IOCB
PRAGUE

Knowing When to Trust Machine Learning Models: A Bayesian Perspective

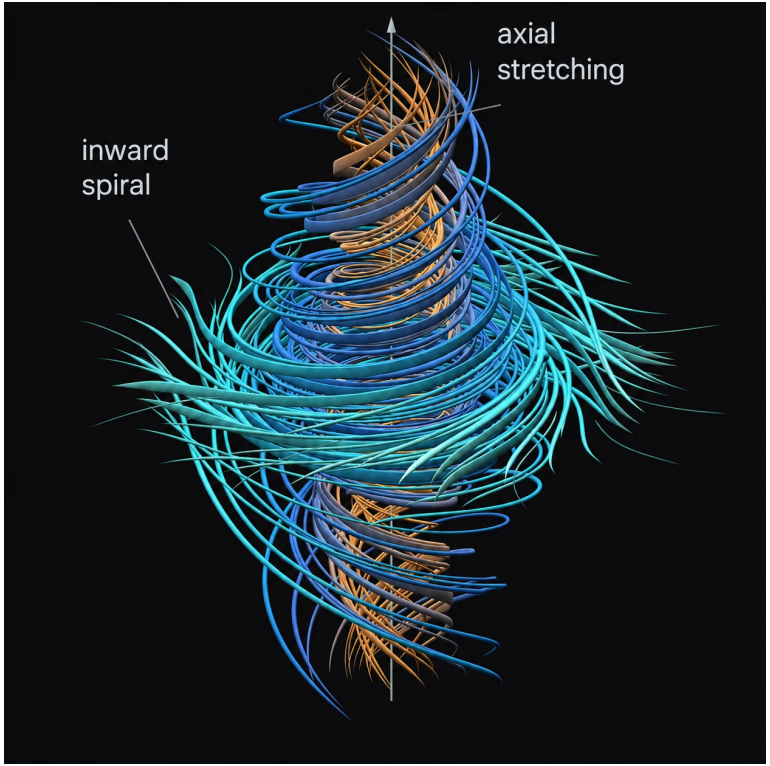
Brennon L. Shanks and Ondrej Tichacek

Institute of Organic Chemistry and Biochemistry Prague

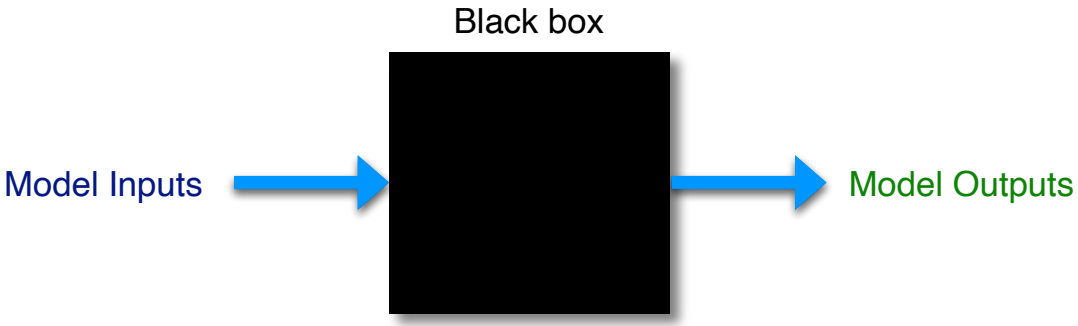
Email: shanks.brennon@uochb.cas.cz

Navier-Stokes millennium prize problem solved by OpenAI?

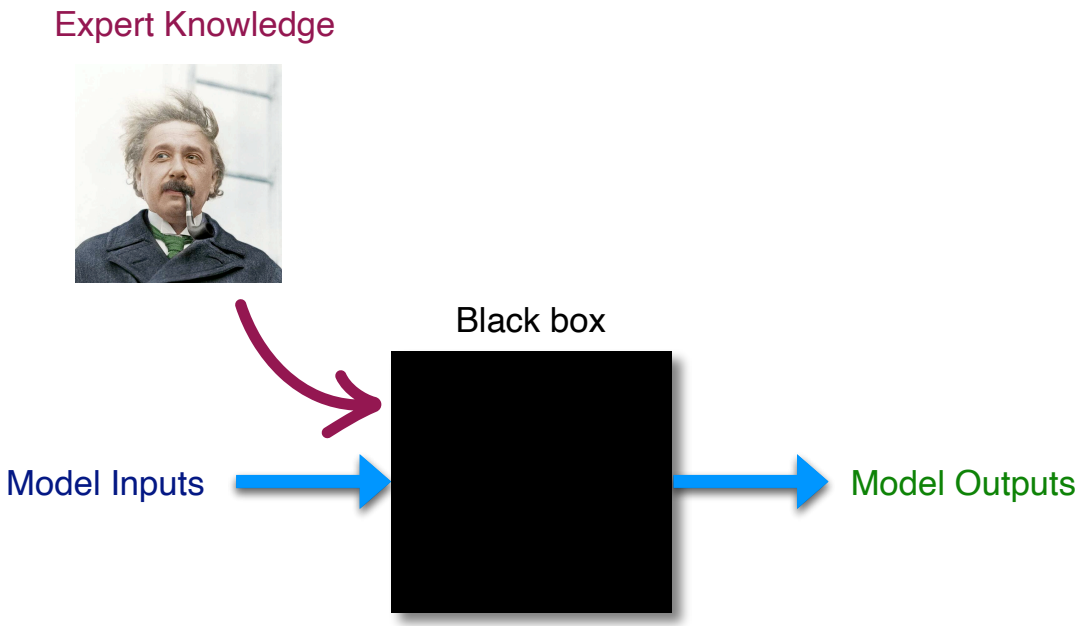
Our system produced an analytical proof and a Lean formalization that an initially smooth fluid at rest can develop a singularity in a finite time. The fluid has a smooth force applied to it, and its energy remains finite through the entire dynamics, from rest to the formation of the singularity. This resolves the Navier–Stokes Millennium Prize problem by establishing statement “C” (and also “D”) in the official Millennium Prize formulation(opens in a new window).



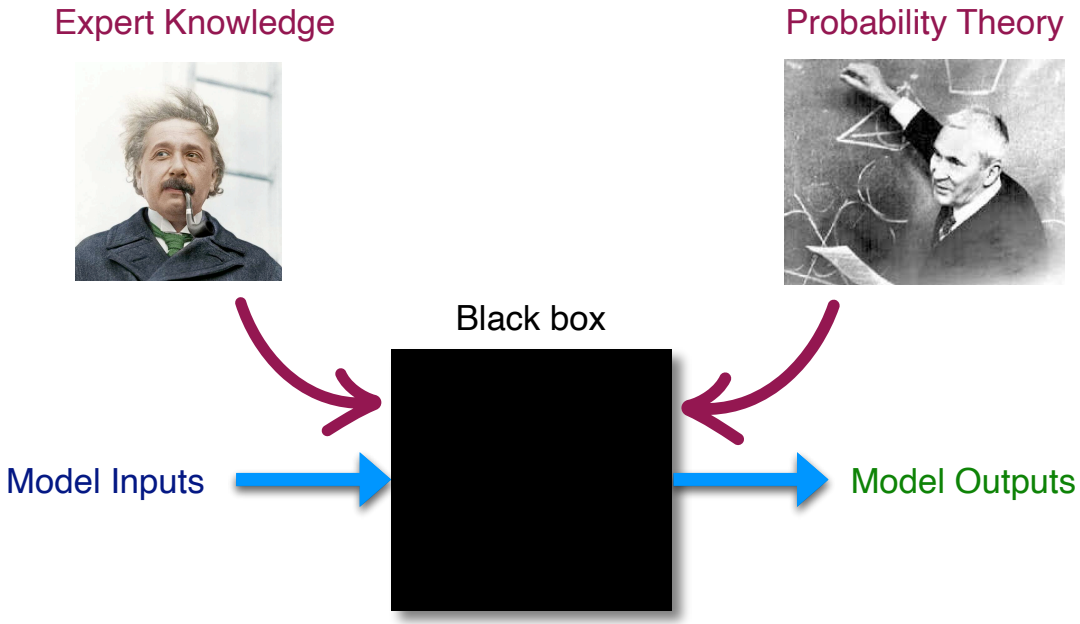
How can we make ML/AI more trustworthy?



