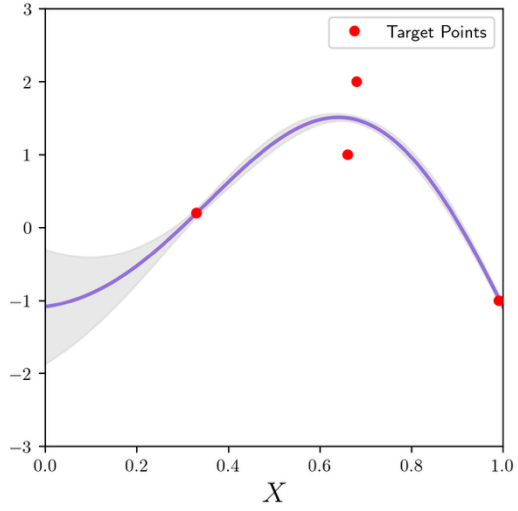
$$\log p(\boldsymbol{y}|X) = -\frac{1}{2}\boldsymbol{y}^T(K_\theta(X,X) + \sigma_n^2 I)^{-1}\boldsymbol{y}$$
$$-\frac{1}{2}\log|K_\theta(X,X) + \sigma^2 I| - \frac{n}{2}\log 2\pi$$



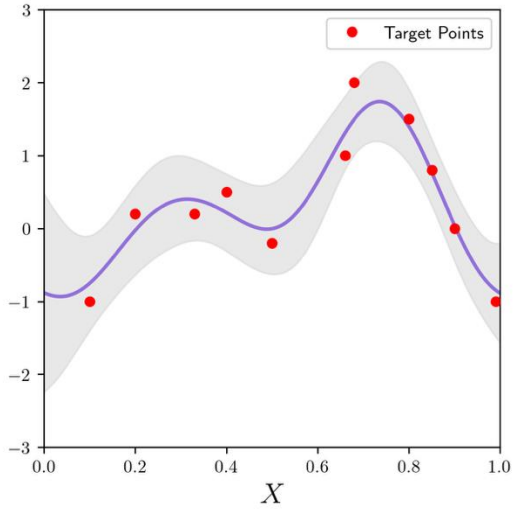


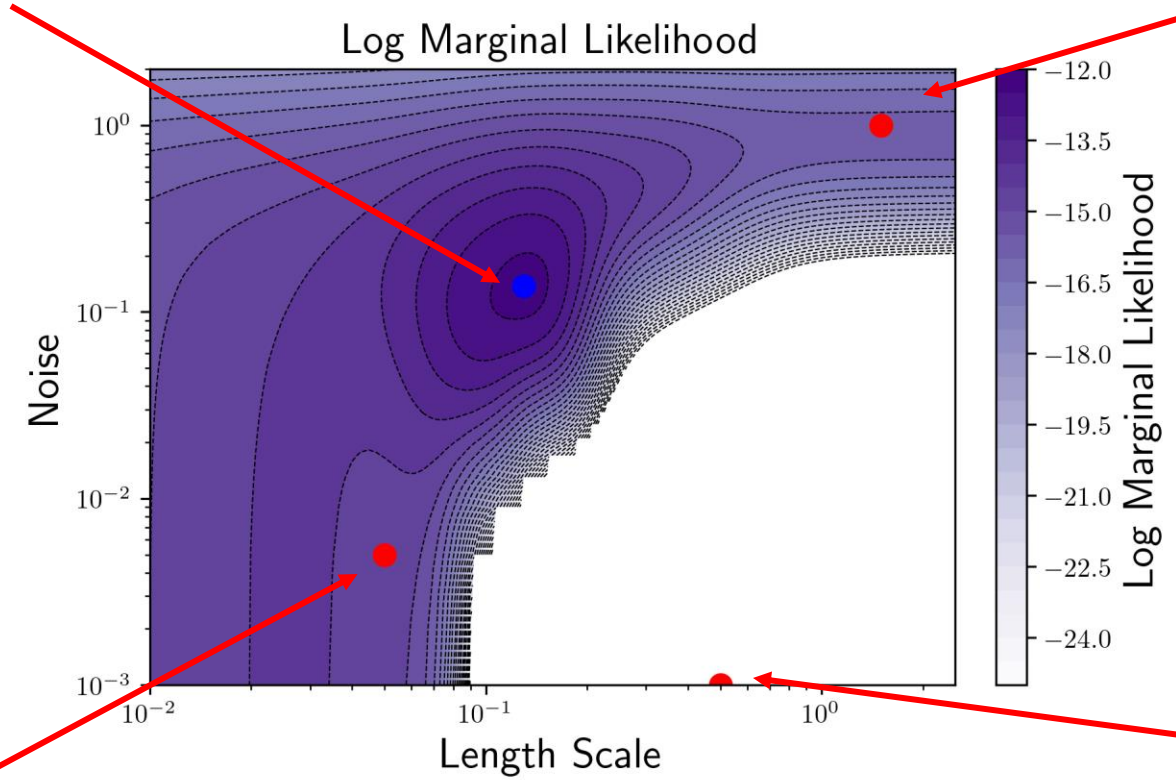


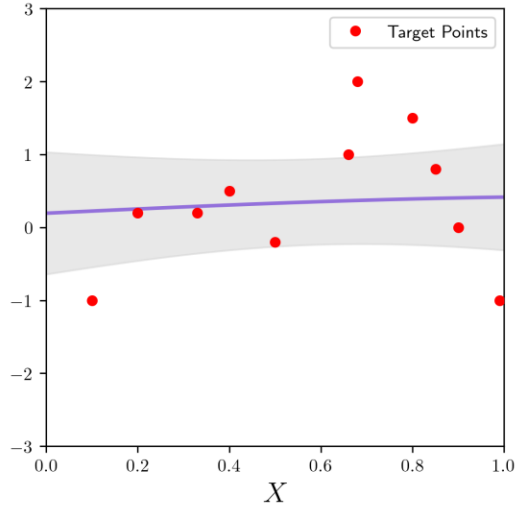


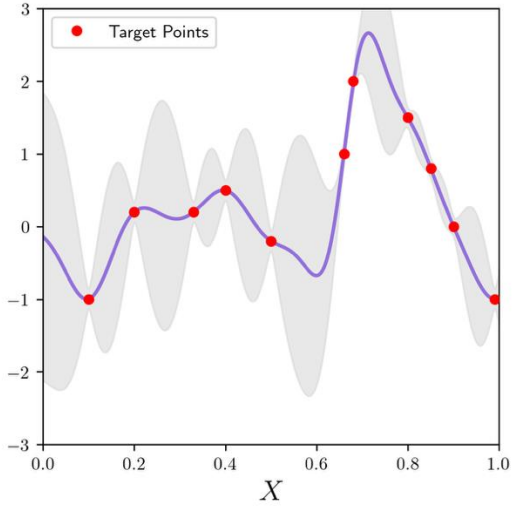


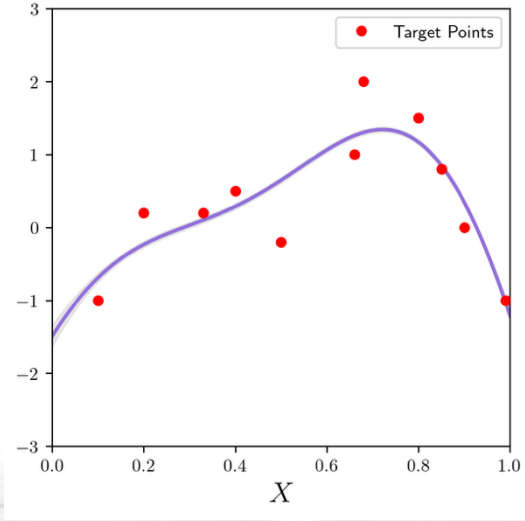
$$\log p(\boldsymbol{y}|X) = -\frac{1}{2}\boldsymbol{y}^T(K_\theta(X,X) + \sigma_n^2 I)^{-1}\boldsymbol{y}$$
$$-\frac{1}{2}\log|K_\theta(X,X) + \sigma^2 I| - \frac{n}{2}\log 2\pi$$

