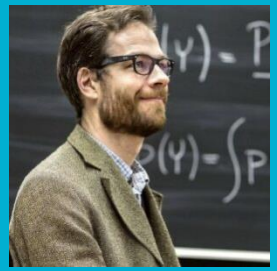
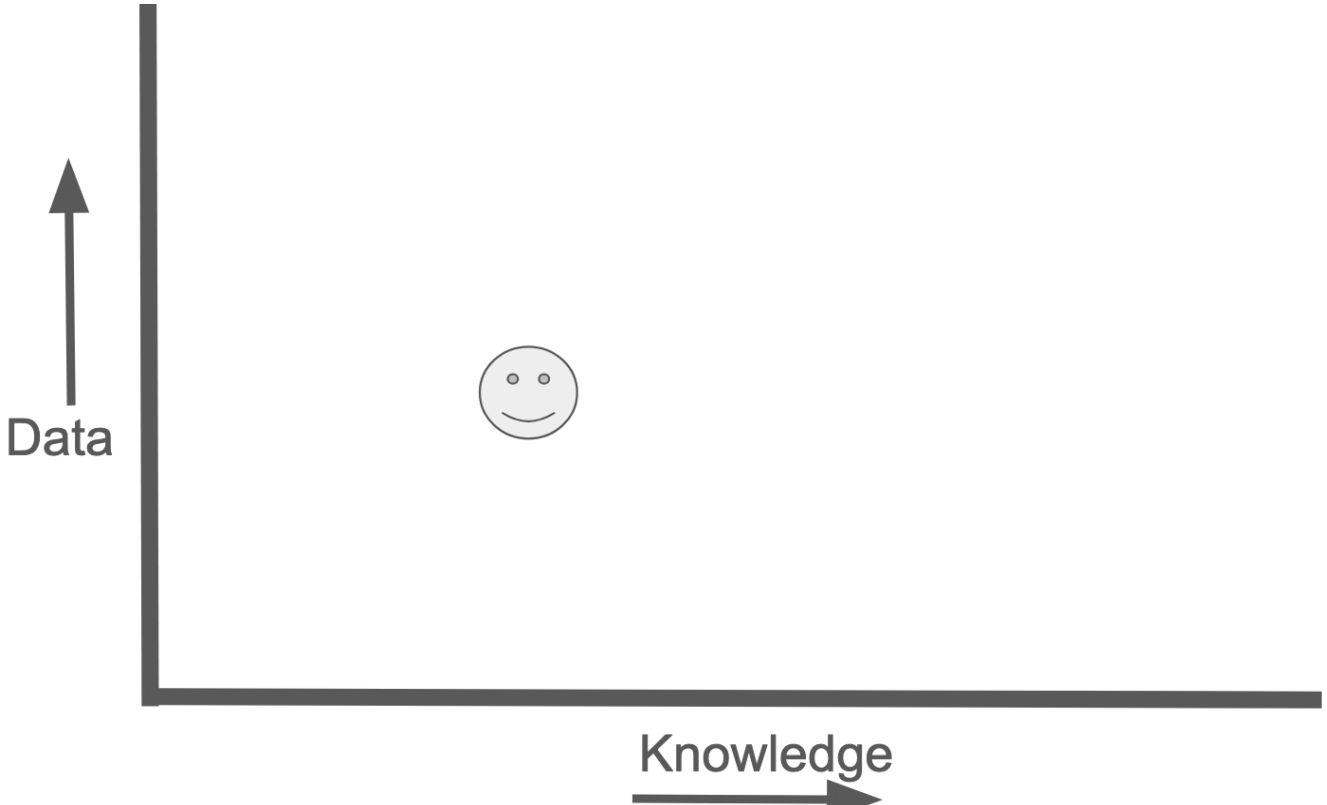
# Bayesian optimisation and beyond ....

Henry Moss
Lancaster University

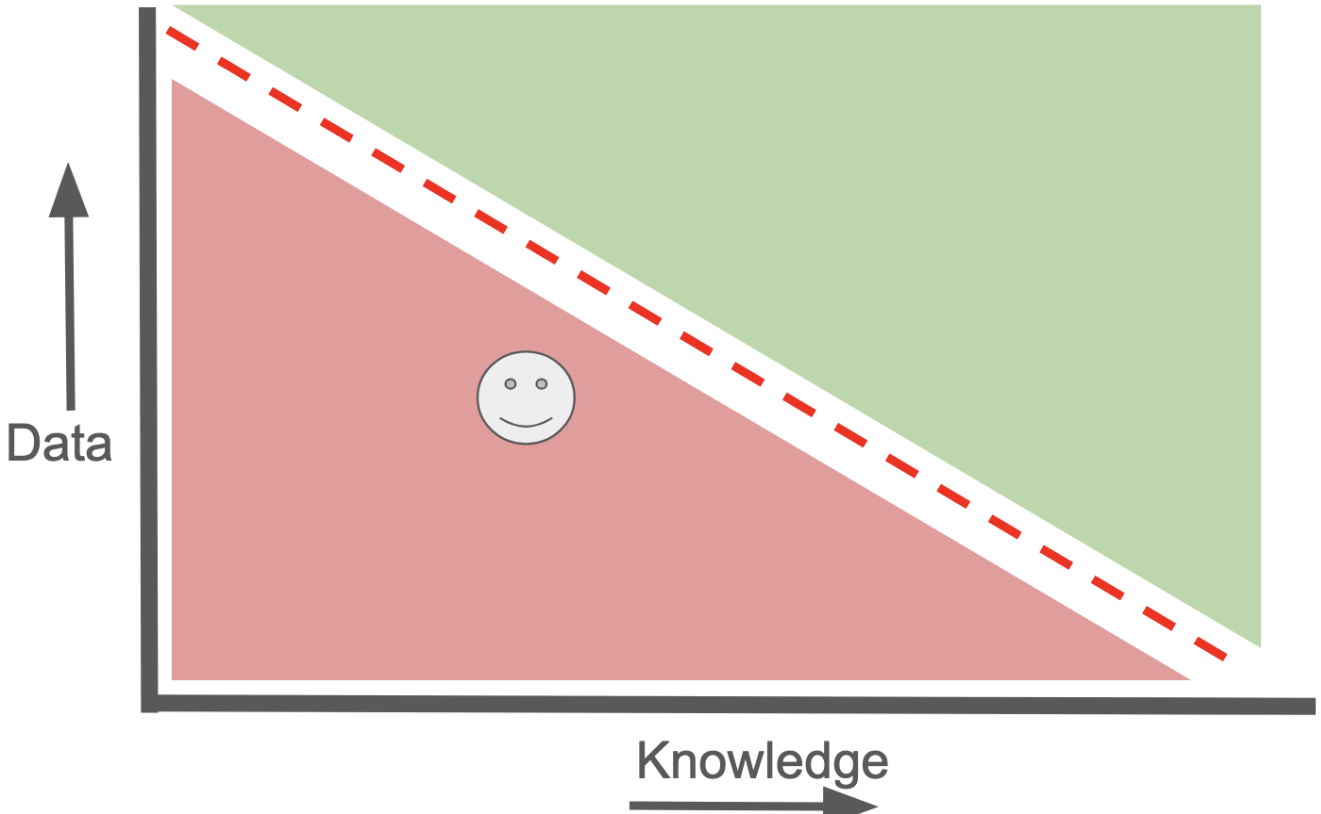
# Why bother with all this probabilistic ML stuff?