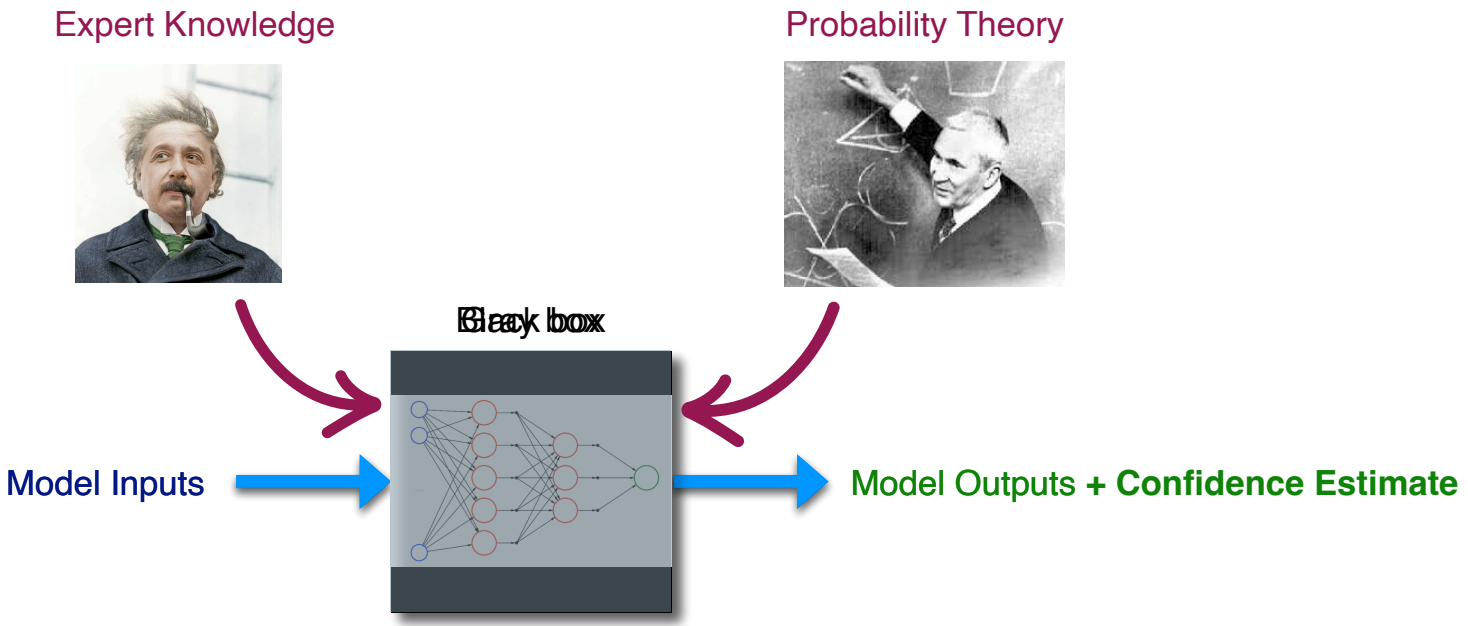
How can we make ML/AI more trustworthy?



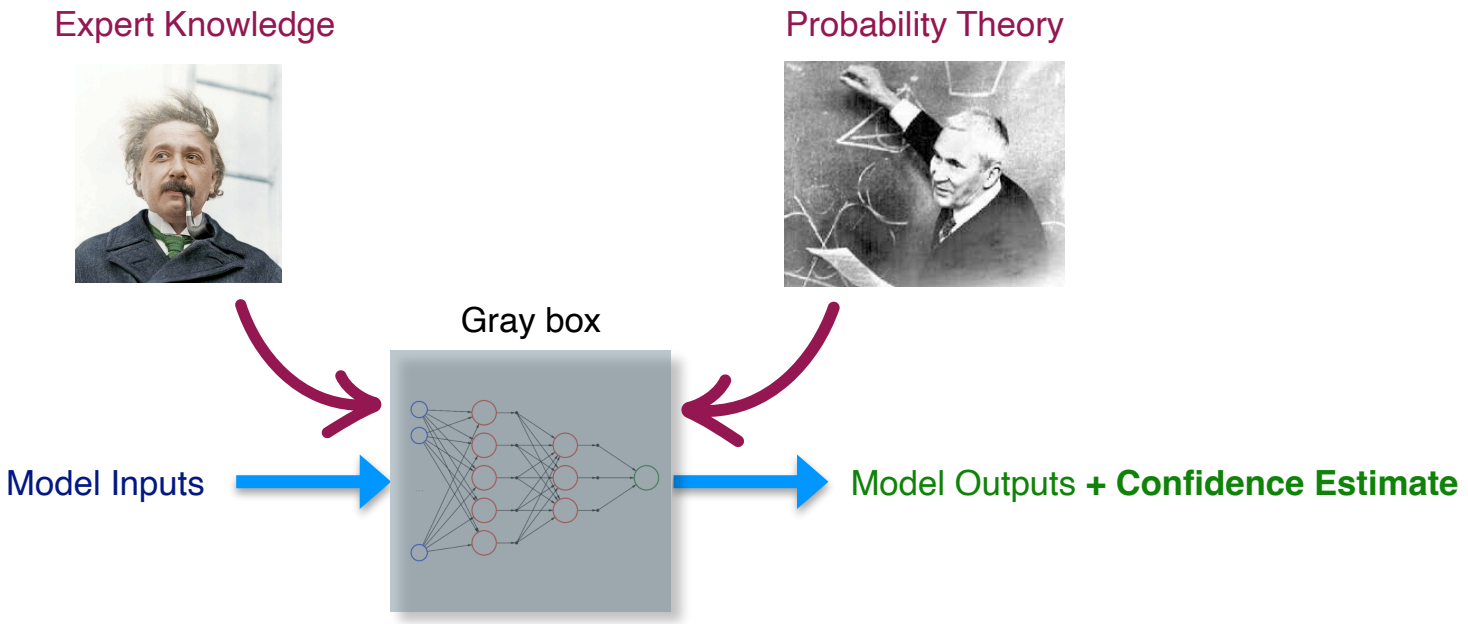
How can we make ML/AI more trustworthy?



How can we make ML/AI more trustworthy?



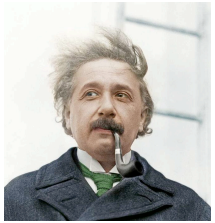
How can we make ML/AI more trustworthy?



How do we combine prior knowledge and probability theory?

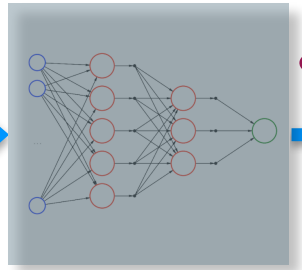
How can we make ML/AI more trustworthy?

Expert Knowledge



Probability Theory

Gray box



Model Inputs → Model Outputs **+ Confidence Estimate**

Bayes' Theorem!

How do we combine prior knowledge and probability theory?

$$p(\theta \mid Y) \propto p(Y \mid \theta) \times p(\theta)$$

Bayesian Intuition: Guessing the weight of a suitcase

prior

Bayesian Intuition: Guessing the weight of a suitcase

prior

Guess: 5-50 lbs

Bayesian Intuition: Guessing the weight of a suitcase

prior

Guess: 5-50 lbs

likelihood

You observe a child pick up the suitcase with one hand

Bayesian Intuition: Guessing the weight of a suitcase

prior

Guess: 5-50 lbs

likelihood

You observe a child pick up the suitcase with one hand

posterior

Update: 5-10 lbs

Bayesian Intuition: Guessing the weight of a suitcase

prior

Guess: 5-50 lbs

likelihood

You observe a child pick up the suitcase with one hand

posterior

Update: 5-10 lbs

Bayes theorem formalizes this process in a mathematically rigorous way.

posterior likelihood prior

$$p(\theta \mid Y) \propto p(Y \mid \theta) \times p(\theta)$$

Bayes' theorem as a general framework for learning

- Specify inverse problem:

  What do we want to learn (θ)?
  What do we observe (Y)?