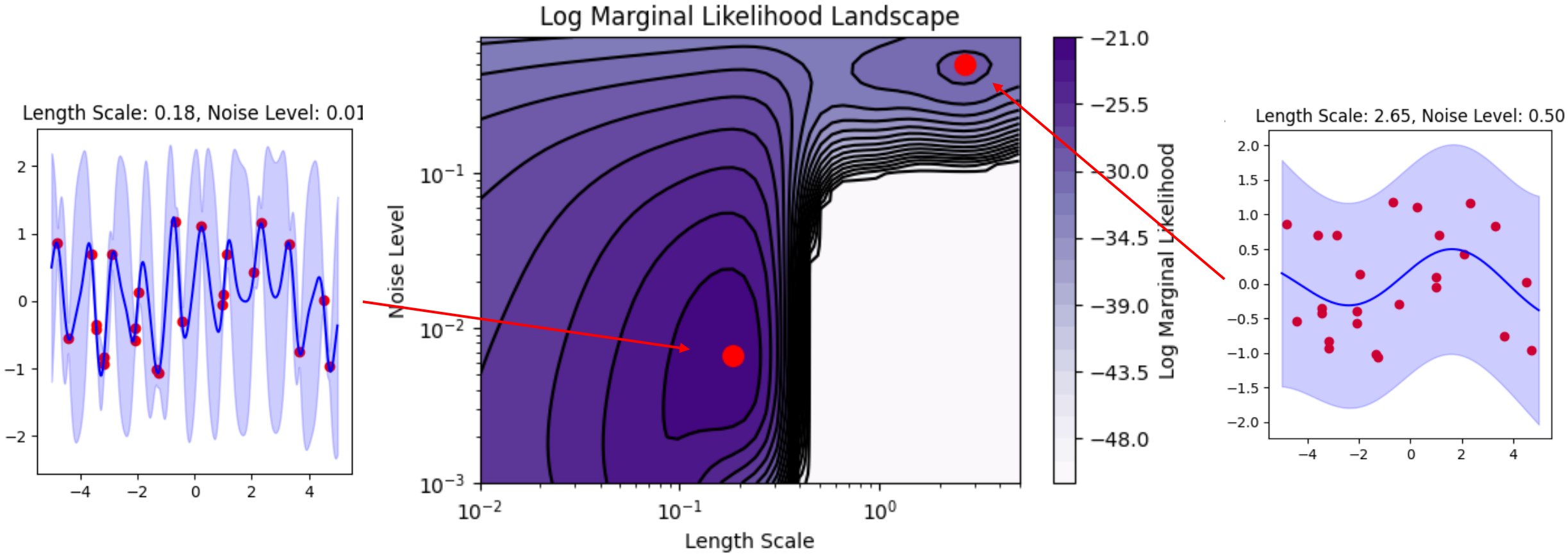








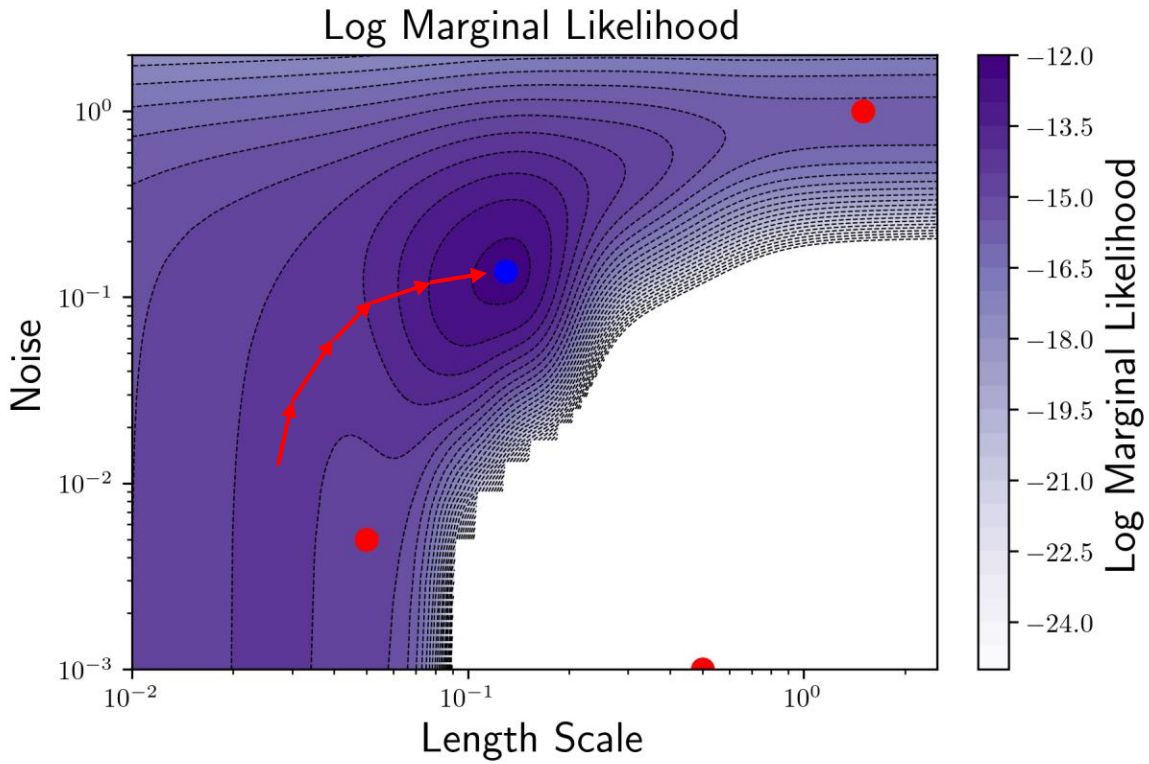



Length Scale: 0.18, Noise Level: 0.01
Log Marginal Likelihood Landscape
Noise Level
Length Scale
Log Marginal Likelihood
Length Scale: 2.65, Noise Level: 0.50

LML typically optimized using gradient-based approach

$$\nabla_\theta \log p_\theta(\boldsymbol{y}|X)$$

Therefore, I recommend using a auto-differentiation framework for GPs (e.g. PyTorch or Jax)



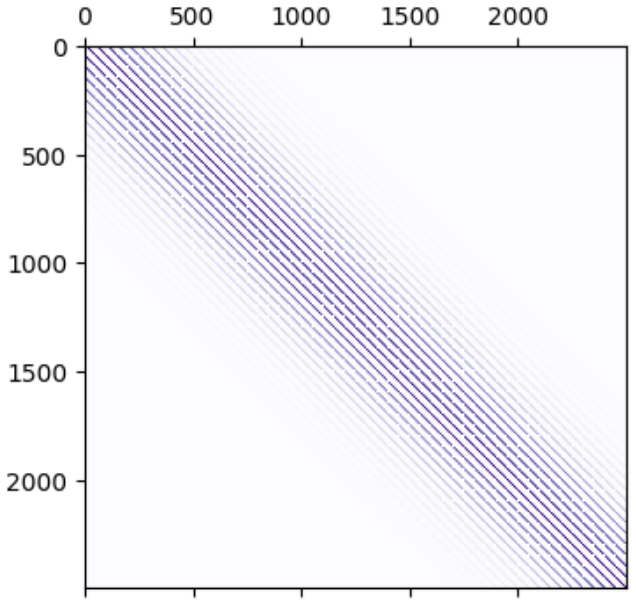
Functions of multiple variables

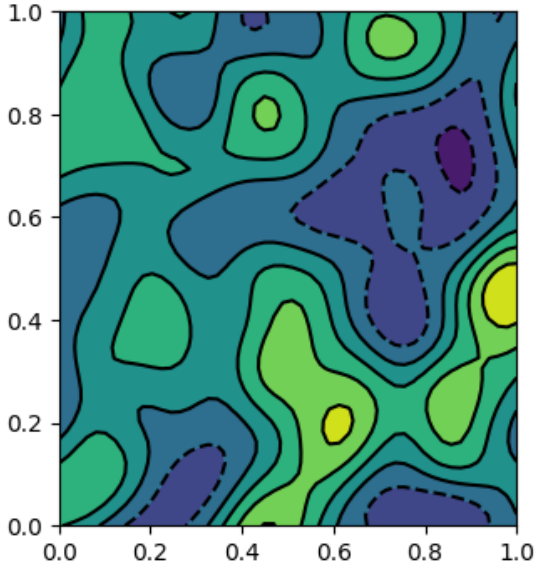
- All handled by the kernel – and all the kernels we've seen can do so.

$$X = [x_1, x_2, \dots, x_n] \rightarrow X = [(x_1, y_1), (x_2, y_2), \dots (x_N, y_N)] = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots \boldsymbol{x}_N]$$

$$K(x_1, x_2) = \exp\left(-\frac{||\boldsymbol{x}_2 - \boldsymbol{x}_1||^2}{2\sigma^2}\right)$$

RBF Covariance of a grid of points in [0, 1] for both x and y.

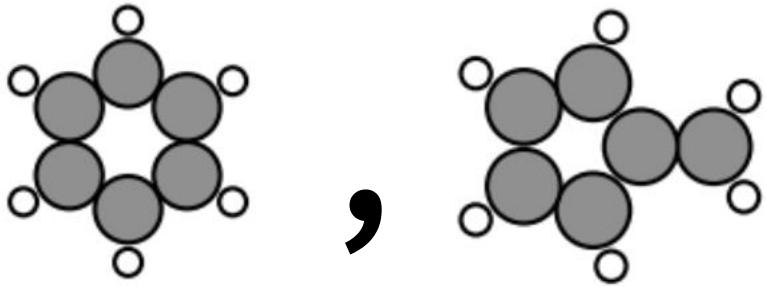




Function sampled from the GP prior.

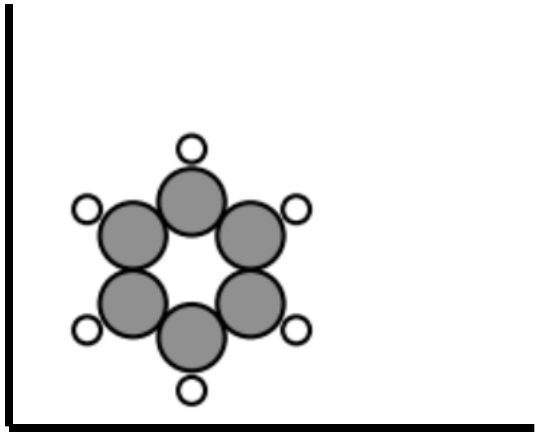
Gaussian Processes & Regressors
For Materials Science / Astrochemistry

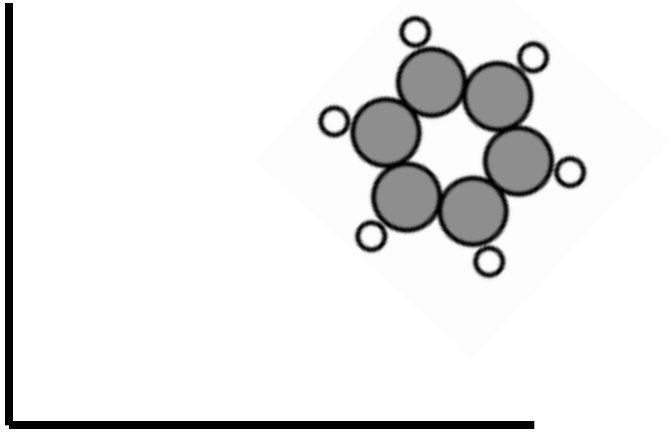
Covariance of atomic configurations.

$K($  $)$

Structural descriptor

➢ A representation of an atomic structure that encodes invariances.





Invariances

- ❖ Translation
- ❖ Rotation
- ❖ Permutation

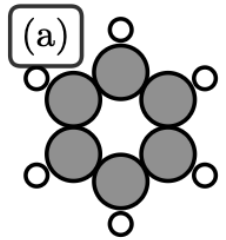
Cartesian coordinates are terrible at this.

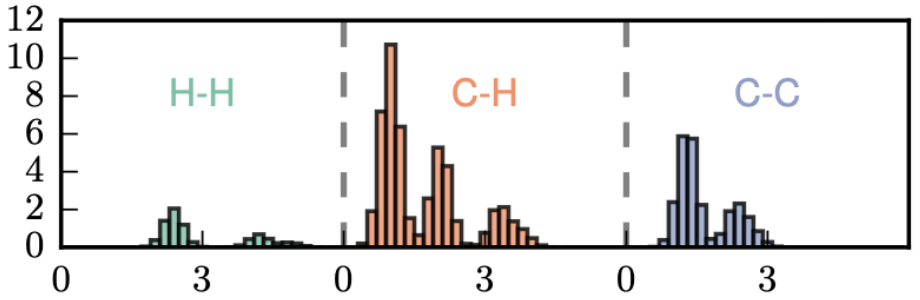
x_1	y_1	z_1
x_2	y_2	z_2
…	…	…
x_N	y_N	z_N

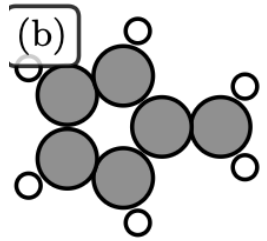
Fingerprint descriptor

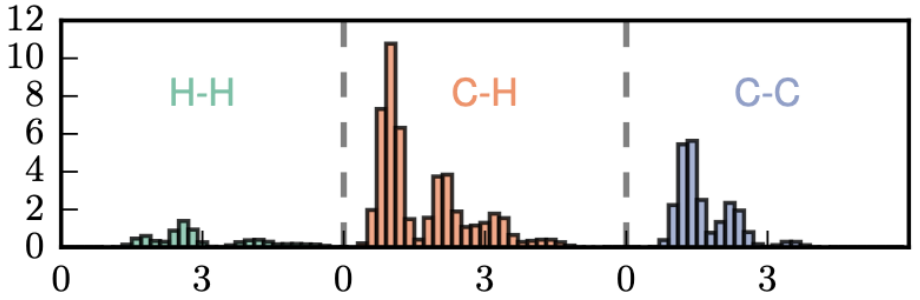
- A smooth histogram of bond lengths (and angles) representing an atomic configuration as a vector.