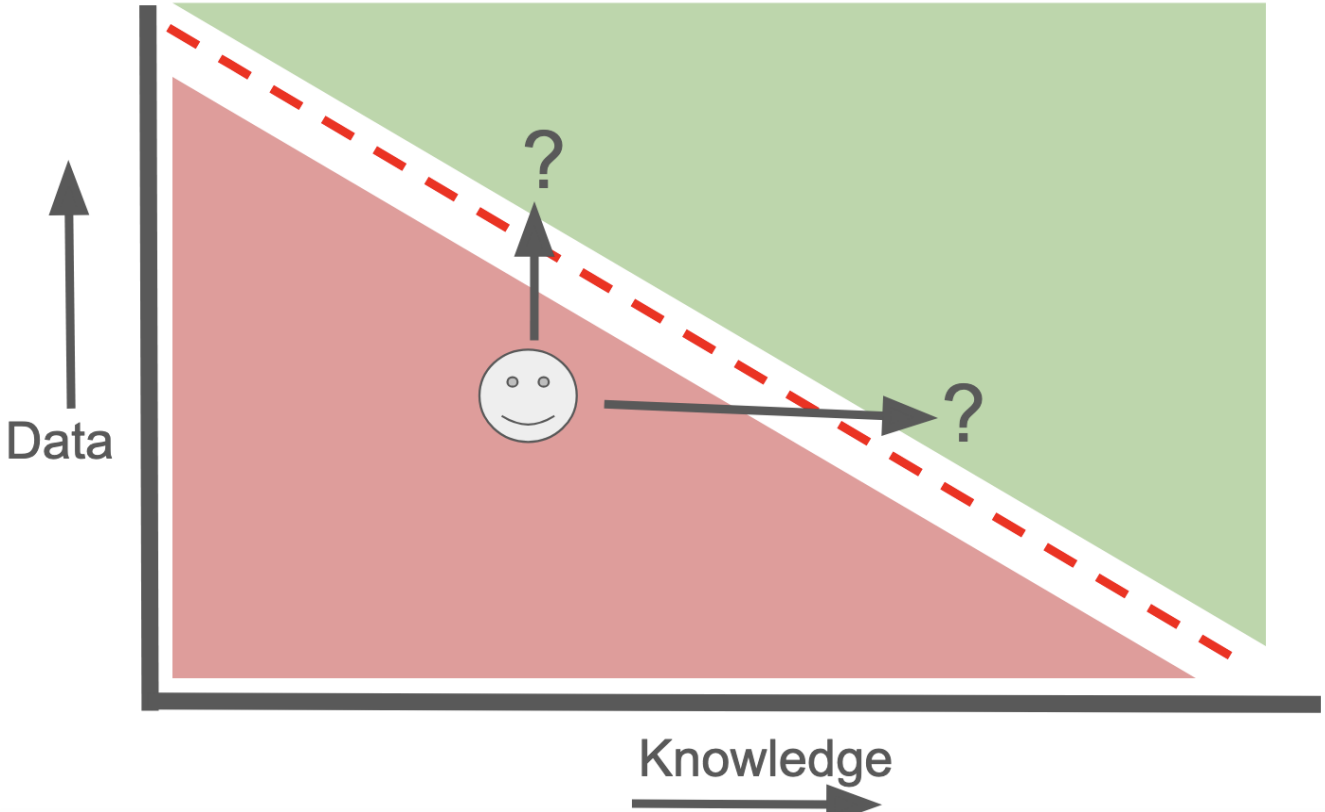
# To **ML** or not to **ML?**



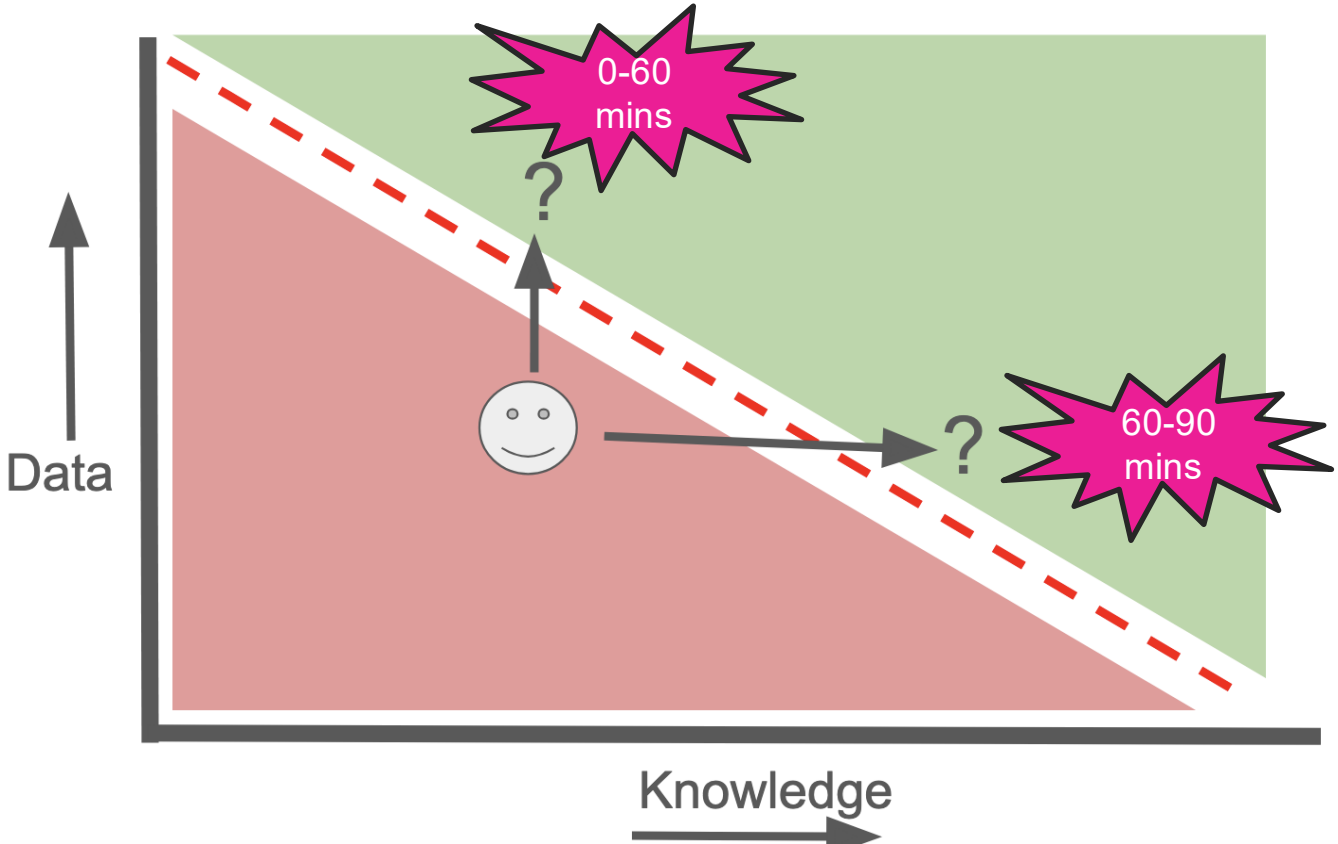
# To ML or not to ML?