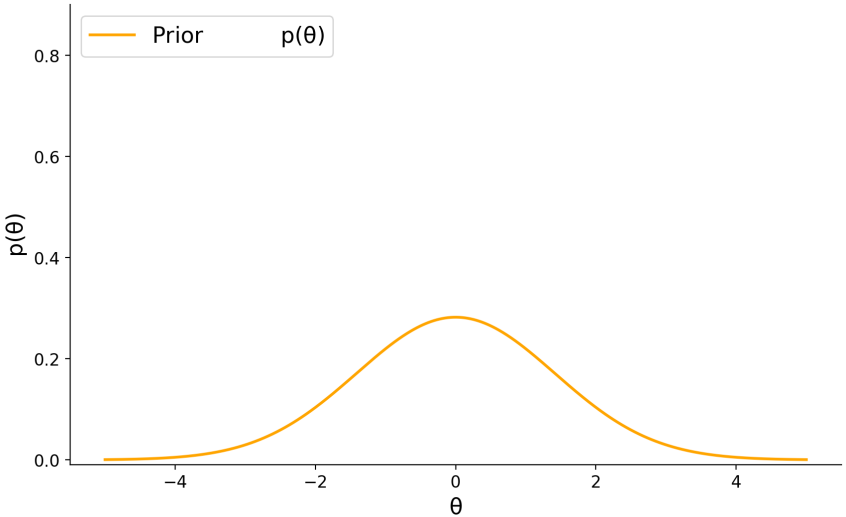
$$p(\theta \mid Y) \propto p(Y \mid \theta) \times p(\theta)$$

Bayes' theorem as a general framework for learning

- Specify inverse problem:

  What do we want to learn (θ)?
  What do we observe (Y)?

- Define **'prior'** probability distributions on θ.



$$p(\theta \mid Y) \propto p(Y \mid \theta) \times p(\theta)$$

prior

Bayes' theorem as a general framework for learning

- Specify inverse problem:

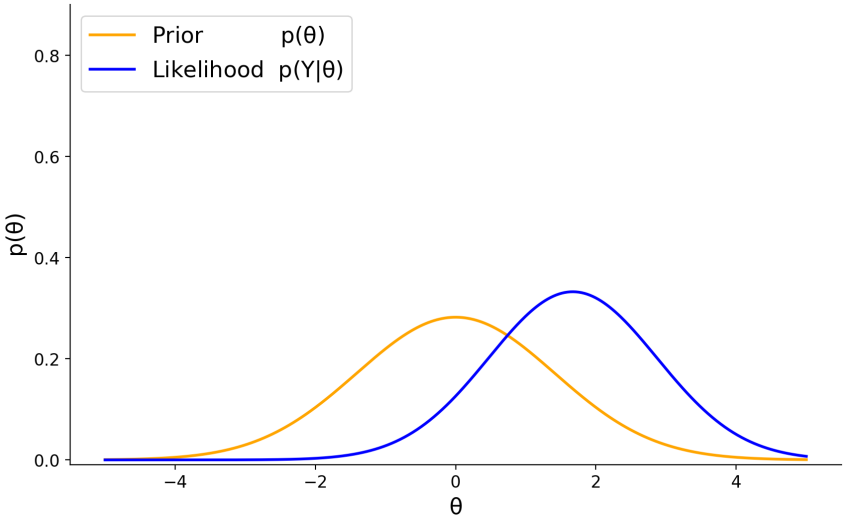
  What do we want to learn (θ)?
  What do we observe (Y)?

- Define **'prior'** probability distributions on θ.

- Define a **'likelihood'** function.

$$p(\theta \mid Y) \propto p(Y \mid \theta) \times p(\theta)$$

likelihood **prior**



Bayes' theorem as a general framework for learning

- Specify inverse problem:

  What do we want to learn (θ)?
  What do we observe (Y)?

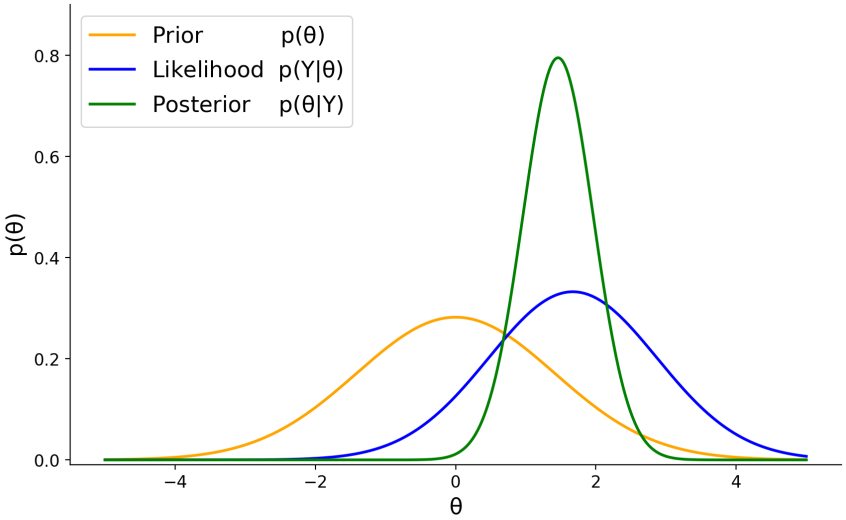
- Define **'prior'** probability distributions on θ.

- Define a **'likelihood'** function.

- Sample the **'posterior'** distribution (MCMC).

$$p(\theta \mid Y) \propto p(Y \mid \theta) \times p(\theta)$$

posterior **likelihood** **prior**



Activity 1: Predicting Win Rate of a Slot Machine

We are going to play a slot machine…

Rules:

1. A play costs 1 token.
2. If you win, you get 2.4 tokens. A loss eats your token.
3. There is some fixed win rate, *p*, that we don't know.

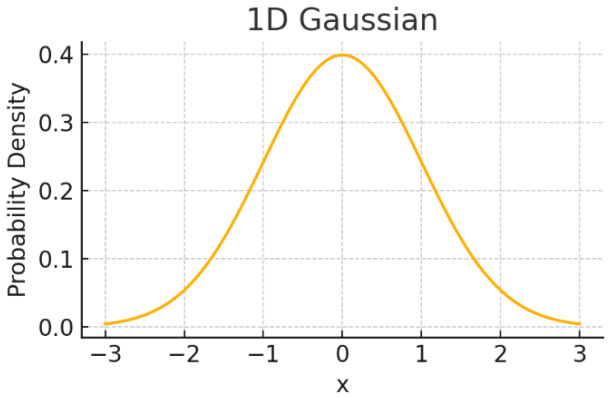
We have been watching this poor chap play, and we will gather data on how often he wins and how often he loses.

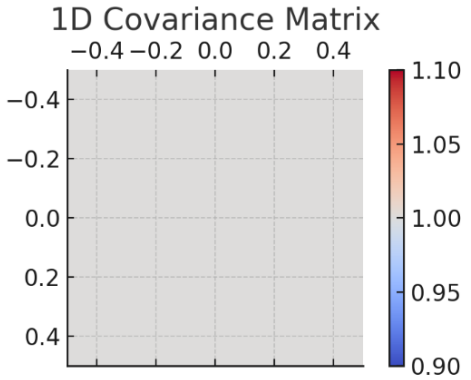
Using prior knowledge and the numbers of wins and losses, you will decide if you want to play this machine.

https://github.com/brennonshanks/czai_letni_skola_bayes

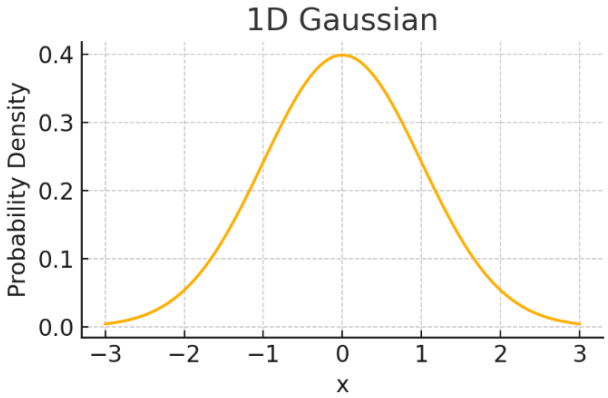
Modeling Functions with Gaussian Processes

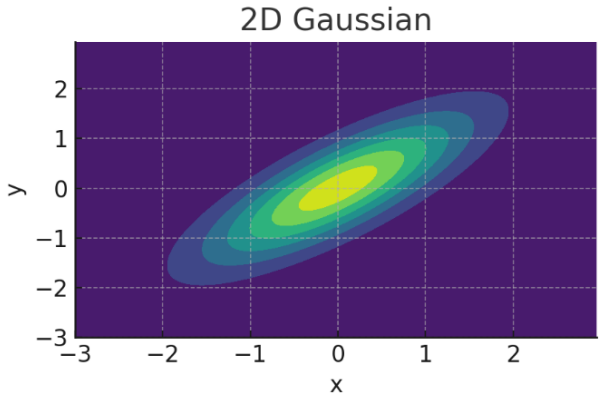
Visualizing Gaussians in Higher Dimensions