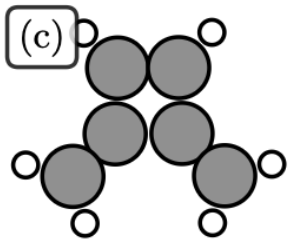
$$F_{A,B}(r) = \frac{V}{4\pi N_A N_B \Delta} \sum_i \sum_j \frac{1}{r_{ij}} \exp\left(-\frac{1}{2}\frac{(r - r_{ij}^2)}{\sigma}\right).$$

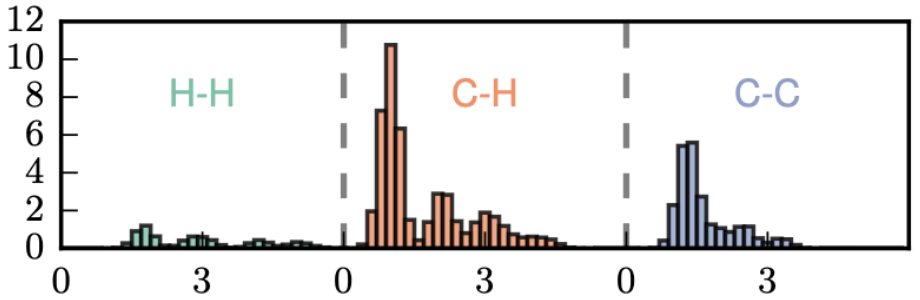








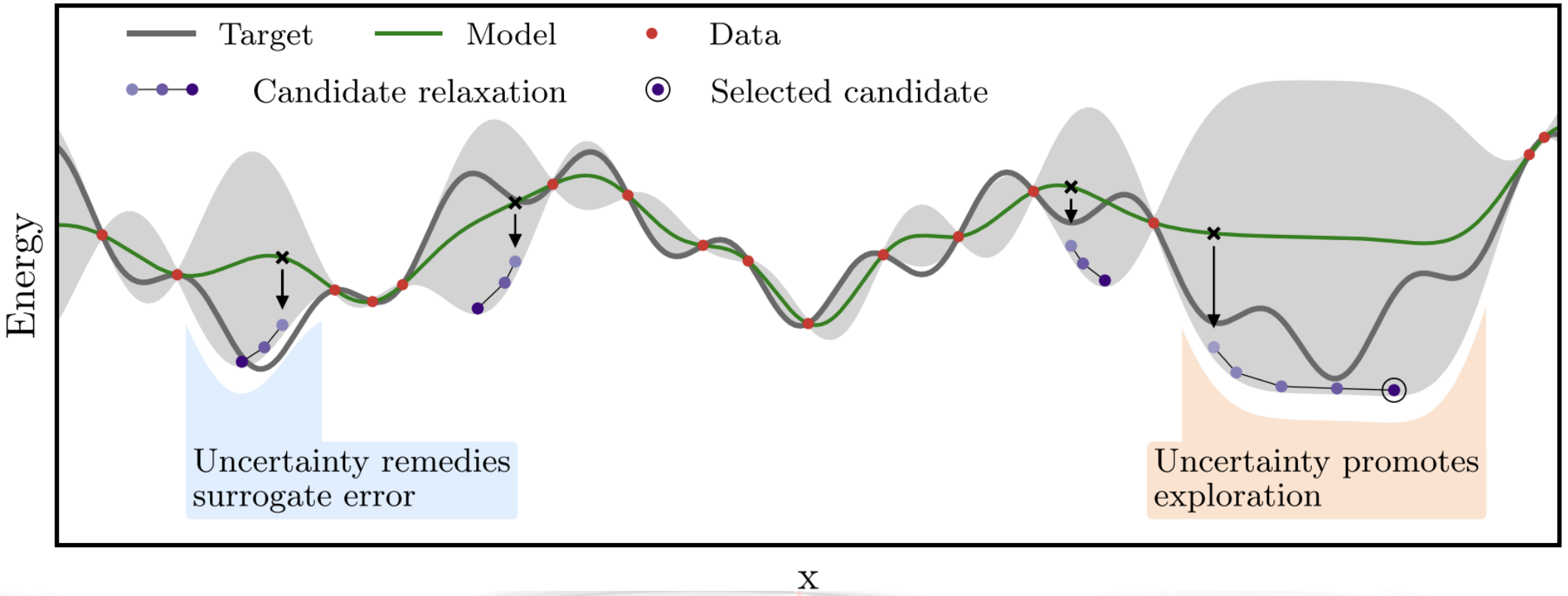


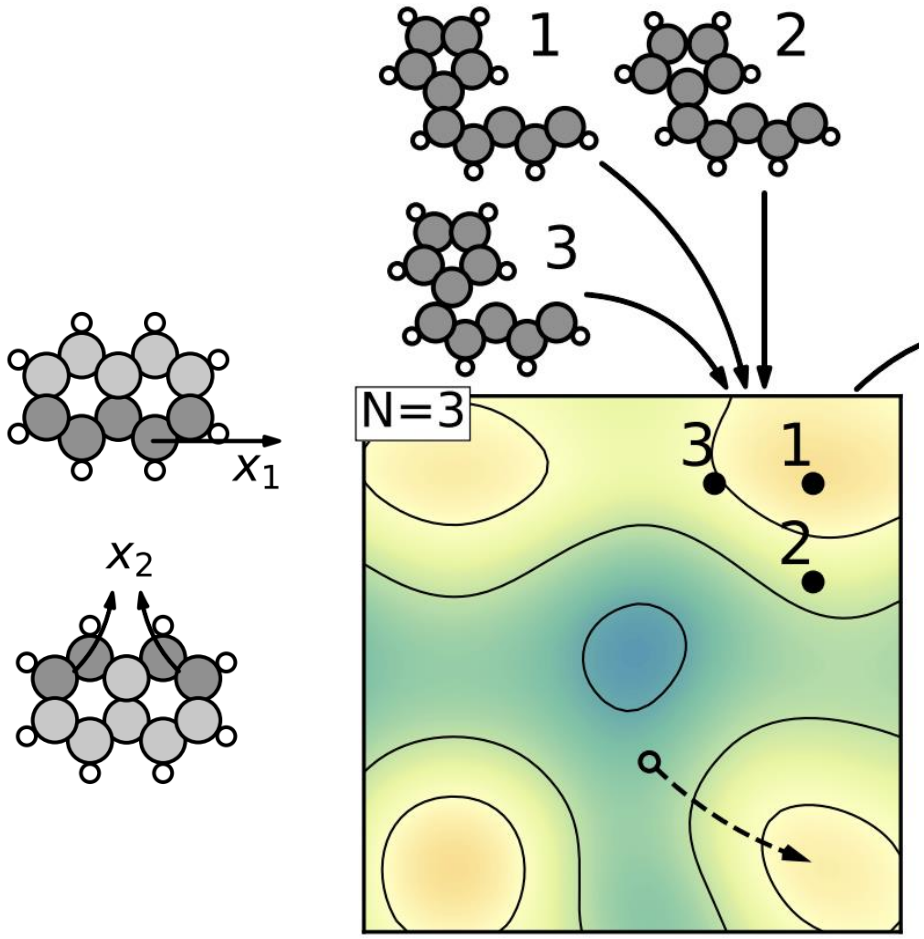


Bayesian Optimization

- Acquisition function based on the GP decides where the target function should be queried.