# To ML or not to ML?



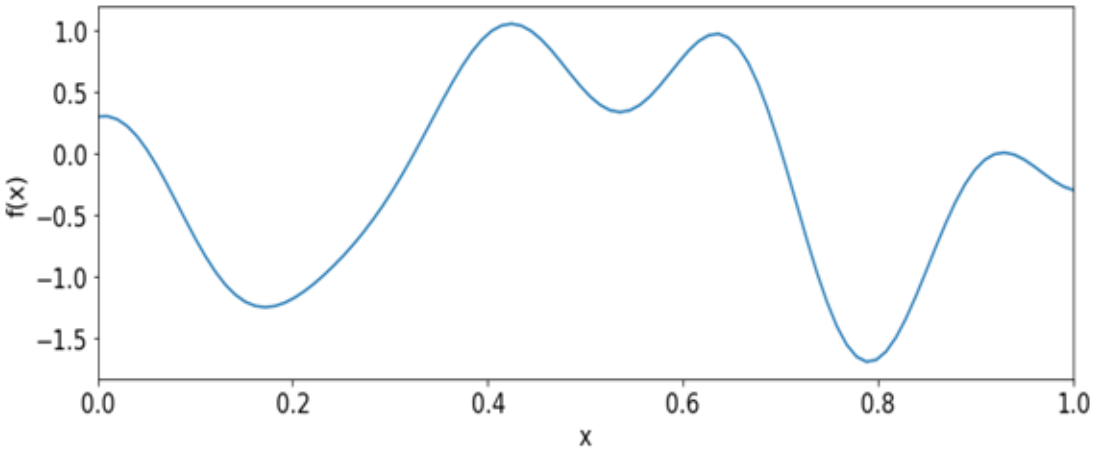
# To ML or not to ML?



# How can we use knowledge of uncertainty?

# How can we use knowledge of uncertainty?

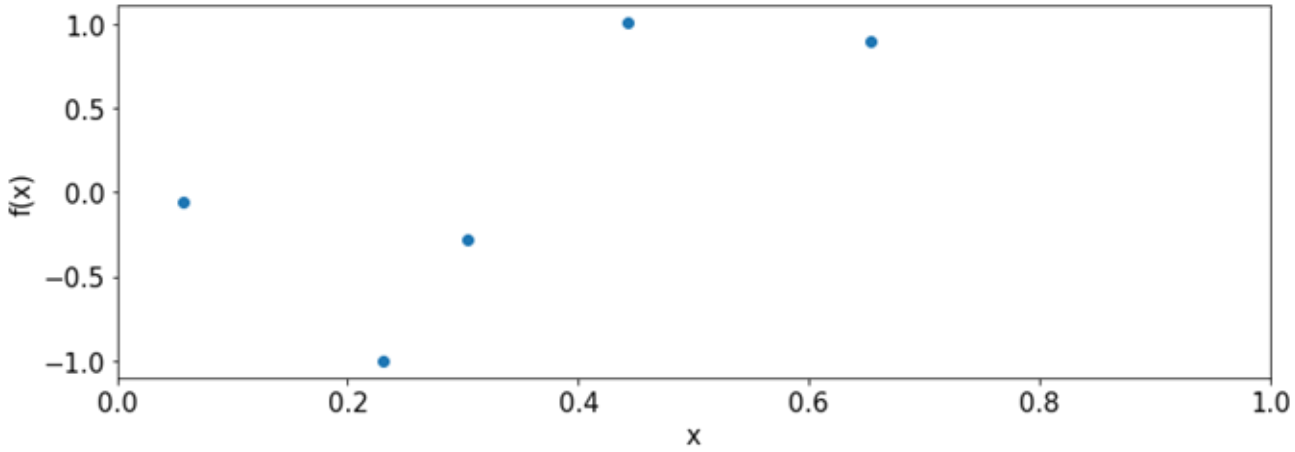
Let's find the maximum of a 1D function:



# How can we use knowledge of uncertainty?

Let's find the maximum of a 1D function:

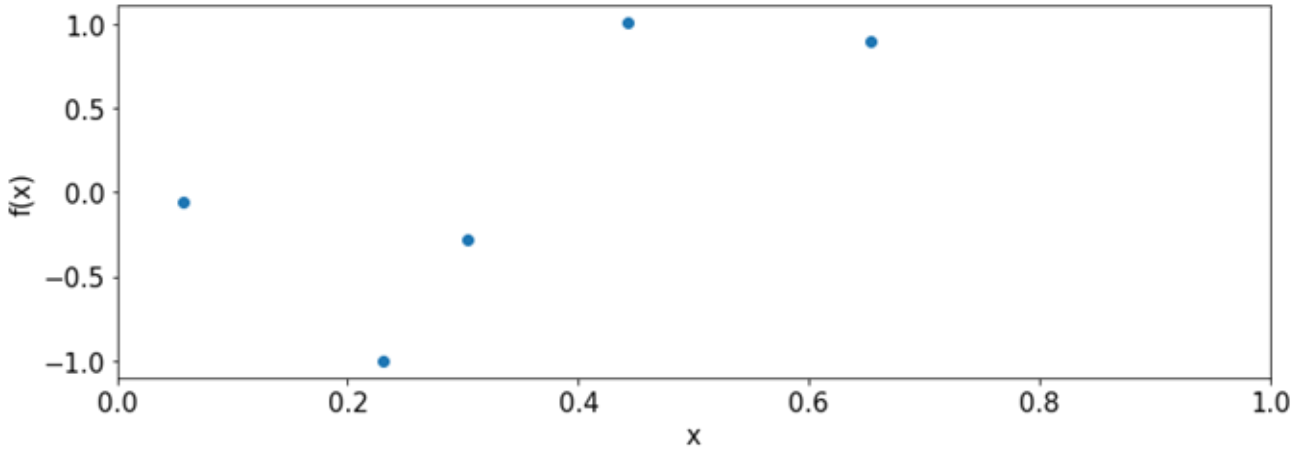
Using as **few** function evaluations as possible!



# How can we use knowledge of uncertainty?

Let's find the maximum of a 1D function:

Using as **few** function evaluations as possible!