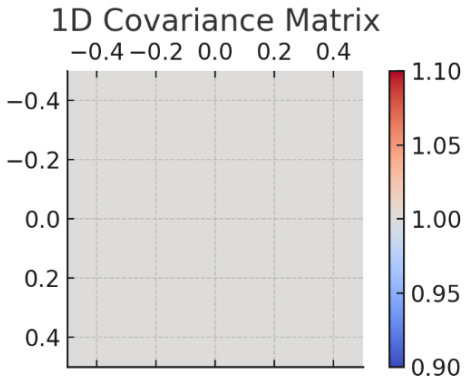


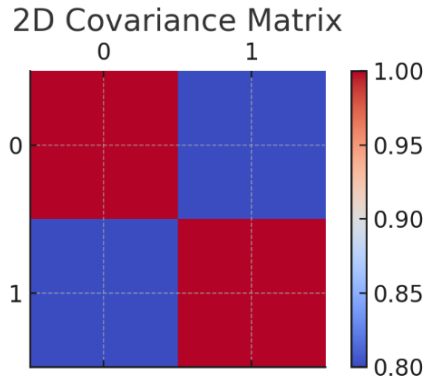


Visualizing Gaussians in Higher Dimensions

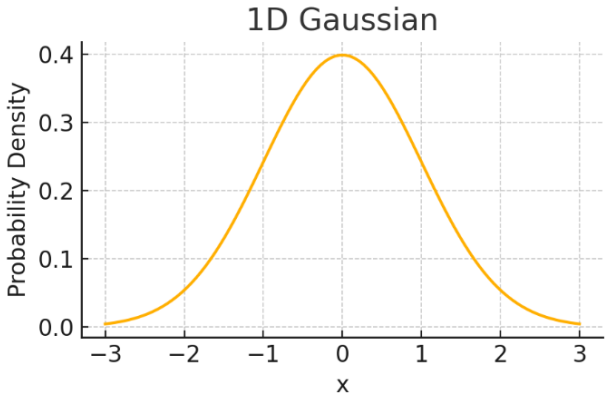


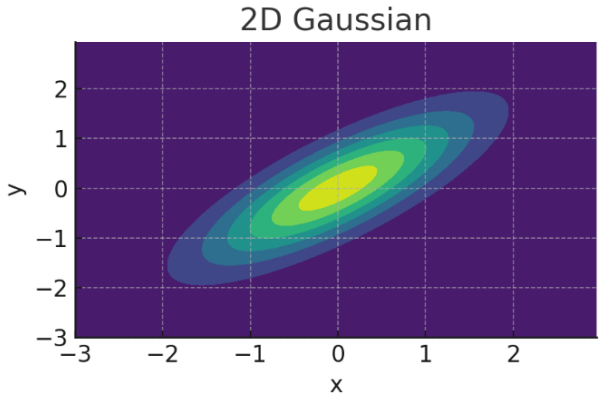


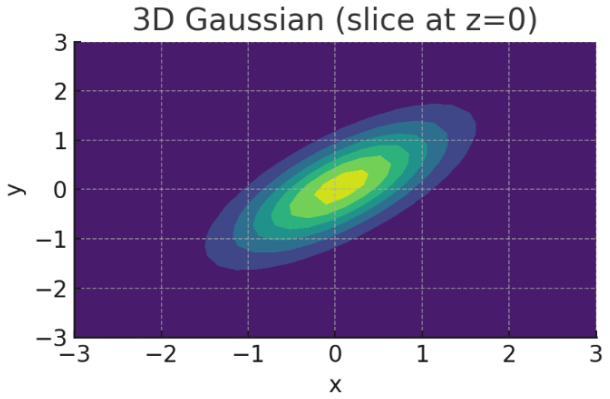


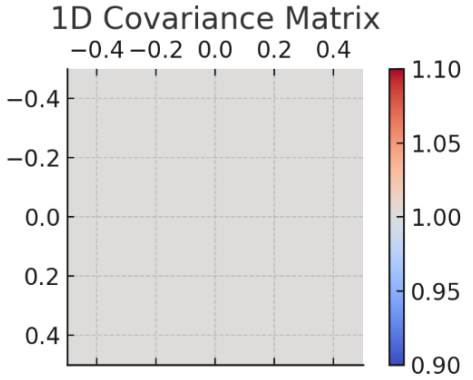


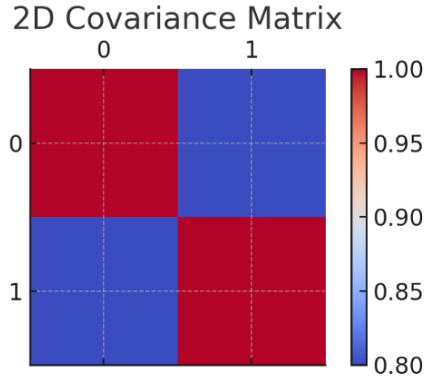
Visualizing Gaussians in Higher Dimensions

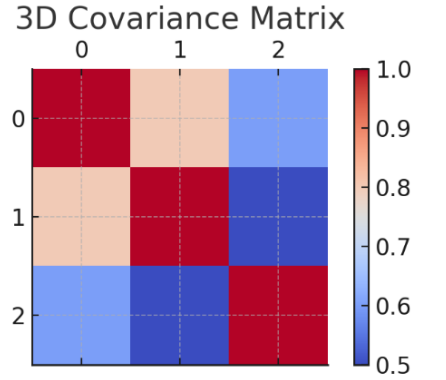








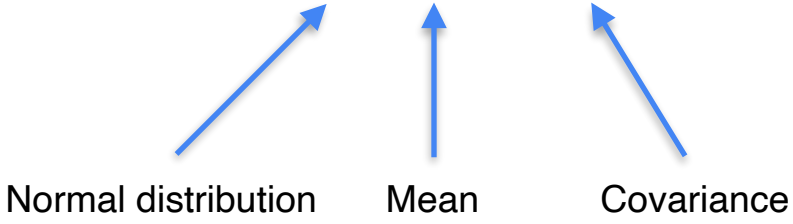




Gaussian processes extend this idea to infinite dimensions

- **Gaussian process regression (GPR)** models functions as infinite dimensional Gaussian distributions.
- Because these are **probability distributions**, we can use them with Bayes' theorem!

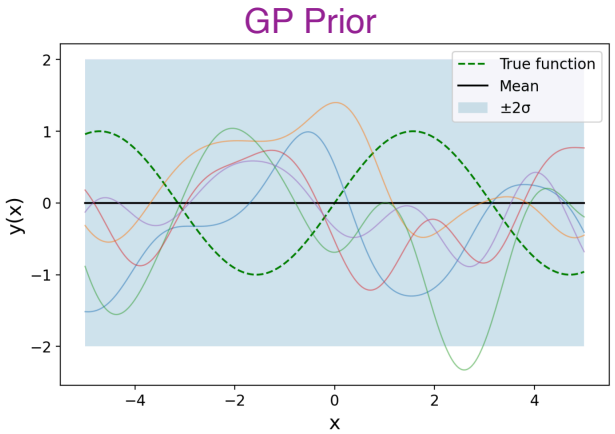
$$y \sim \mathcal{N}(\mu(x), k(x, x'))$$



Gaussian processes extend this idea to infinite dimensions

- **Gaussian process regression (GPR)** models functions as infinite dimensional Gaussian distributions.
- Because these are **probability distributions**, we can use them with Bayes' theorem!

$$y \sim \mathcal{N}(\mu(x), k(x, x'))$$

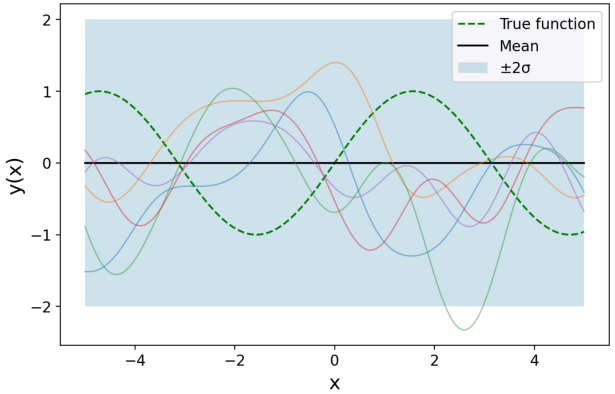


Gaussian processes extend this idea to infinite dimensions

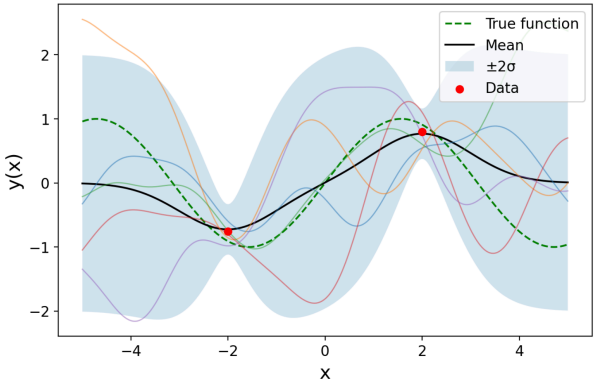
- **Gaussian process regression (GPR)** models functions as infinite dimensional Gaussian distributions.
- Because these are **probability distributions**, we can use them with Bayes' theorem!