$$f(X) = \mu(x) - \kappa\sigma(x)$$



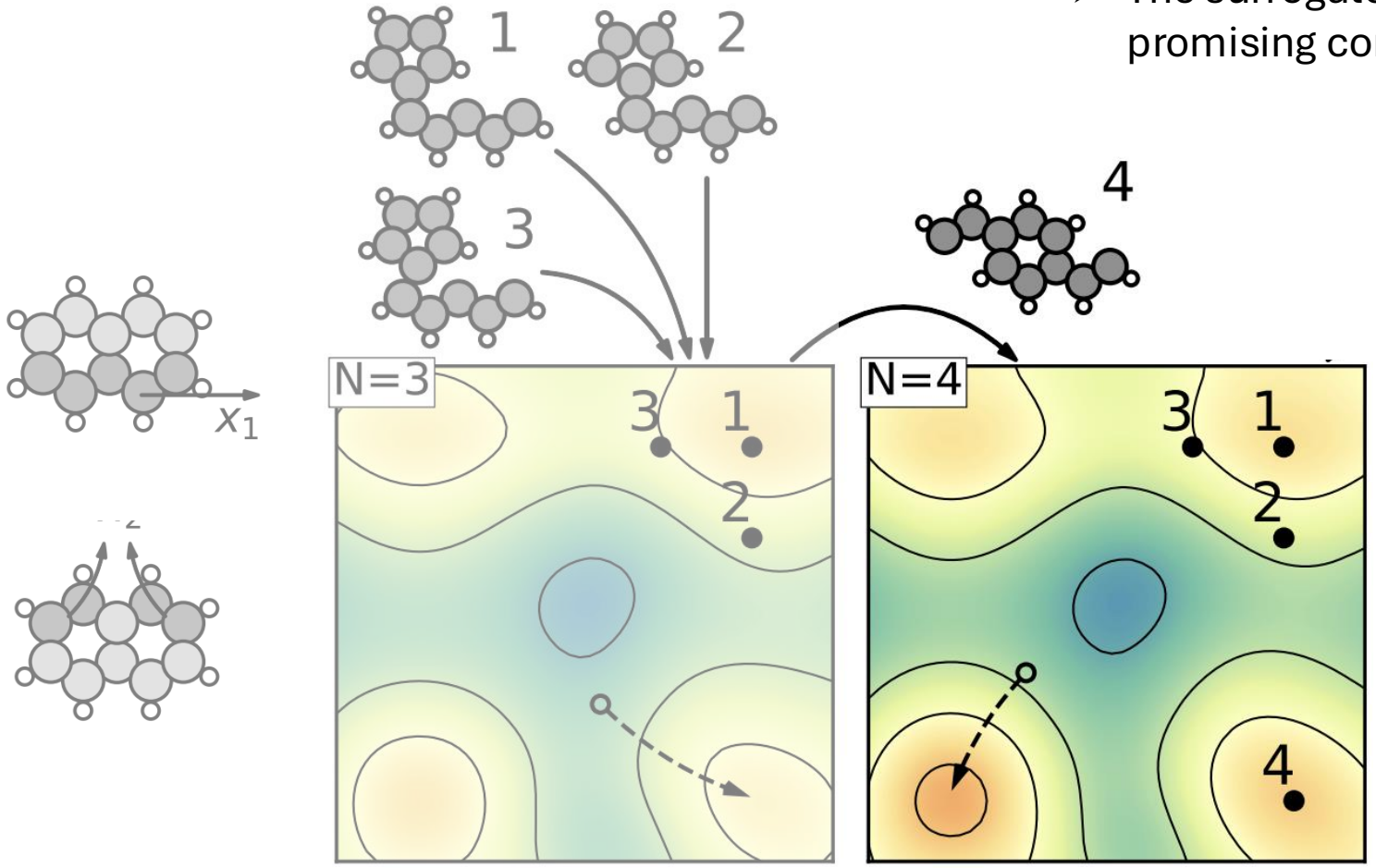


Bayesian Optimization

- The surrogate GP landscape is used to find promising configurations to calculate with DFT.

Bayesian Optimization

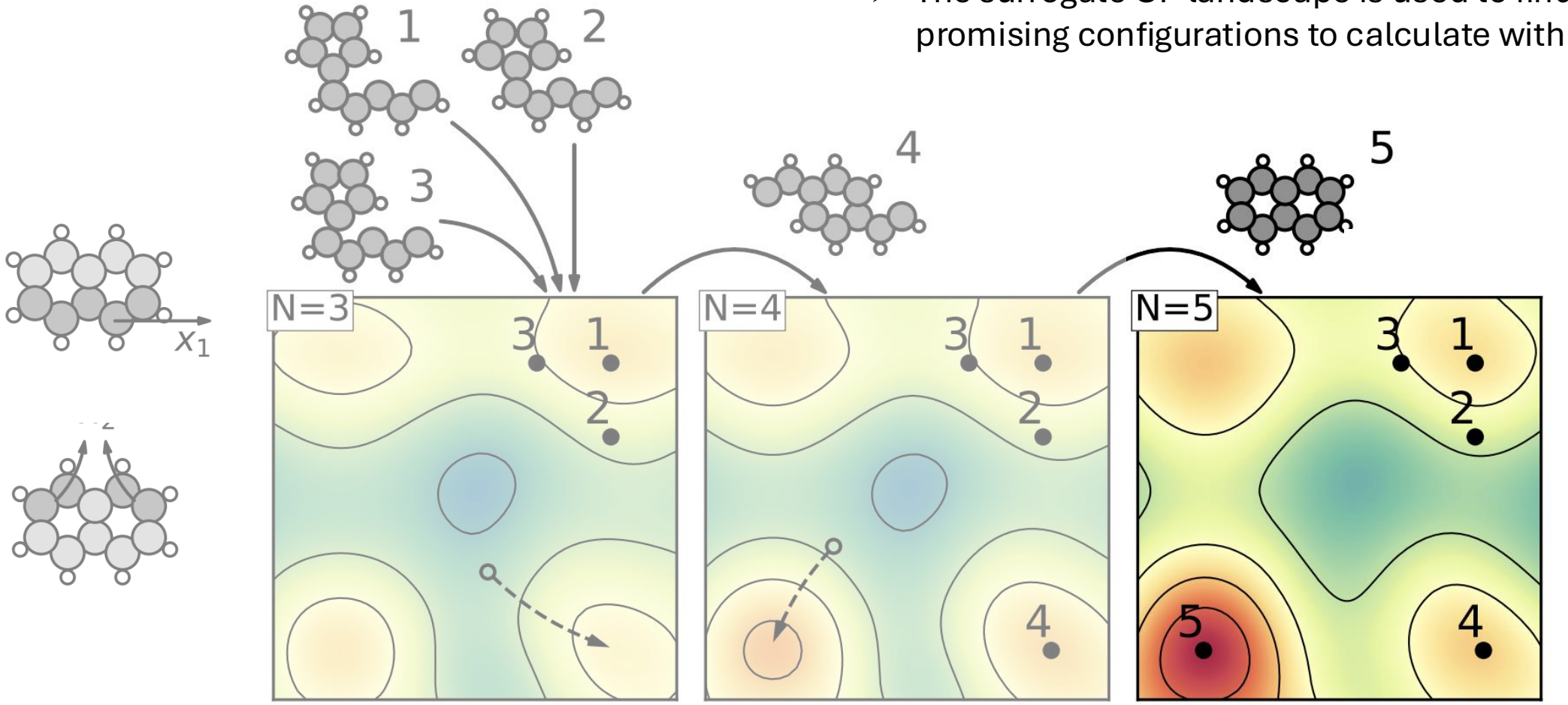
- The surrogate GP landscape is used to find promising configurations to calculate with DFT.



Phys. Rev. Lett. **124**, 086102

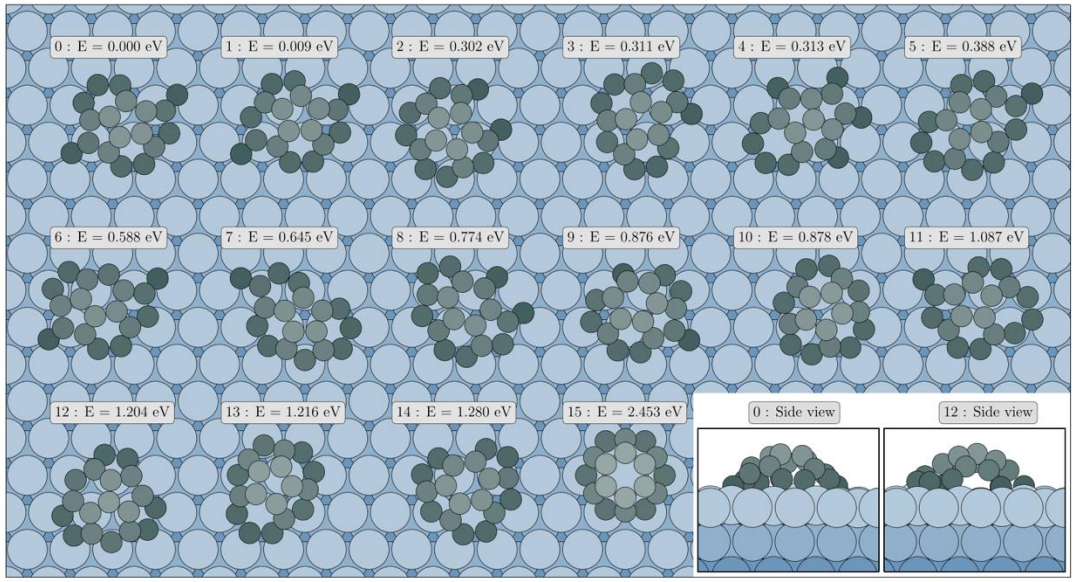
Bayesian Optimization

- The surrogate GP landscape is used to find promising configurations to calculate with DFT.



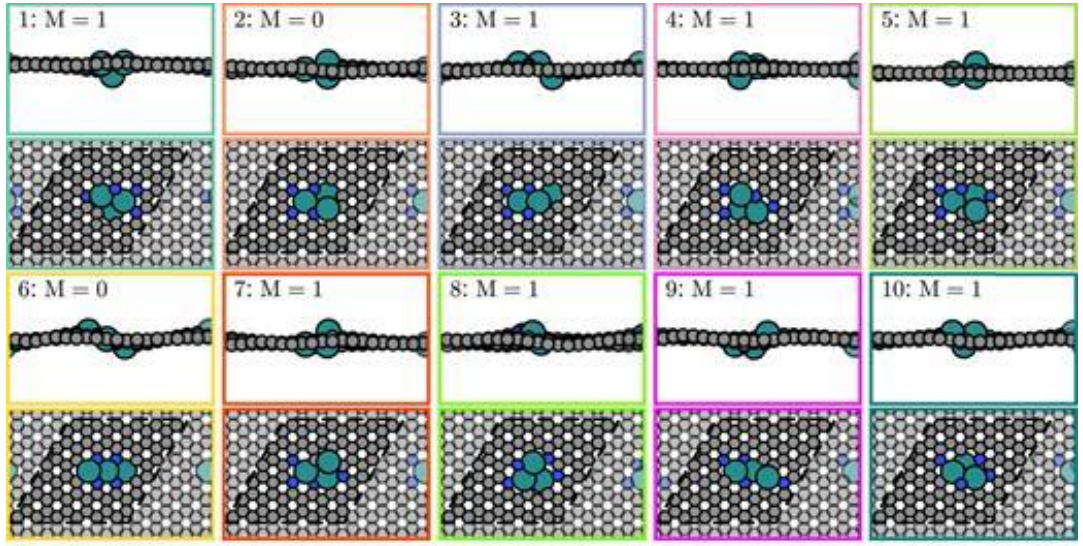
Phys. Rev. Lett. **124**, 086102

Carbon clusters on Ir(111)