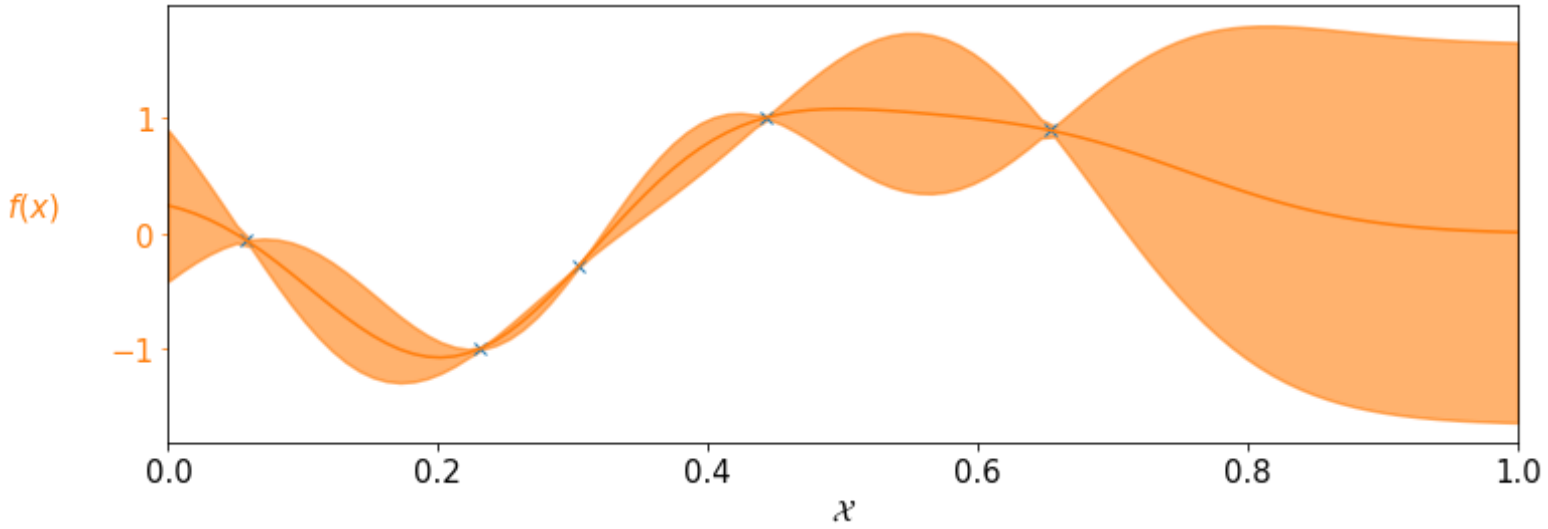

Where should we next evaluate? Explore/Exploit?

# Bayesian optimisation

# Uncertainty allows us to define the expected utility of new data

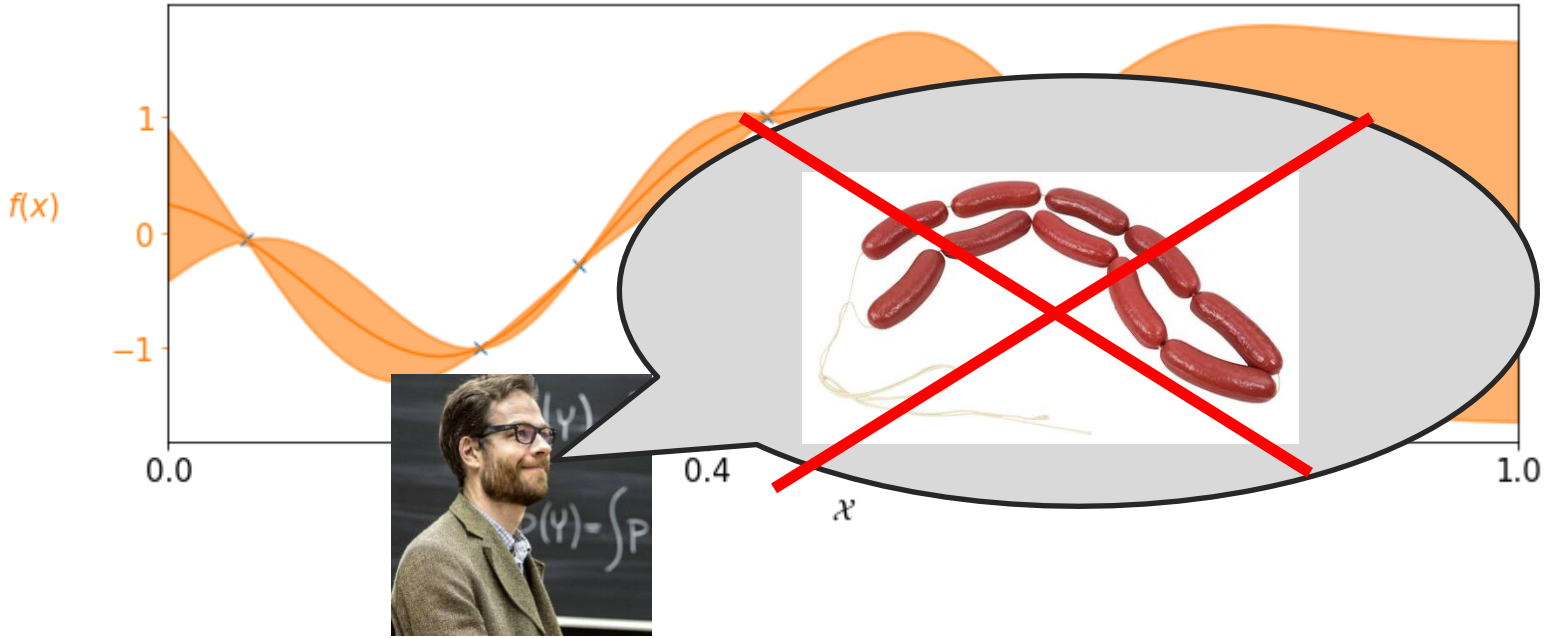
# Uncertainty allows us to define the expected utility of new data