$$y \sim \mathcal{N}(\mu(x), k(x, x'))$$

GP Prior



Posterior w/ 2 observations

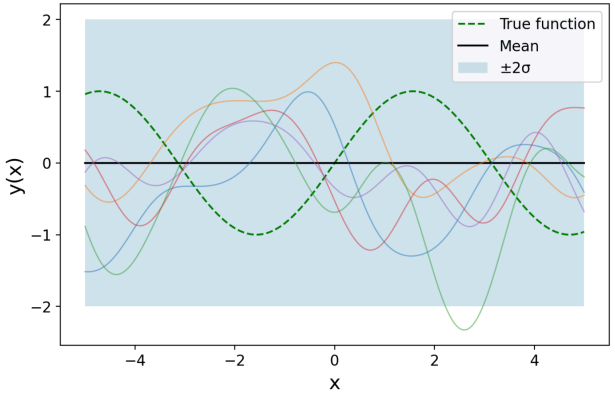


Gaussian processes extend this idea to infinite dimensions

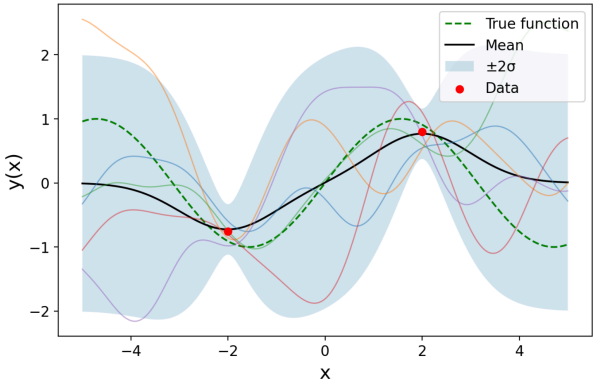
- **Gaussian process regression (GPR)** models functions as infinite dimensional Gaussian distributions.
- Because these are **probability distributions**, we can use them with Bayes' theorem!

$$y \sim \mathcal{N}(\mu(x), k(x, x'))$$

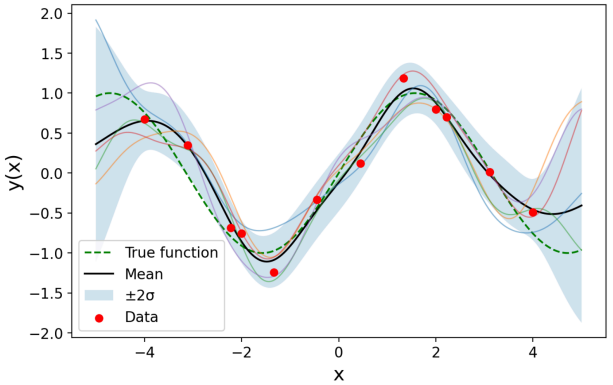
GP Prior



Posterior w/ 2 observations



Posterior w/ 12 observations



Gaussian processes can include physical information and are well-behaved

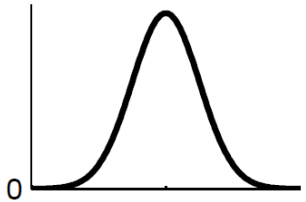
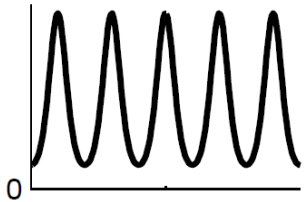
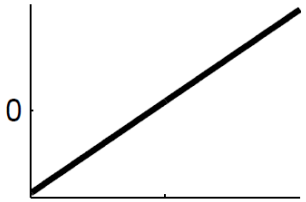
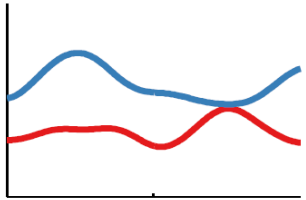
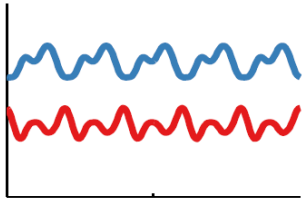
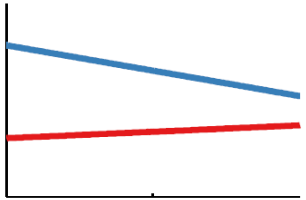
Property: Mean and covariance (k) can be designed to include physical properties or “expert knowledge”

$$y \sim \mathcal{N}(\mu(x), k(x, x'))$$

- Knowledge/physics-based simulation prediction
- Can be predicted by another ML model (GP, NN)

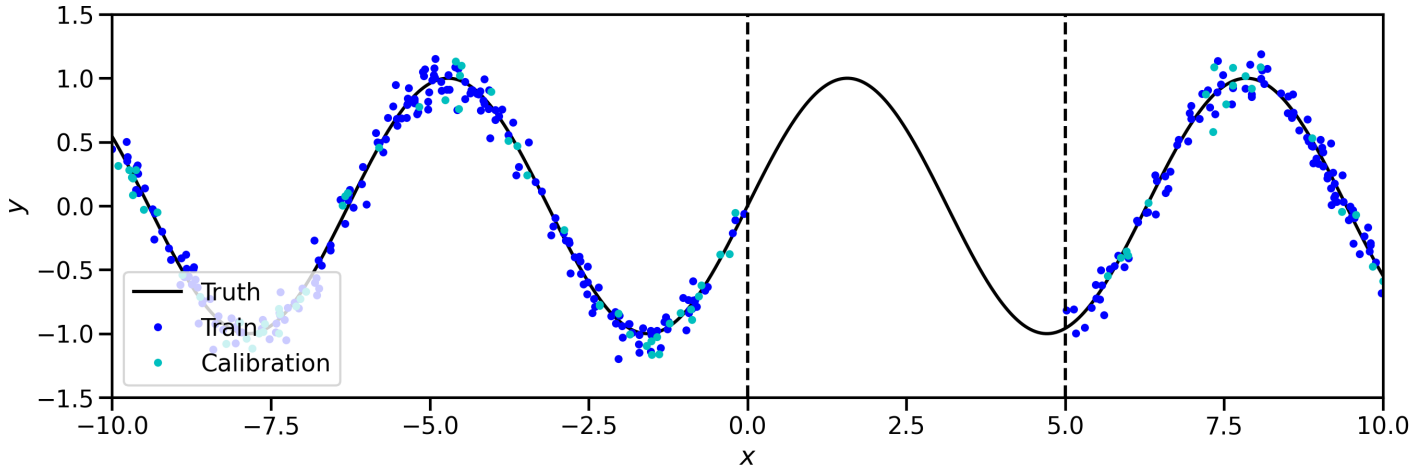
- known limiting behaviors
- known relationships between inputs

Kernel choice is critical to specifying a prior for a function

Kernel name:	Squared-exp (SE)	Periodic (Per)	Linear (Lin)
$k(x, x') =$	$\sigma_f^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$	$\sigma_f^2 \exp\left(-\frac{2}{\ell^2}\sin^2\left(\pi\frac{x-x'}{p}\right)\right)$	$\sigma_f^2(x-c)(x'-c)$
Plot of $k(x, x')$:			
	$x - x'$	$x - x'$	x (with $x' = 1$)
Functions $f(x)$ sampled from GP prior:			
	x	x	x
Type of structure:	local variation	repeating structure	linear functions

Activity 2: Gaussian process demo

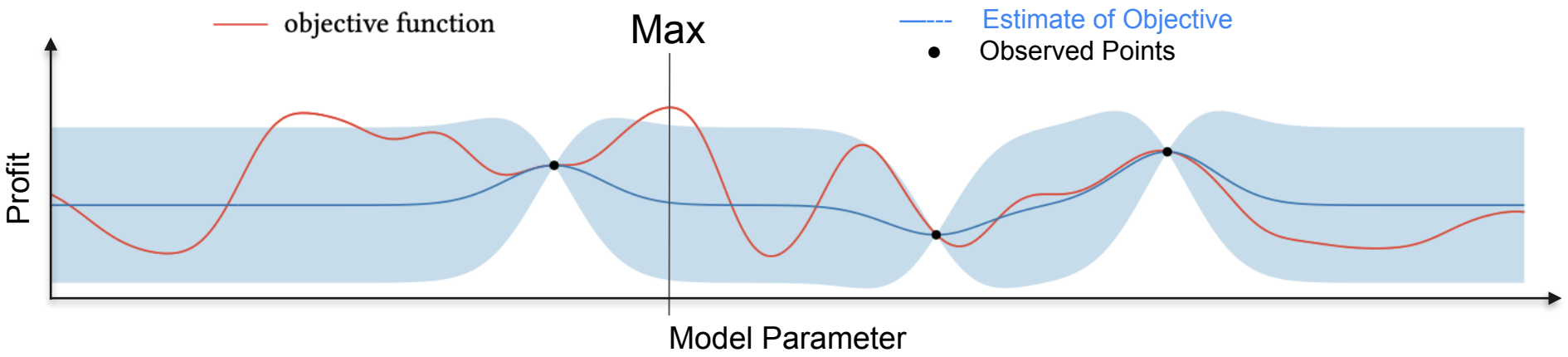
https://kermodegroup.github.io/demos/regression-demo.html



Efficient optimization requires knowledge of uncertainty and risk

Suppose we want to maximize an objective, but obtaining training data is expensive.