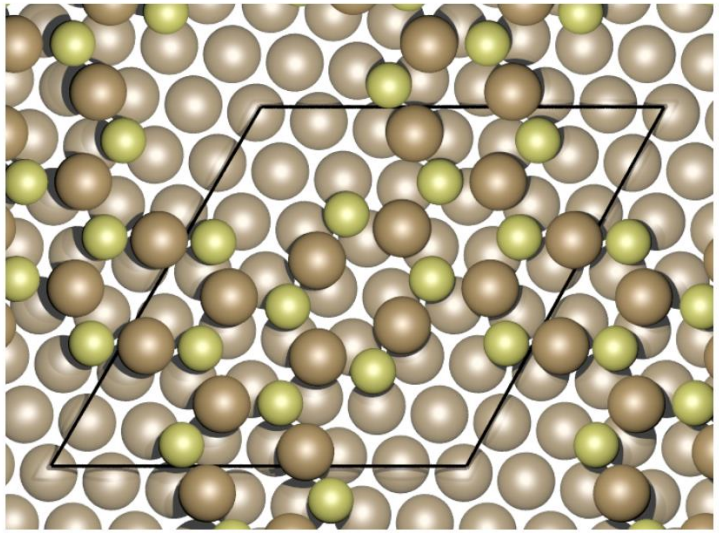
Phys. Rev. B **105**, 245404

Nitrogen-Ruthenium defects in graphene



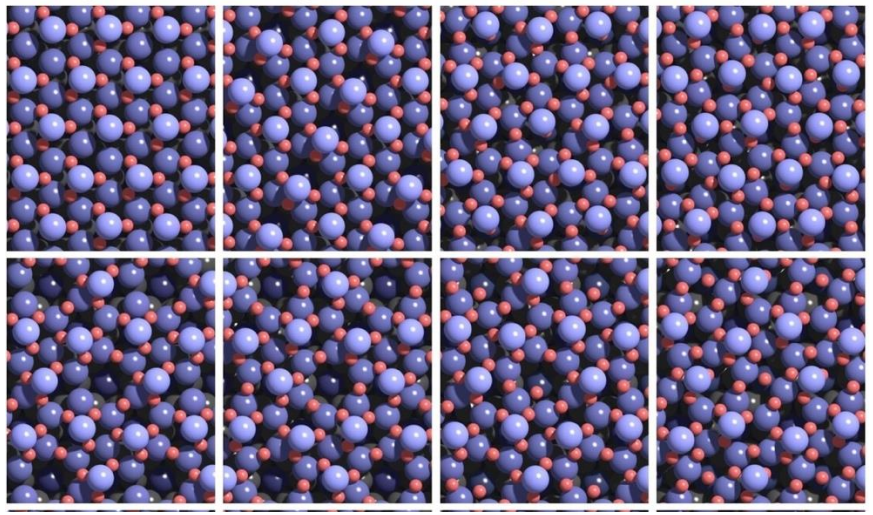
J. Chem. Phys. 157, 054701 (2022)

Sulfur-induced Cu(111) reconstruction



J. Chem. Phys. 160, 174107 (2024)

Tin-oxide on $Pt_3Sn(111)$

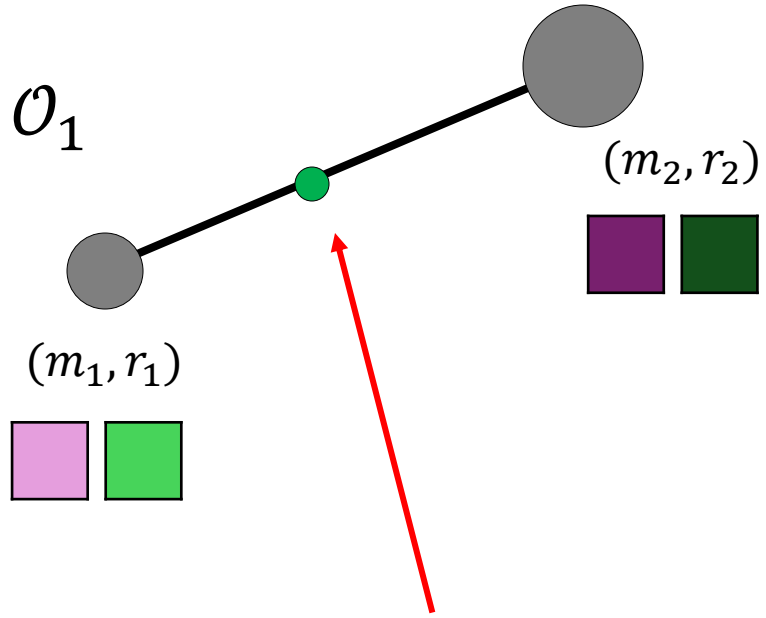


Angew. Chem. Int. Ed. **2022**, *61*, e202204244

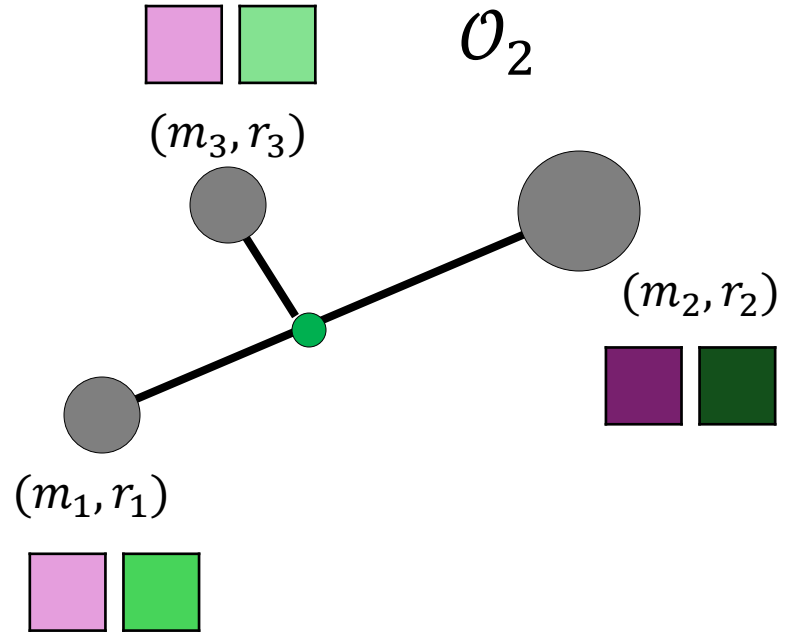
Moment of inertia

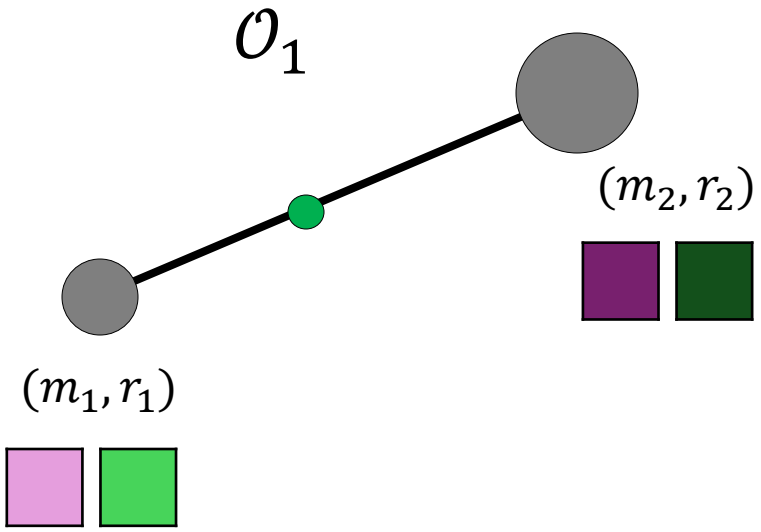
$$I = \sum_i m_i r_i^2$$

How to calculate the covariance between these $K(\mathcal{O}_1, \mathcal{O}_2)$**?**



Center of rotation.