Use a statistical model like a Gaussian process



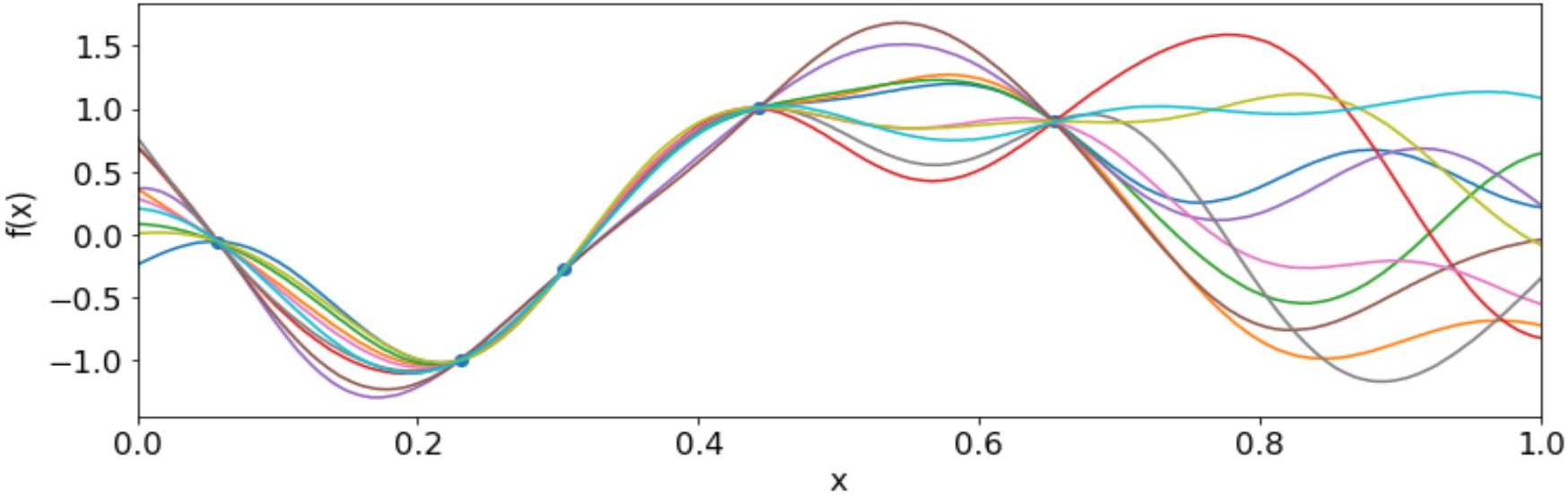
# Uncertainty allows us to define the expected utility of new data

Use a statistical model like a Gaussian process



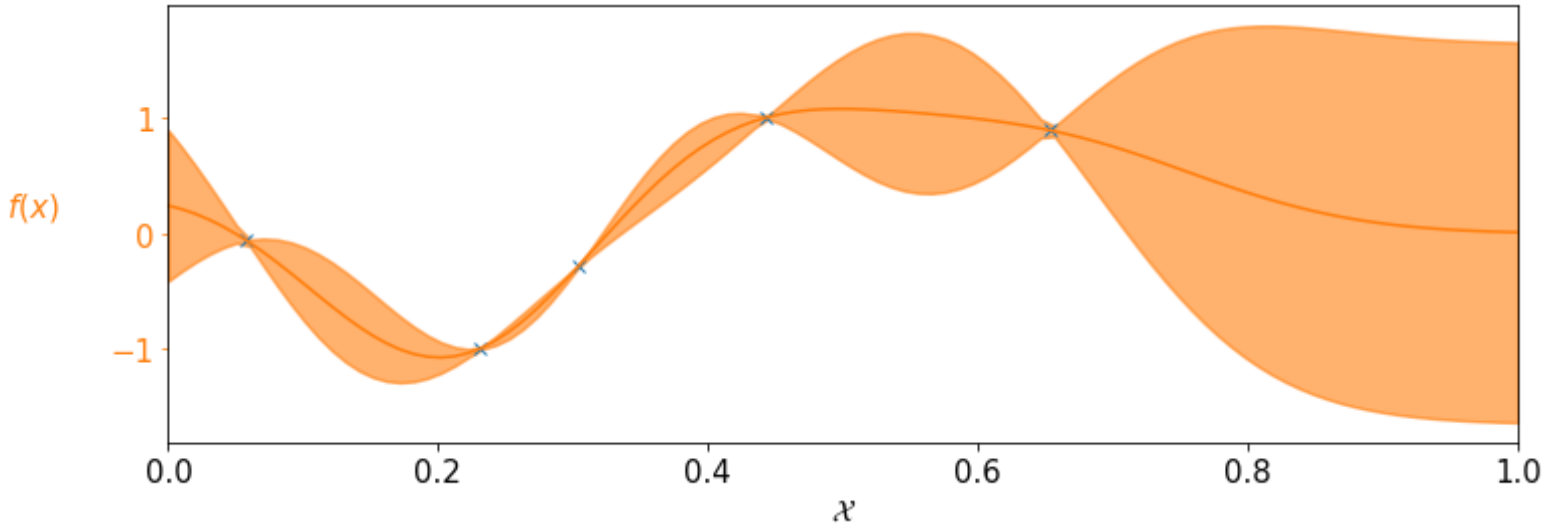
# Uncertainty allows us to define the expected utility of new data

Use a statistical model like a Gaussian process



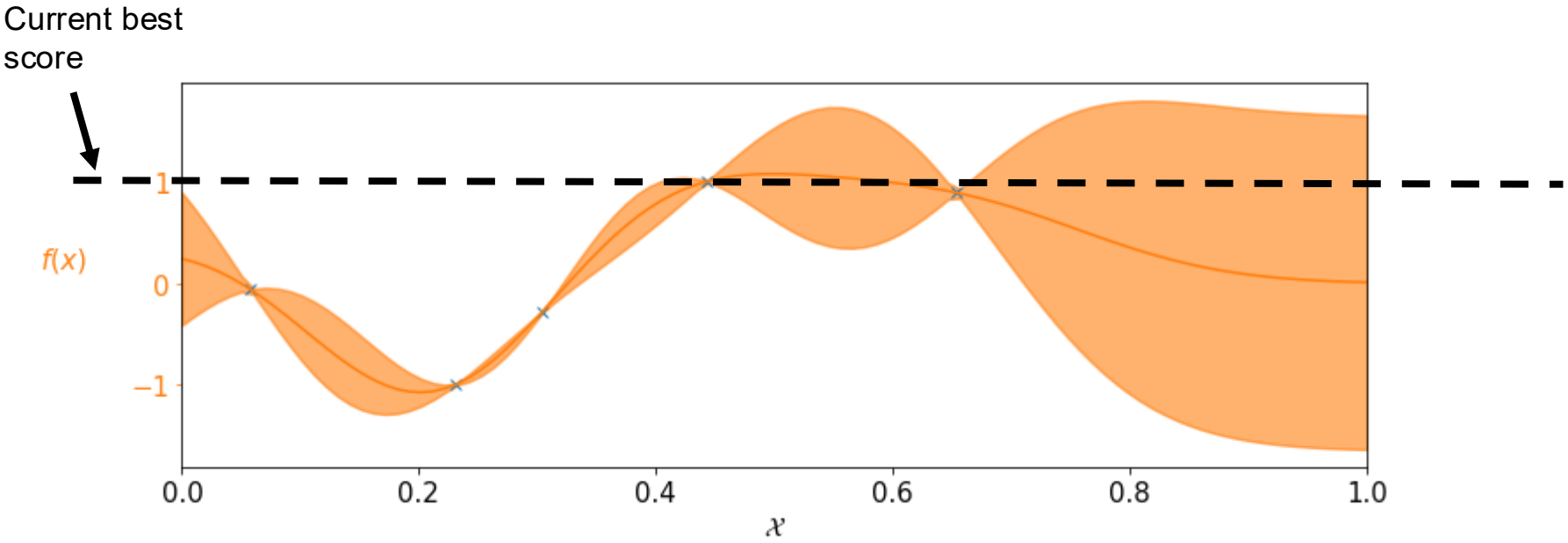
# Uncertainty allows us to define the expected utility of new data