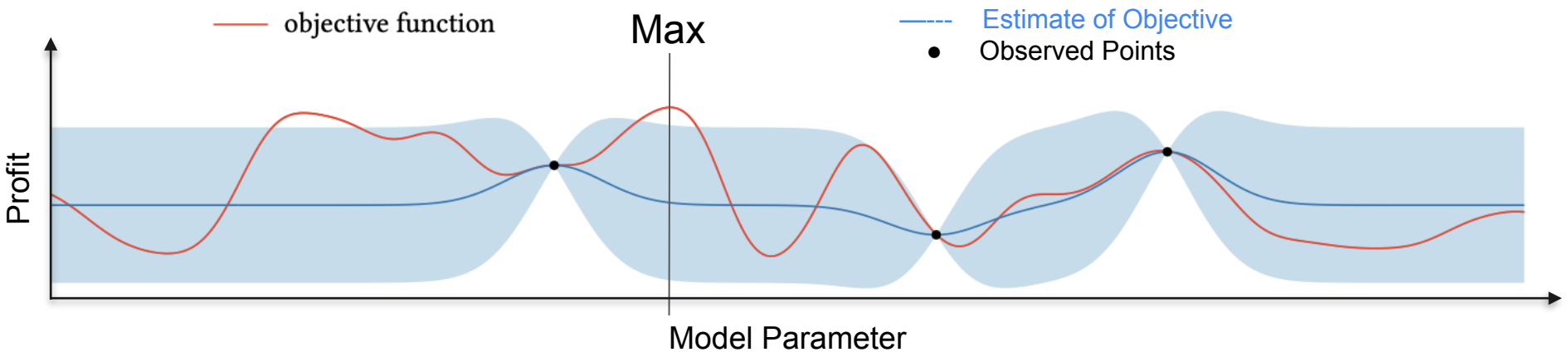
To minimize training time, we want to find the optimal next training point.



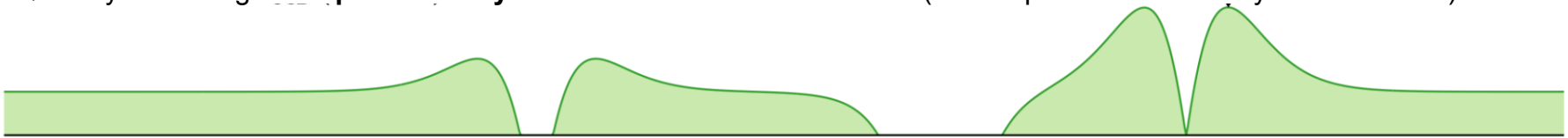
Efficient optimization requires knowledge of uncertainty and risk

Suppose we want to maximize an objective, but obtaining training data is expensive.

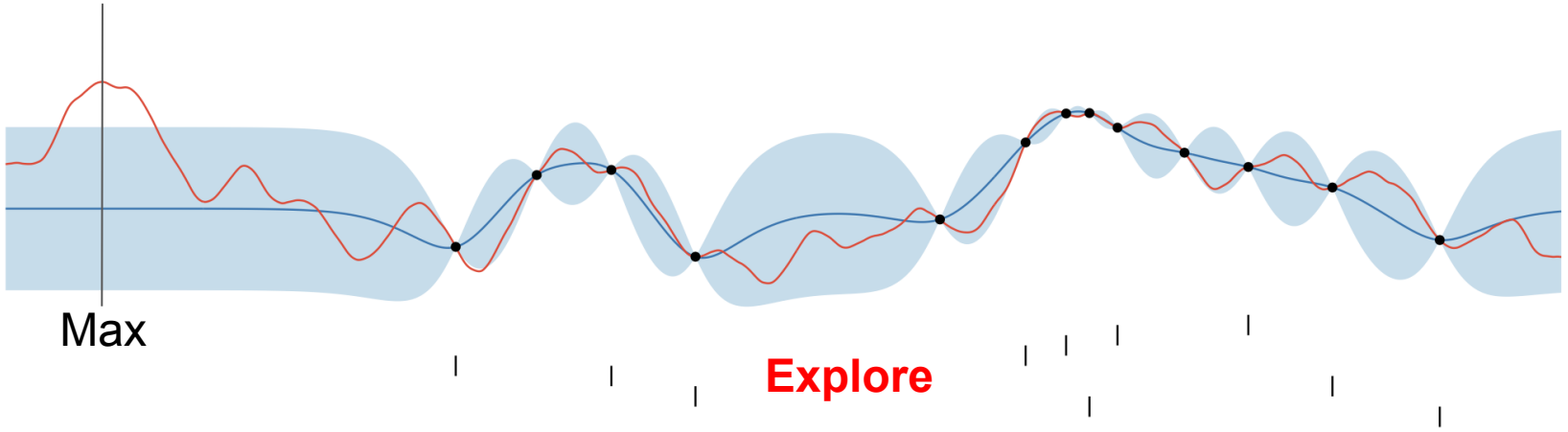
To minimize training time, we want to find the optimal next training point.



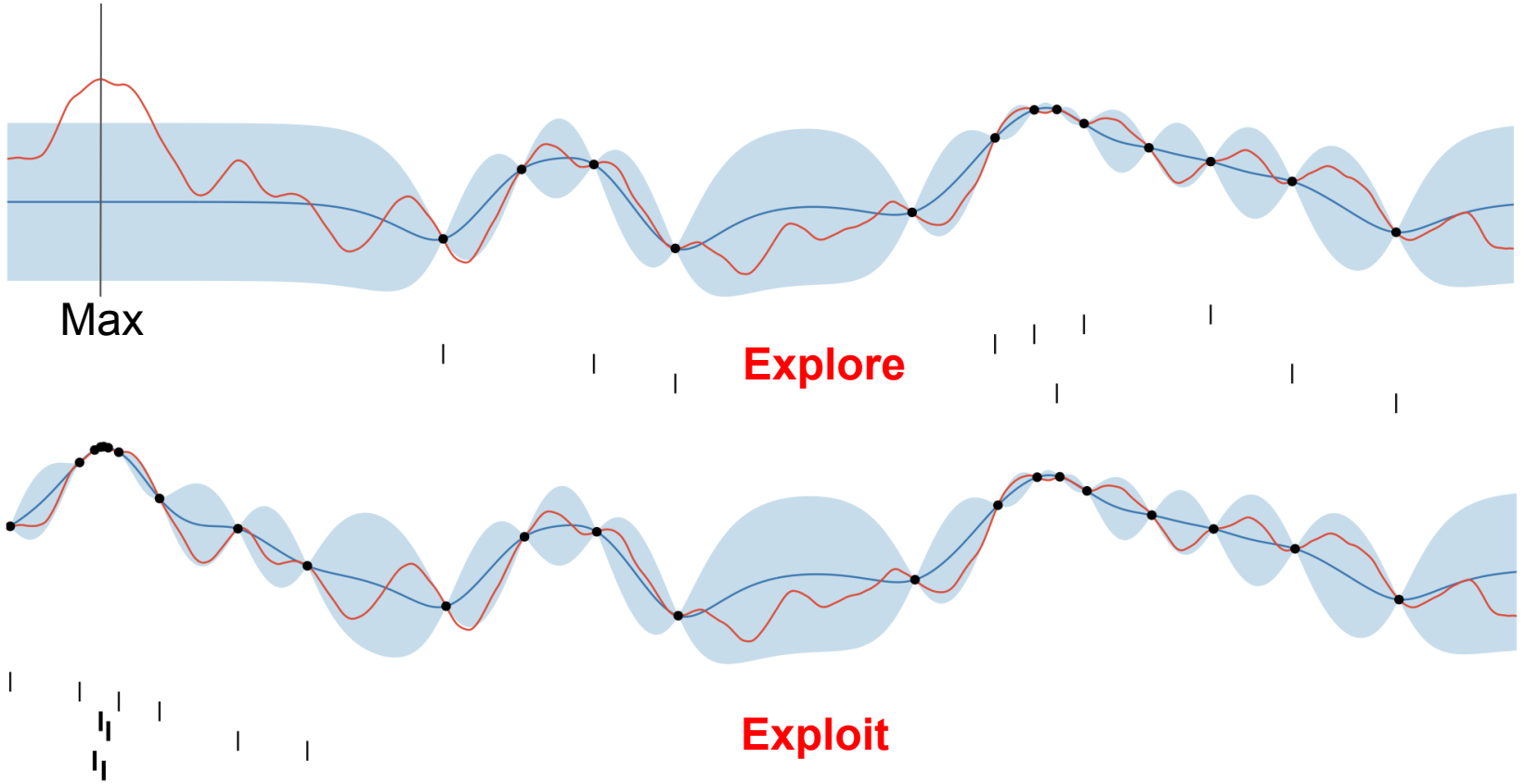
Quantify the change in **expected utility** and choose the maximum value (can emphasize uncertainty or risk or both)