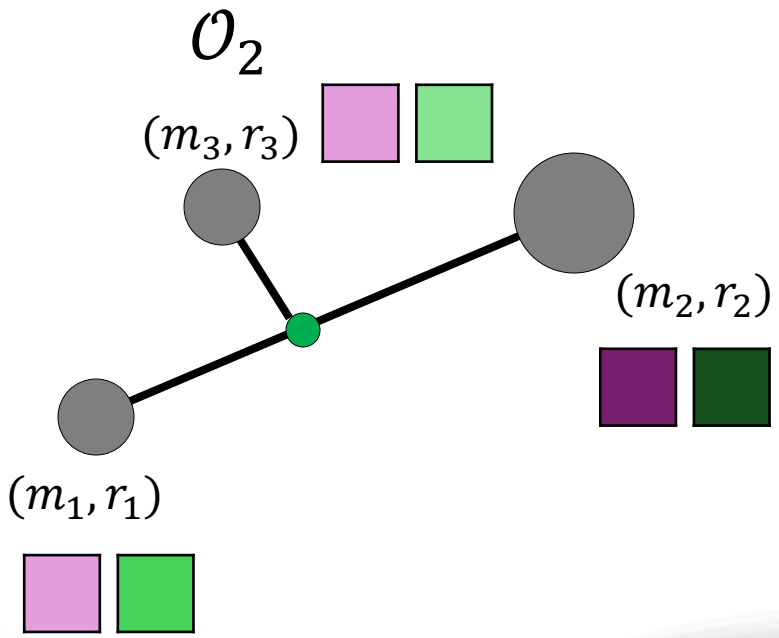




$$X_1 =$$

$$X_2 =$$

$$K(X_1, X_2) =$$

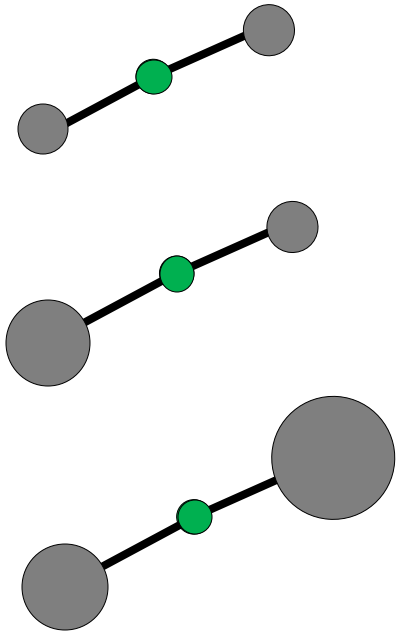


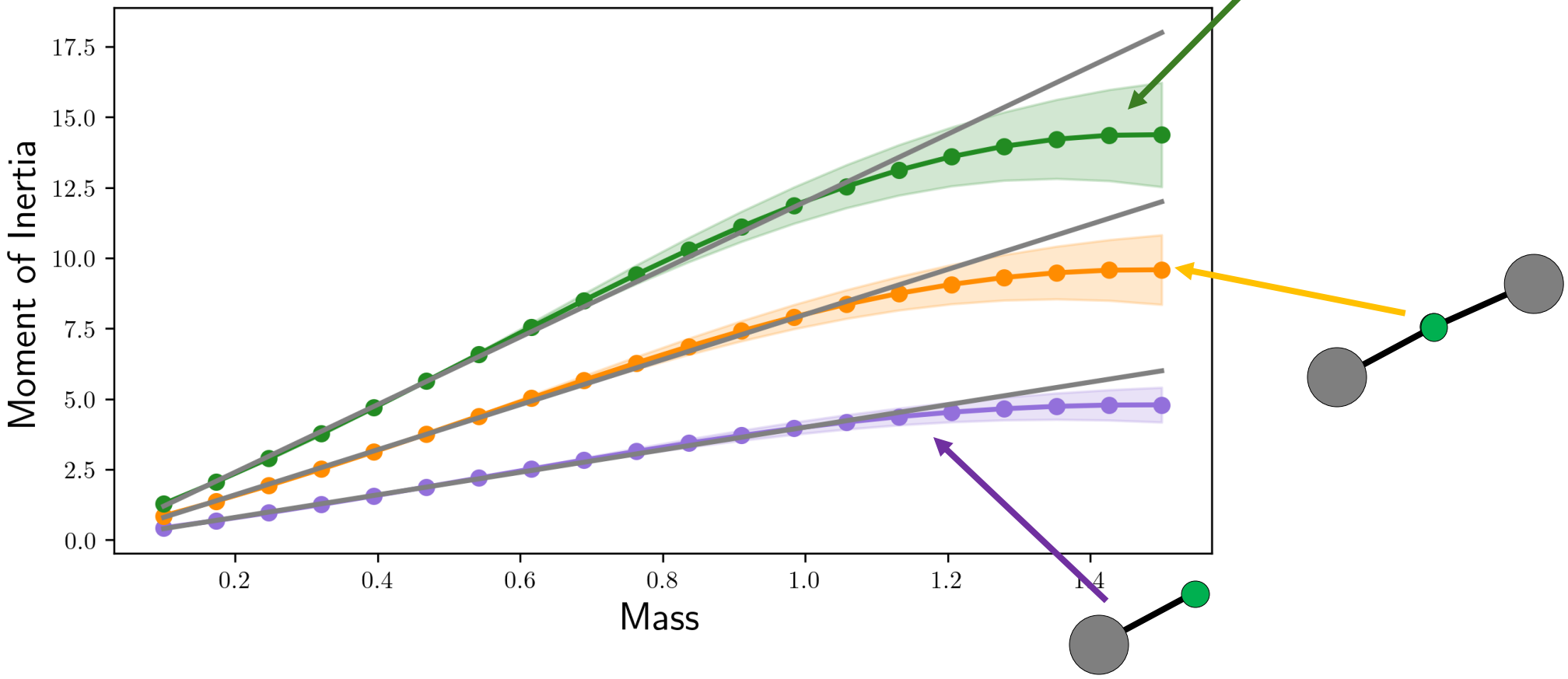
$$K(\mathcal{O}_1, \mathcal{O}_2) = \begin{array}{|c|c|}\hline 1 & 1 \\ \hline \end{array}$$

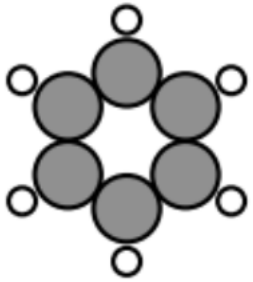
$$= \begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline 1 \\ \hline \end{array}$$

$$K(\mathcal{O}_1, \mathcal{O}_2) = L_1 K(X_1, X_2) L_2^T$$

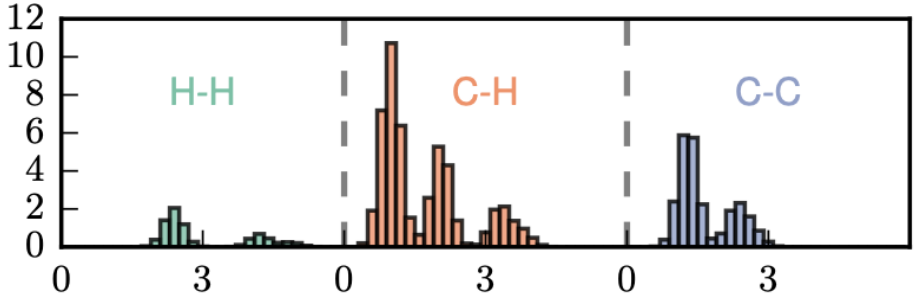
Training examples

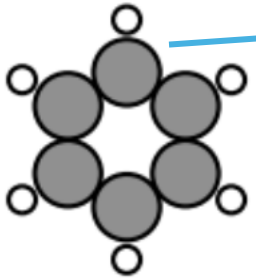






One vector

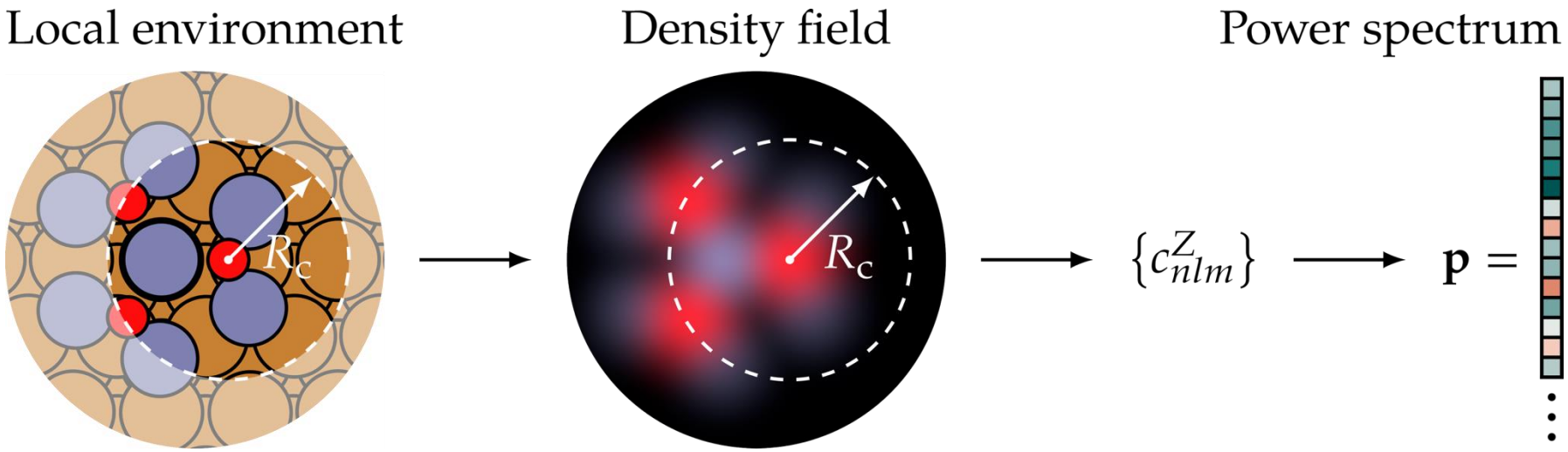




N_{atoms} vectors

Smooth Overlap of Atomic Positions (SOAP)

- Power spectrum of the smoothend atomic density around an atom yields an invariant descriptor.

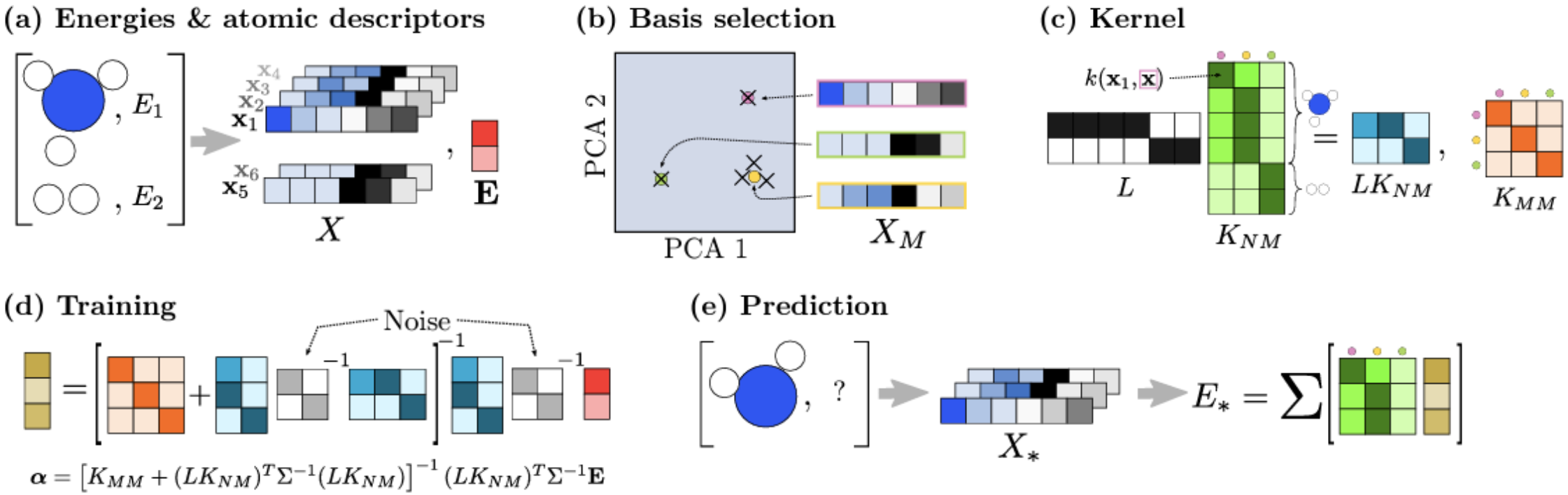


Phys. Rev. B **87**, 184115, on SOAP

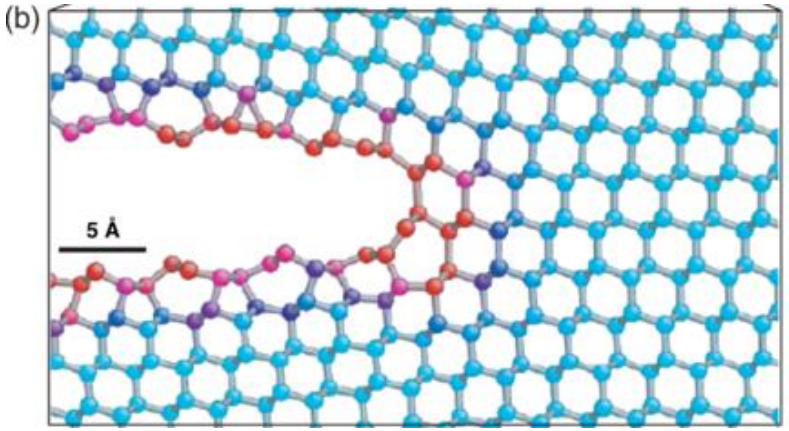
Figure from "Leveraging machine learning for materials discovery: Inverse catalysts for CO_2 hydrogenation", Luuk Kempen

Sparse Gaussian Processes

- For large (>1-5k) datasets regular GPR becomes prohibitively expensive, $\mathcal{O}(n^3)$ training time and $\mathcal{O}(n)$ inference time. Sparse GPRs can be constructed with $\mathcal{O}(m^2 n)$ training and $\mathcal{O}(m)$ inference where $m \ll n$.