Use a statistical model like a Gaussian process



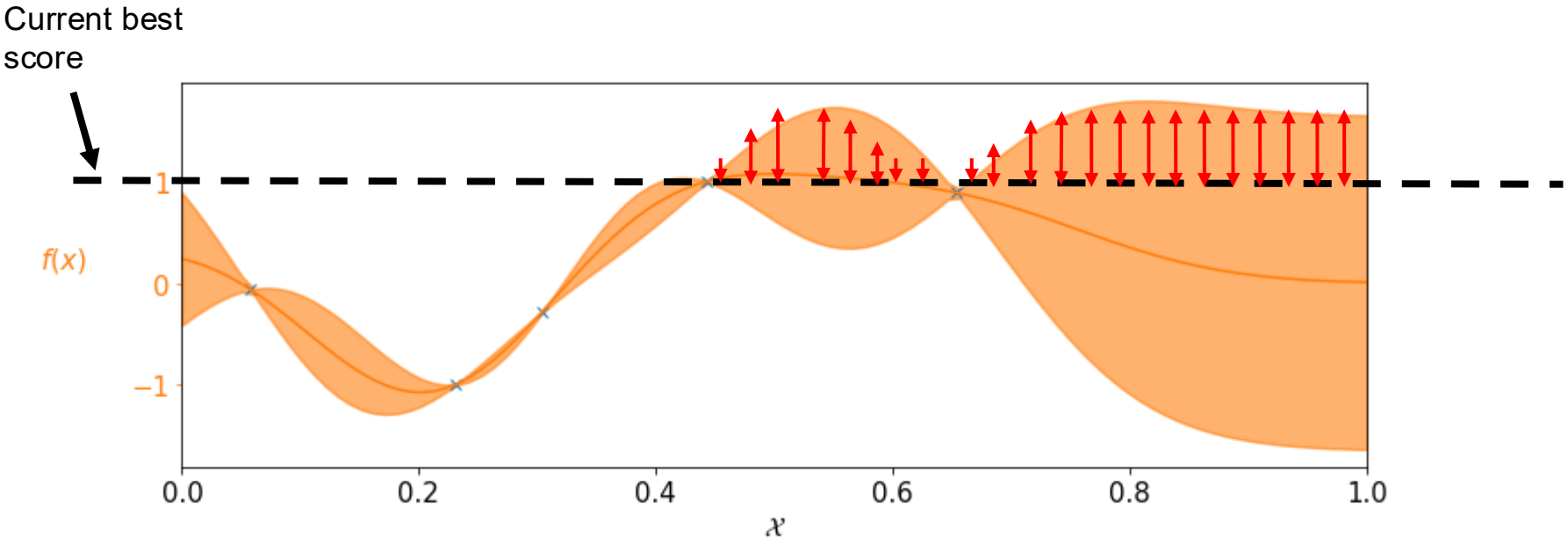
# Uncertainty allows us to define the expected utility of new data

We want to improve over our current best solution



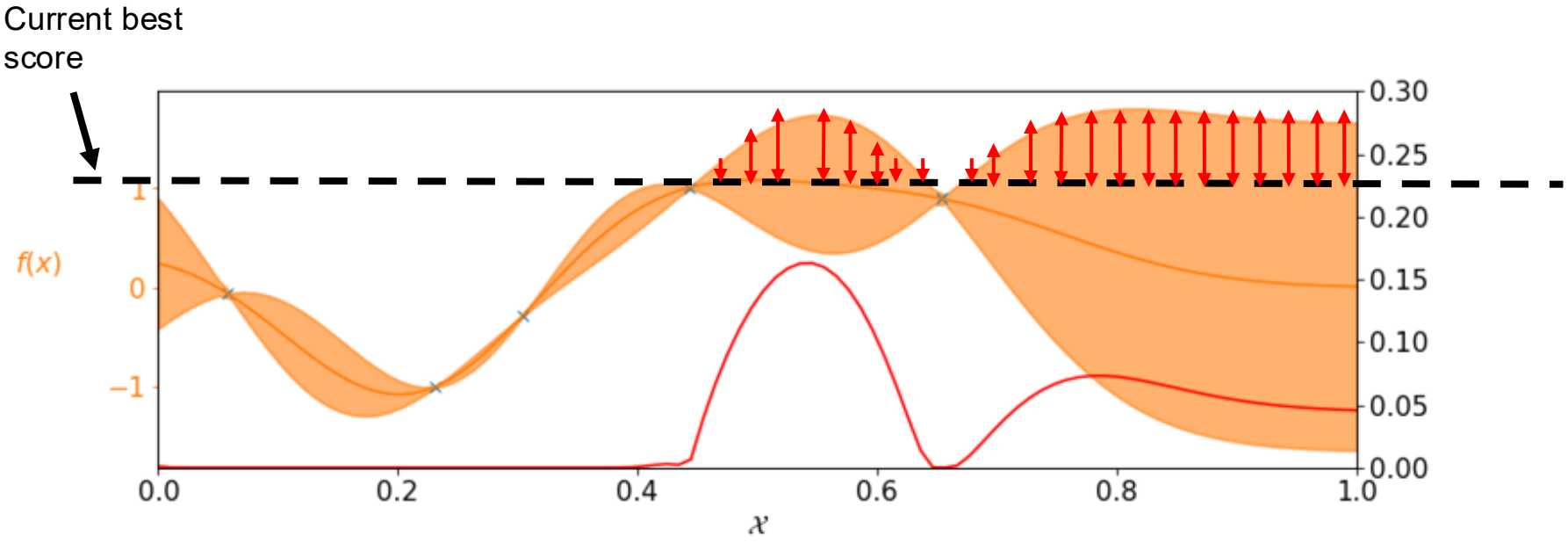
# Uncertainty allows us to define the expected utility of new data

What's the probability of improving ?