Efficient optimization requires knowledge of uncertainty and risk

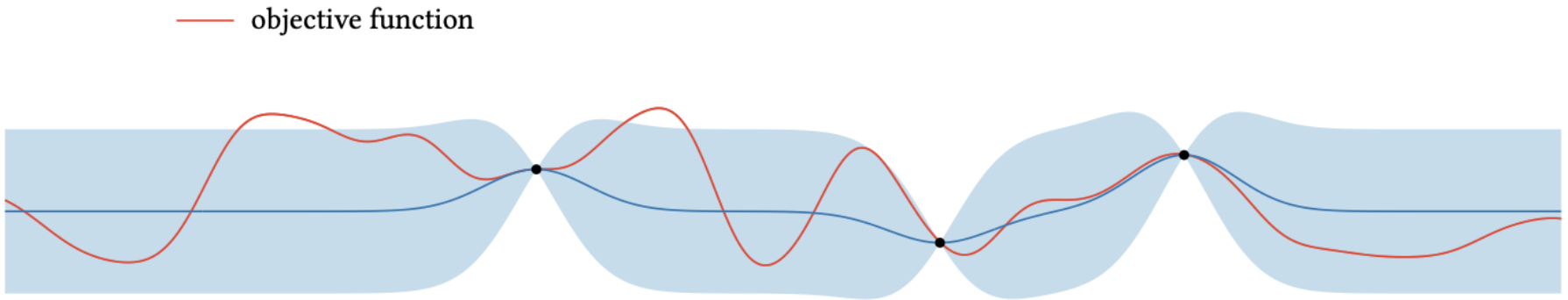


Efficient optimization requires knowledge of uncertainty and risk



Activity 3: Bayesian optimization

https://github.com/brennonshanks/czai_letni_skola_bayes



Strengths, Limitations, and Outlook

Suggested packages: Python - PyMC, Pyro; Julia - Turing

Bayesian methods are extremely powerful, but also have limitations…

Pros…

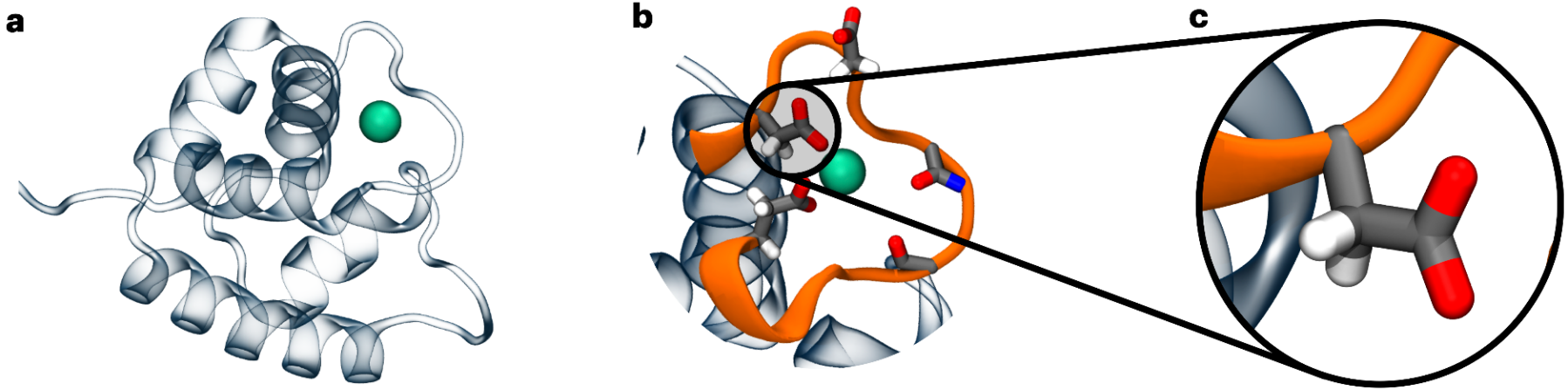
1. UQ with low data
2. Compatible with physics models
3. Mathematically “well-behaved”
4. Gold-standard UQ capability
5. Best for “risky” problems where UQ matters

Cons…

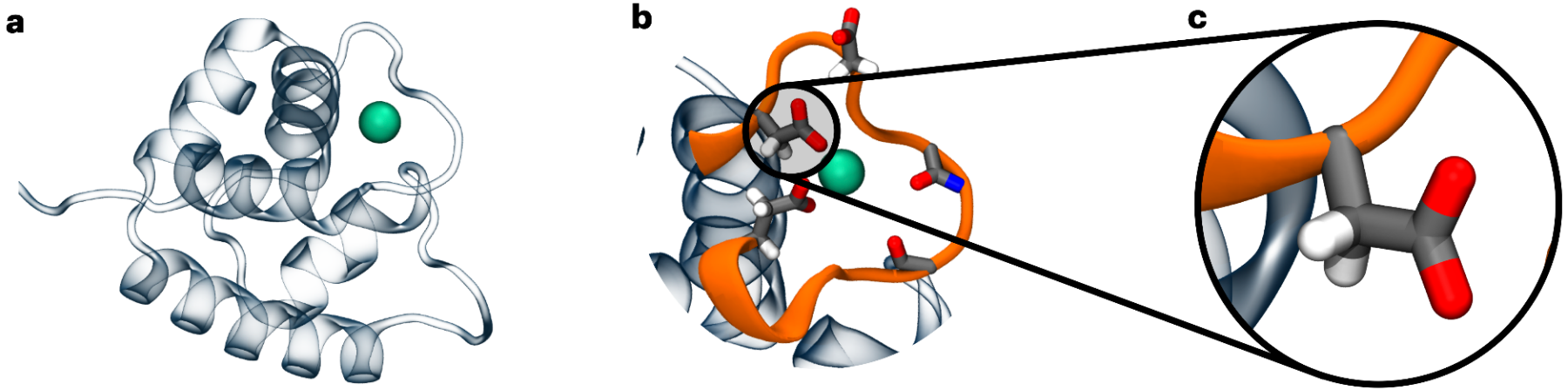
1. Computationally expensive (MCMC sampling of posteriors)
2. Prior selection can bias results - sensitivity analysis and thinking are important
3. GPs have no fixed size (matrix grows with training data)

Applications in our research

Binding of Ca 2+ to troponin C regulates heart rate

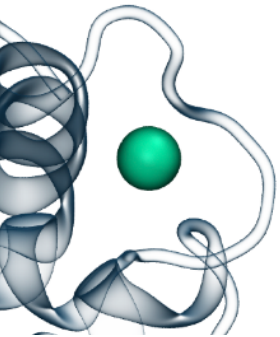


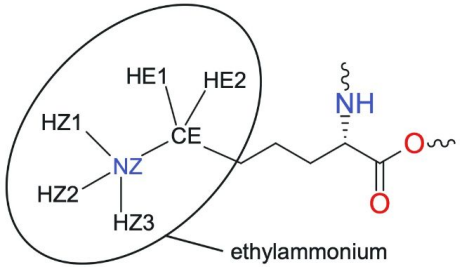
Binding of Ca 2+ to troponin C regulates heart rate