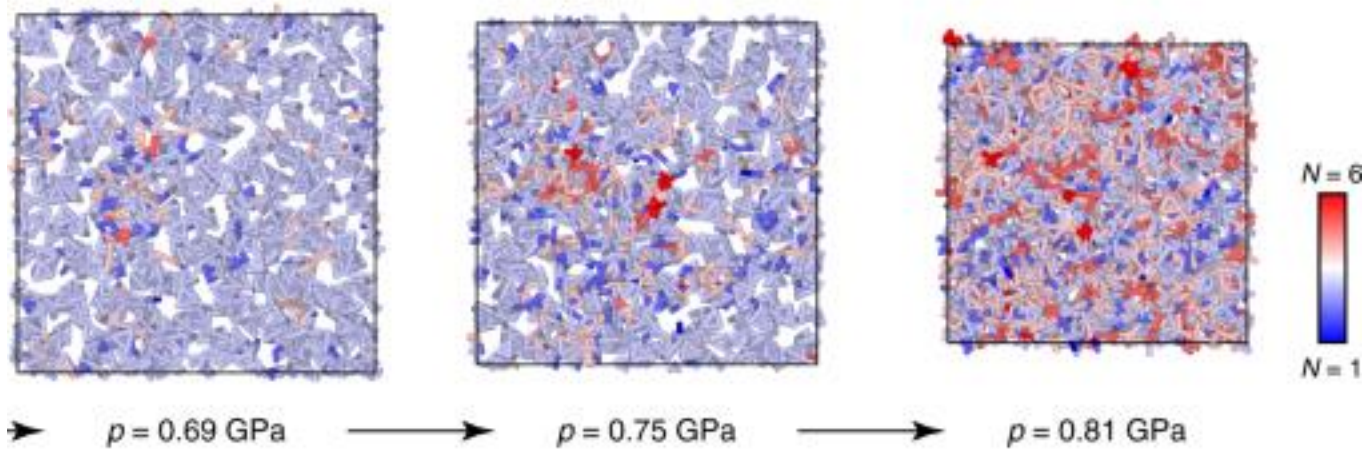


General purpose potential for Si



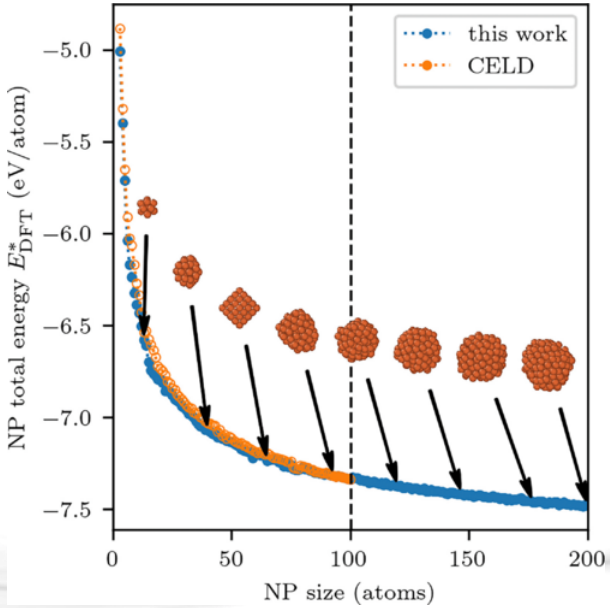
Phys. Rev. X **8**, 041048

General purpose potential for phosphorus



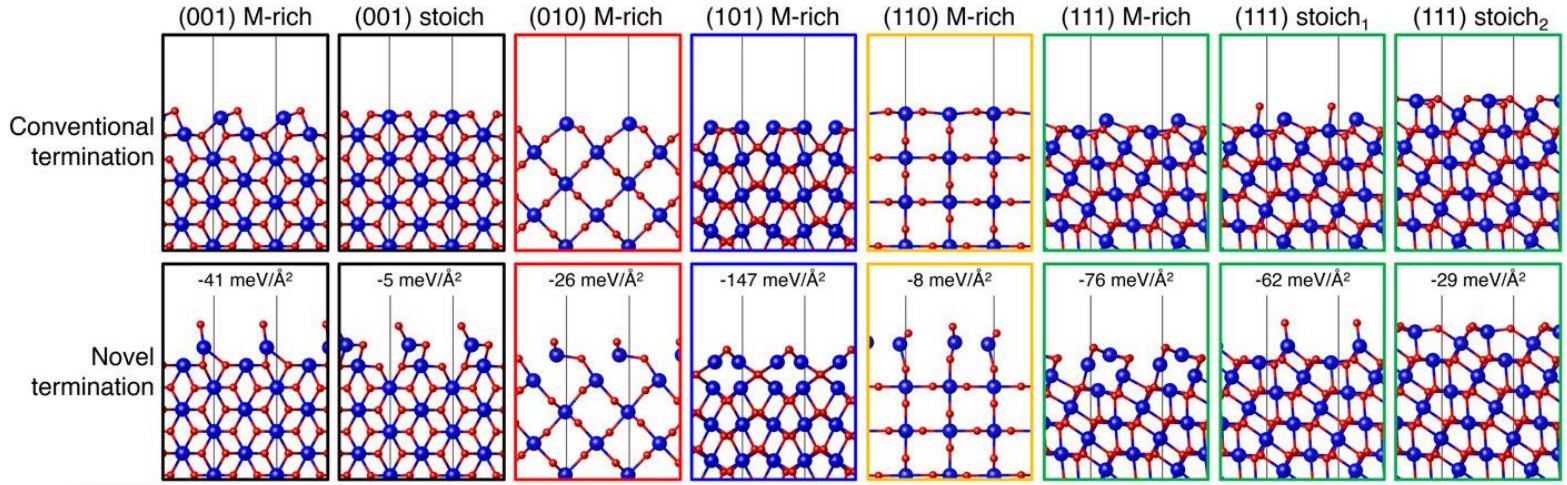
Nat Commun **11**, 5461 (2020)

Iron nanoparticles



Phys. Rev. B **107**, 245421

Surface structure determination of IrO_2.



J. Chem. Phys. 155, 244107 (2021)

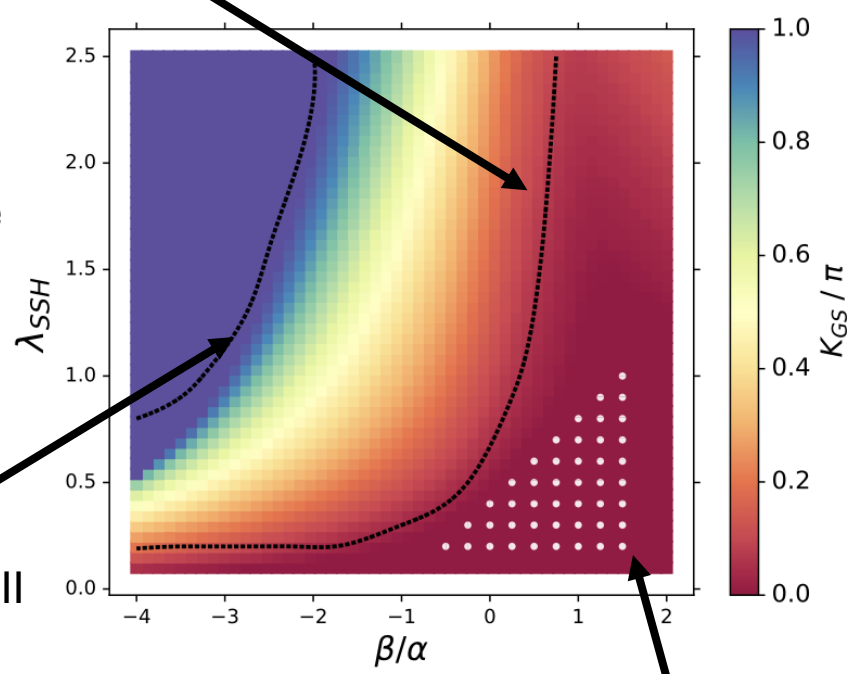
Phase transition predictions

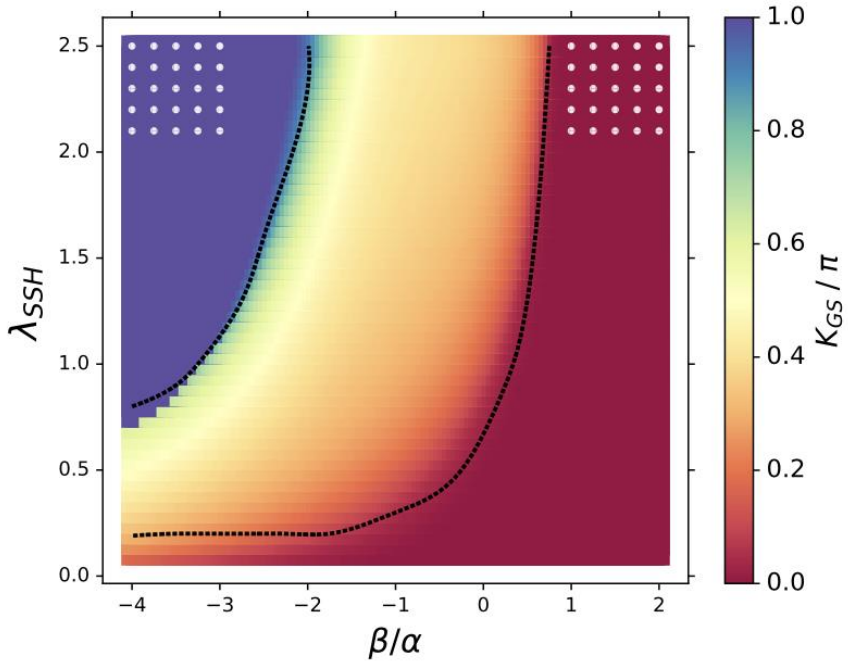
- Optimized kernel allows GP model to predict two phase transitions while only having seen data from one phase.