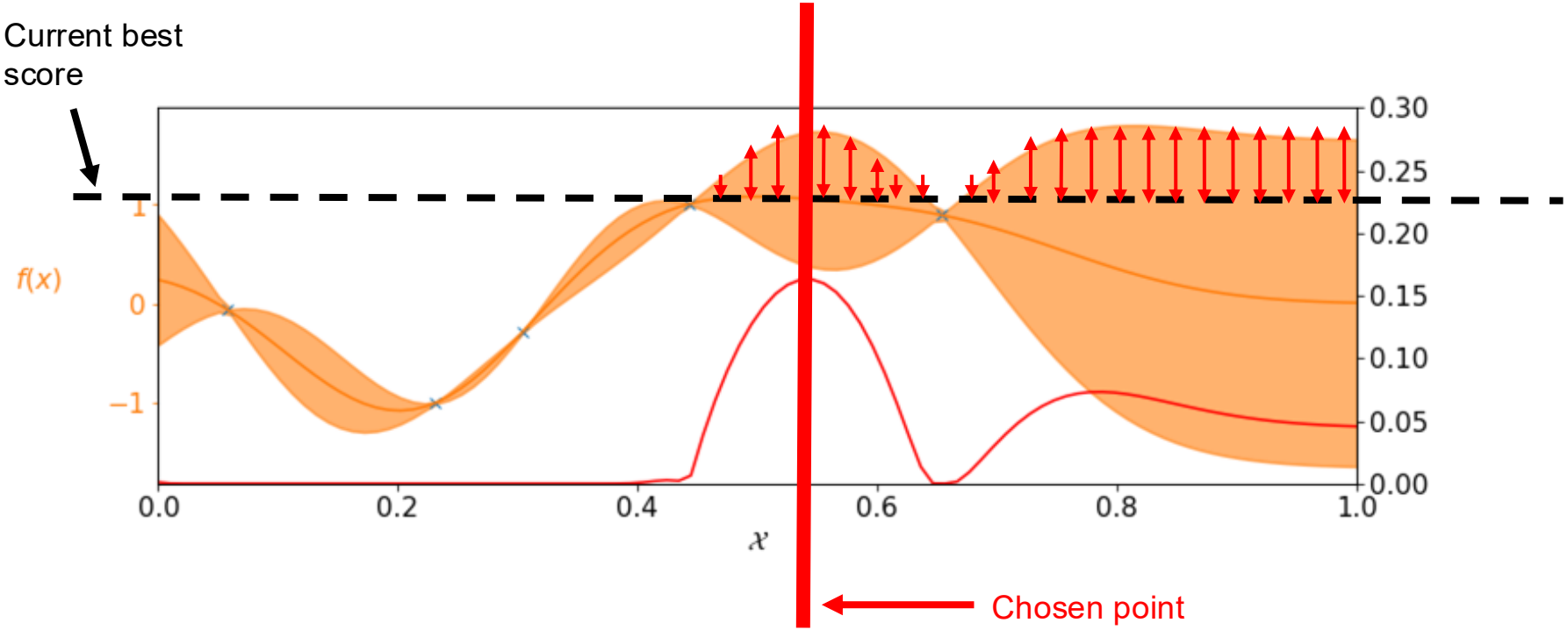
# Uncertainty allows us to define the expected utility of new data

What's the probability of improving ?



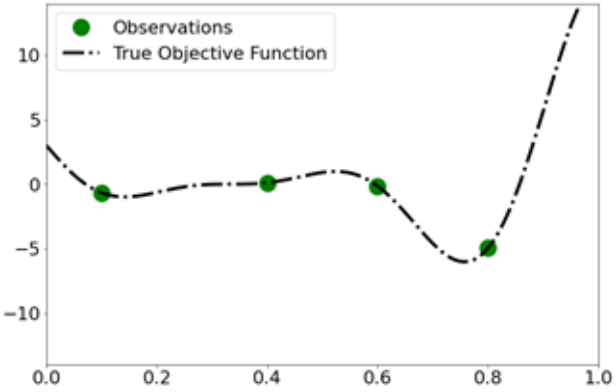
# Uncertainty allows us to define the expected utility of new data

What's the probability of improving ?