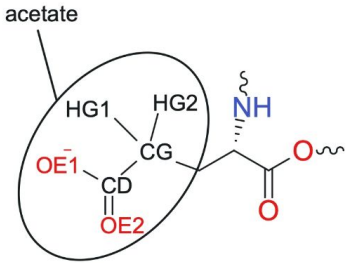
Standard MD methods predict a heartbeat approximately every 3,000,000 years…

Side chains in the troponin EF Hand loop binding domain

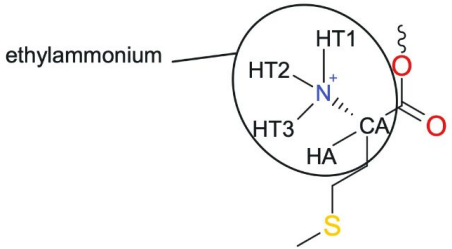




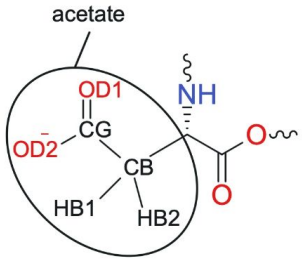
Lys



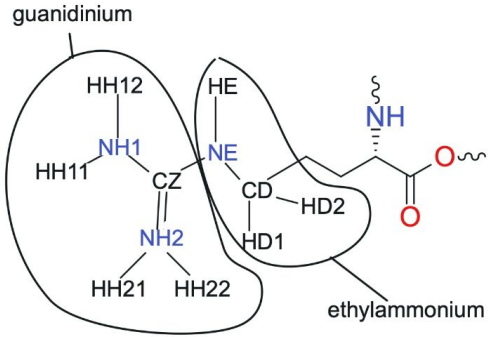
Glu



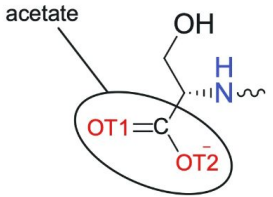
Met (N-terminus)



Asp



Arg



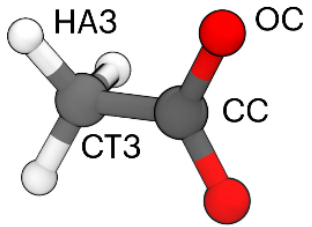
Ser (C-terminus)

Charged residue of cTnC decomposed into the fragments for parameterization. Atoms to-be-parameterized are labeled by the CHARMM36 atom names.

Using Bayes to learn model parameters for molecules

1. Simulate fragment in water with AIMD (DFT-MD)

Example: acetate ion

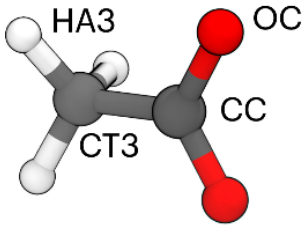


$\boldsymbol{Y}$ = Structural Data from DFT
- radial distribution functions
- hydrogen bond counts
- restrained distance distributions

Using Bayes to learn model parameters for molecules

1. Simulate fragment in water with AIMD (DFT-MD)

Example: acetate ion



$\boldsymbol{Y}$ = Structural Data from DFT
- radial distribution functions
- hydrogen bond counts
- restrained distance distributions

2. Train an ML surrogate on the classical MD model

Model: ECC MD

$$H(\{\mathbf{r}_i\},\{\mathbf{p}_i\}) = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + U(\{\mathbf{r}_i\})$$

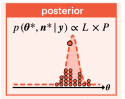
$$U_{ij}^{\mathrm{ECC}}(r_{ij}) = \frac{1}{4\pi\varepsilon_0}\frac{q_i^{\mathrm{ECC}} q_j^{\mathrm{ECC}}}{r_{ij}} = \frac{1}{4\pi\varepsilon_0}\frac{q_i q_j}{\varepsilon_{\mathrm{el}}\, r_{ij}}$$

Machine Learning Model (GP)

Charges  Pred. Y

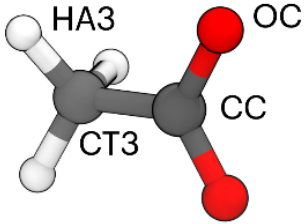
$\boldsymbol{\theta}$ = Partial charge of each atom in the fragment

Using Bayes to learn model parameters for molecules



1. Simulate fragment in water with AIMD (DFT-MD)

Example: acetate ion



$\boldsymbol{Y}$ = Structural Data from DFT
- radial distribution functions
- hydrogen bond counts
- restrained distance distributions

2. Train an ML surrogate on the classical MD model

Model: ECC MD

$$H(\{\mathbf{r}_i\},\{\mathbf{p}_i\}) = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + U(\{\mathbf{r}_i\})$$

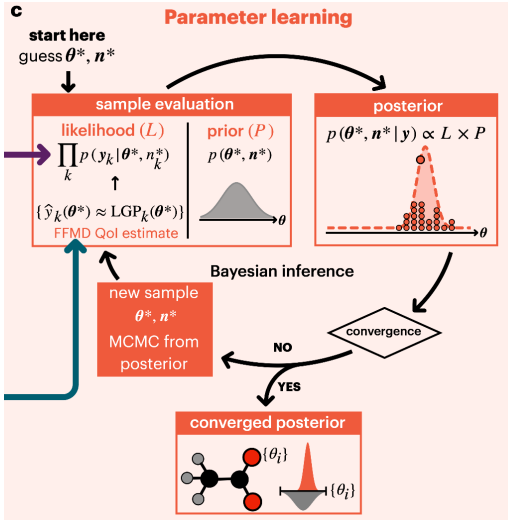
$$U_{ij}^{\mathrm{ECC}}(r_{ij}) = \frac{1}{4\pi\varepsilon_0}\frac{q_i^{\mathrm{ECC}} q_j^{\mathrm{ECC}}}{r_{ij}} = \frac{1}{4\pi\varepsilon_0}\frac{q_i q_j}{\varepsilon_{\mathrm{el}} r_{ij}}$$

Machine Learning Model (GP)

Charges  Pred. Y

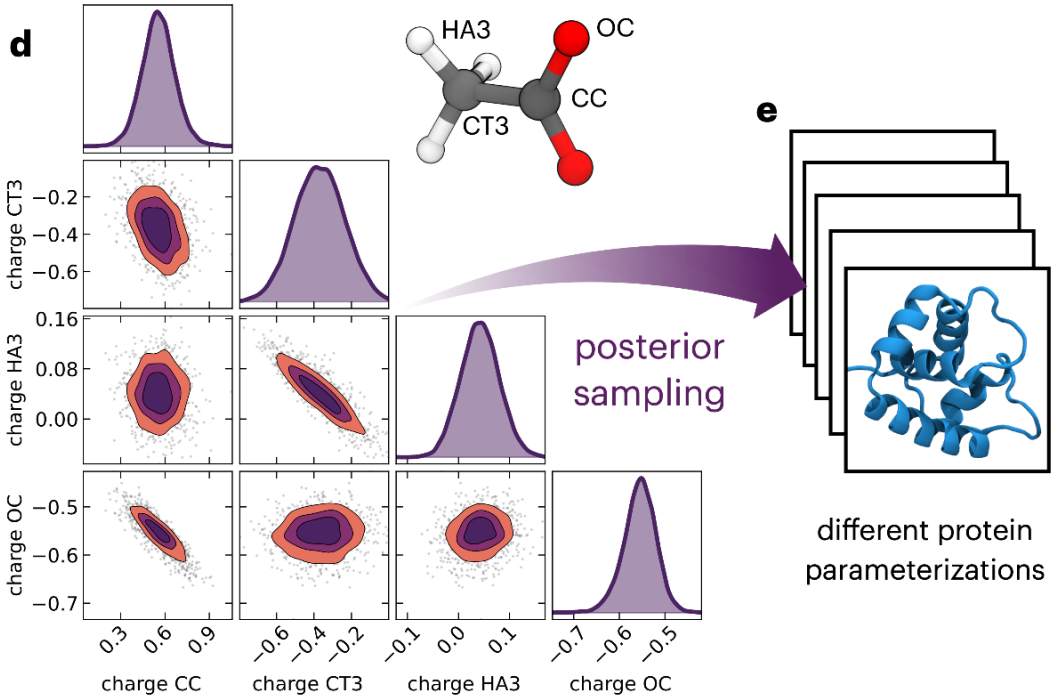
$\boldsymbol{\theta}$ = Partial charge of each atom in the fragment

3. Learn the Posterior



$p(\boldsymbol{\theta} \mid \boldsymbol{Y})$

Posterior binding free energy reproduces experiment



Posterior binding free energy reproduces experiment