Phase transition I

Both solid lines are from reference calculations

Phase transition II



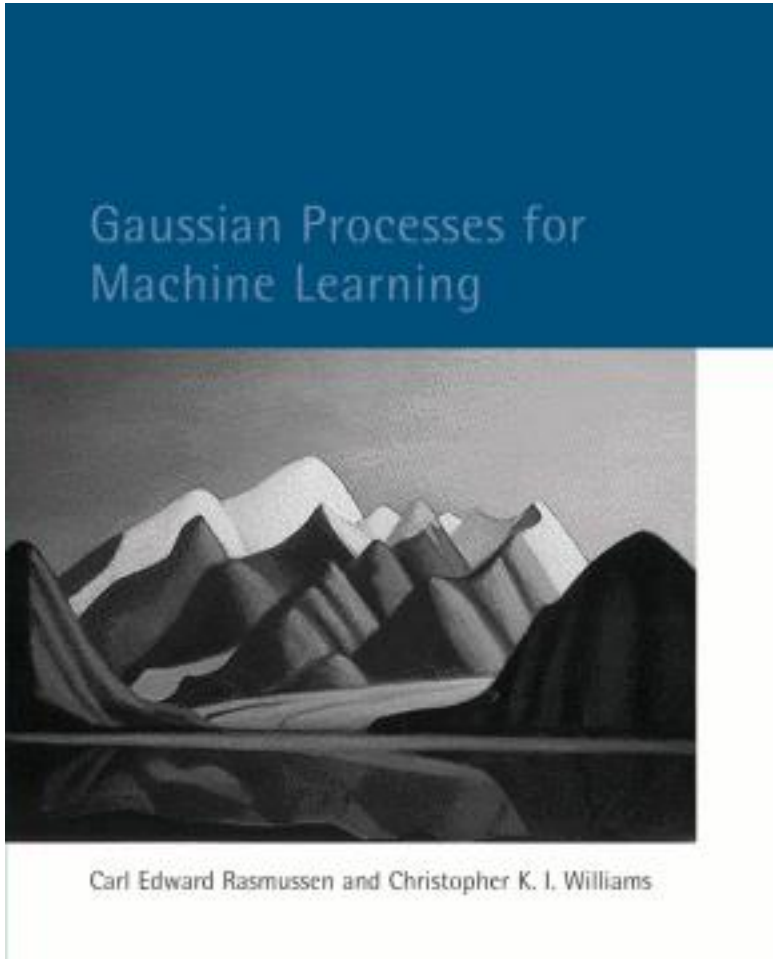


Observations

PRL 121, 255702 (2018)

Gaussian Processes for Machine Learning

- The bible of GPs for ML. PDF freely available from https://gaussianprocess.org/gpml/



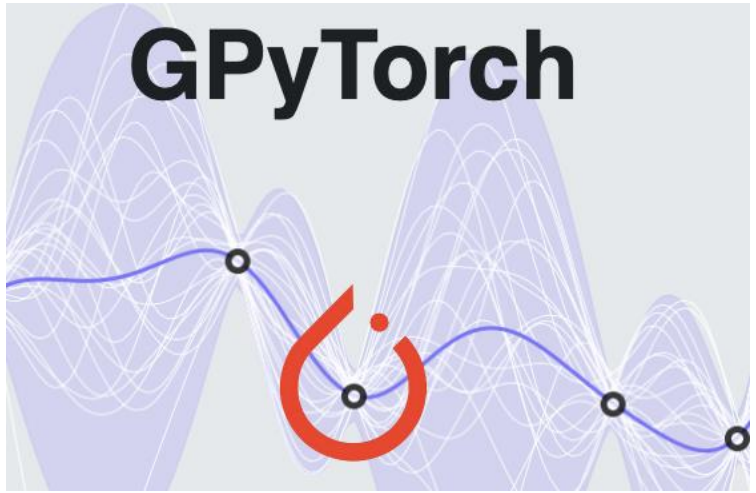
Packages / Libraries:

- Scikit-learn
  - Numpy implementation of GP regression.
- GPyTorch
  - Torch.
- GPJax
  - Jax.
- GPFlow
  - Tensorflow.

Among many others…

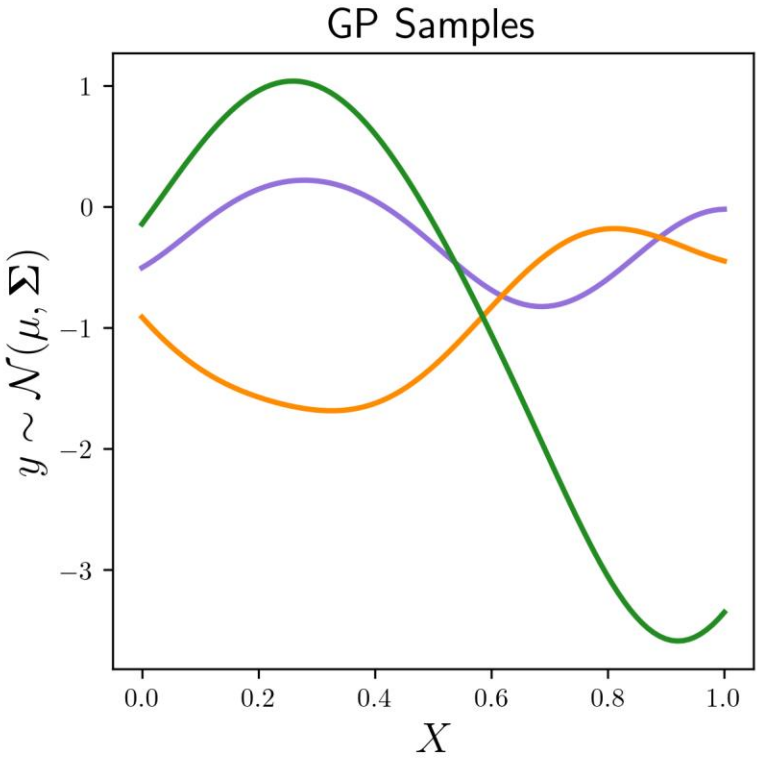
Recommendation

Start with your own code, read the scikit learn implementation if something is tricky (Or from my tutorial on Friday).
If you need something more feature rich or efficient try GPJax or GPyTorch depending on which backend your more comfortable with.



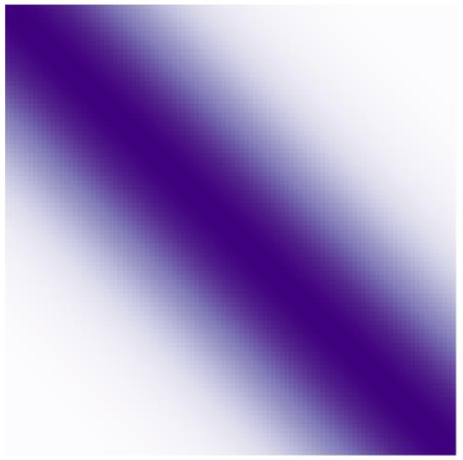
Recap

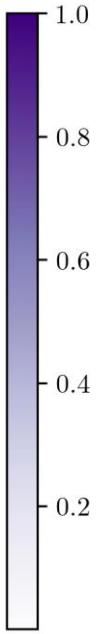
- Gaussian Processes are distributions over functions.
  Uniquely defined by a mean and a covariance function



~~Covariance Matrix~~ Σ

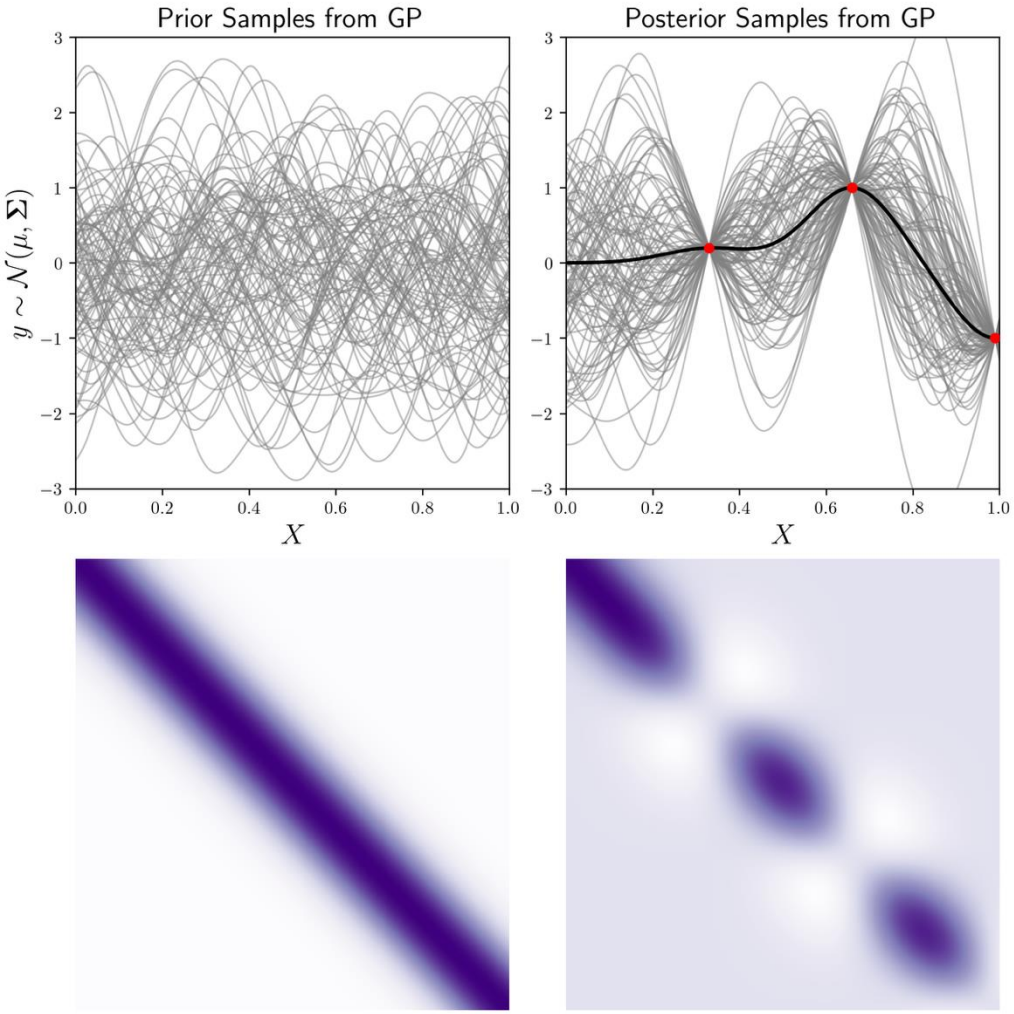
Covariance function





Recap

- Conditioning tightens the distribution such that only relevant functions are kept.



Acknowledgements

Funded by the European Union

• This work is based upon work from COST Action CA21126 - Carbon molecular nanostructures in space (NanoSpace), supported by COST (European Cooperation in Science & Technology).

cost
EUROPEAN COOPERATION IN SCIENCE & TECHNOLOGY

• COST (European Cooperation in Science & Technology) is a funding agency for research and innovation networks. Our Actions help connect research initiatives across Europe and enable scientists to grow their ideas by sharing them with their peers. This boosts their research, career and innovation.

NanoSpace

• www.cost.eu
• https://www.cost.eu/actions/CA21126/
• https://research.iac.es/proyecto/nanospace/