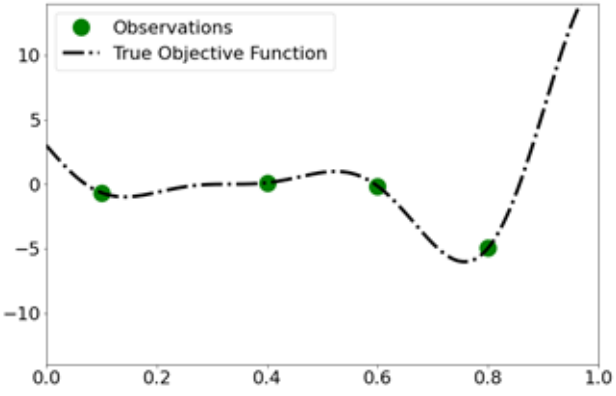
# Automated decision-making

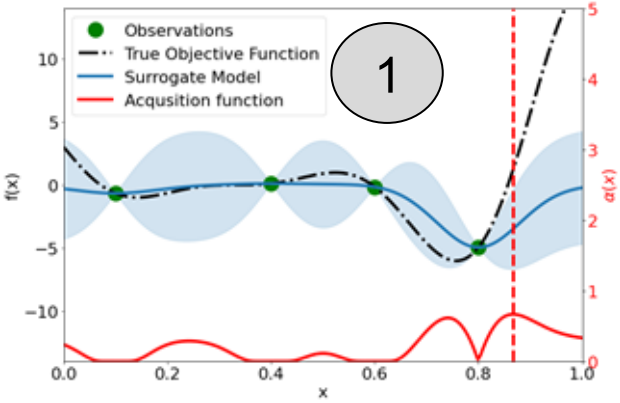
Demo BO loop



# Automated decision-making

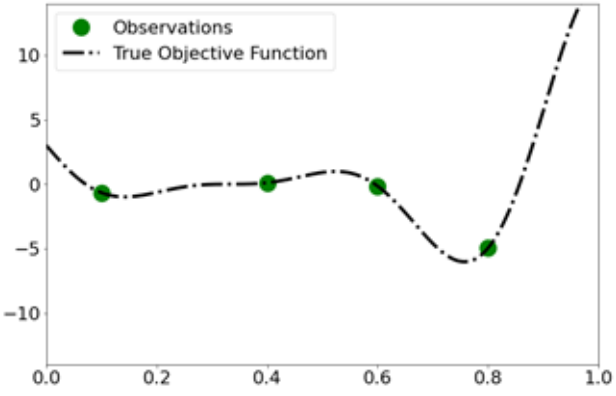
Demo BO loop

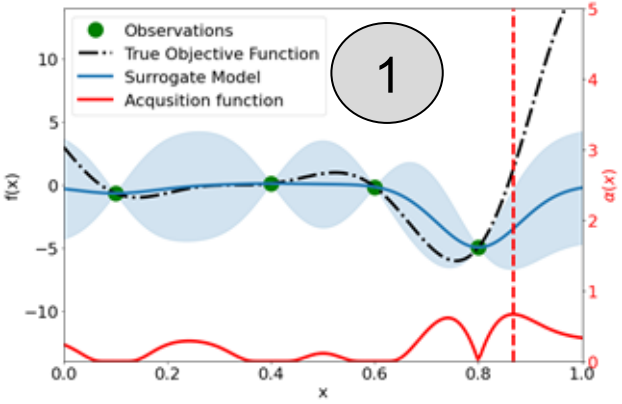


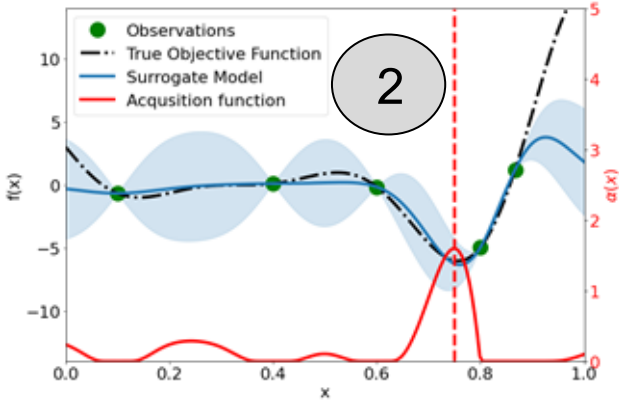



# Automated decision-making

Demo BO loop

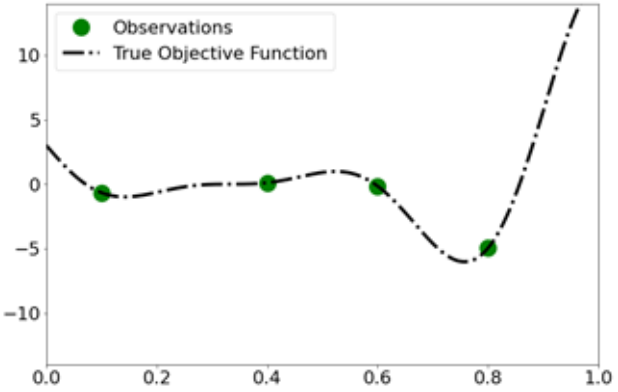


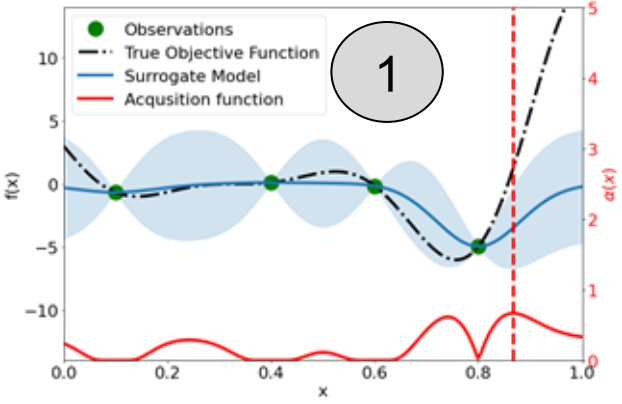



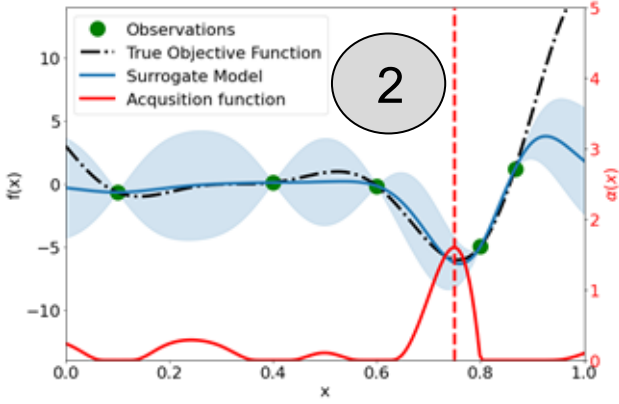



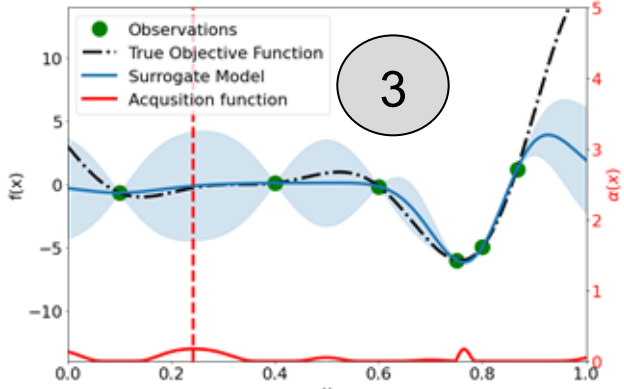
# Automated decision-making

Demo BO loop



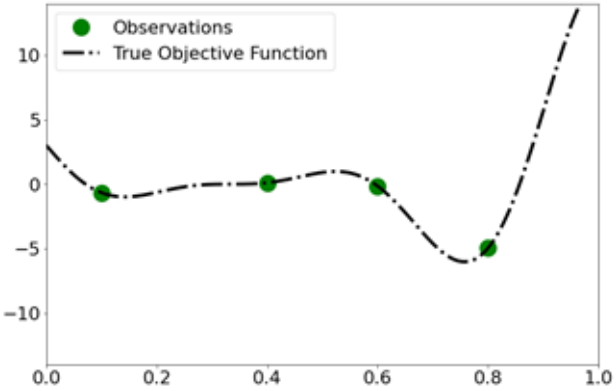



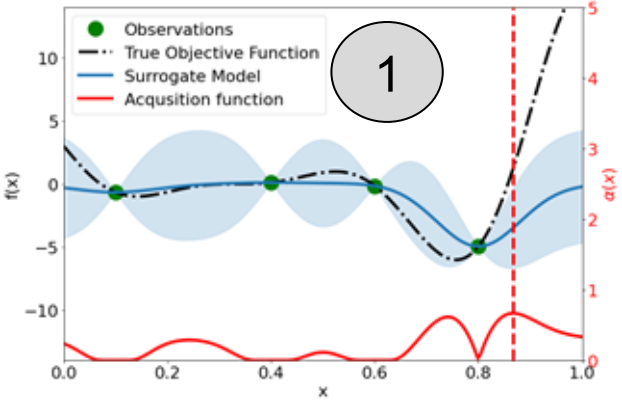



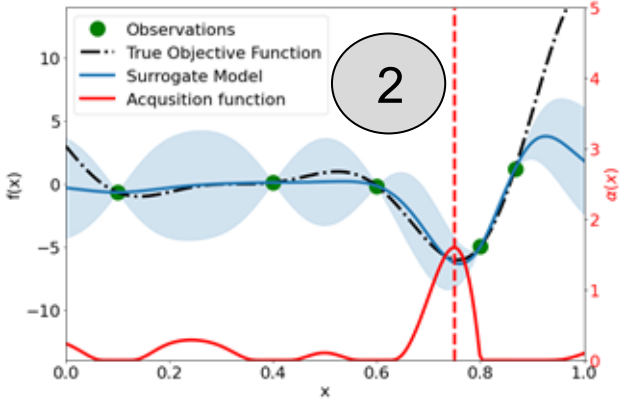



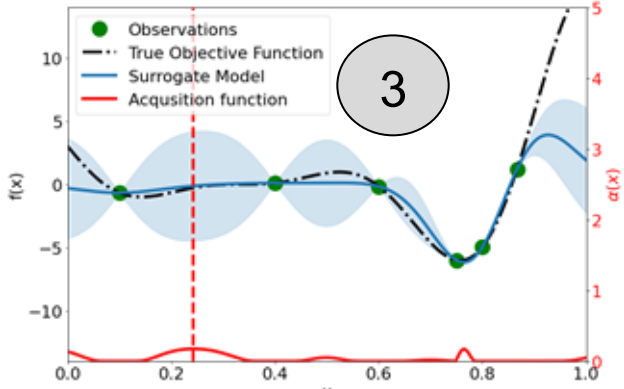
# Automated decision-making

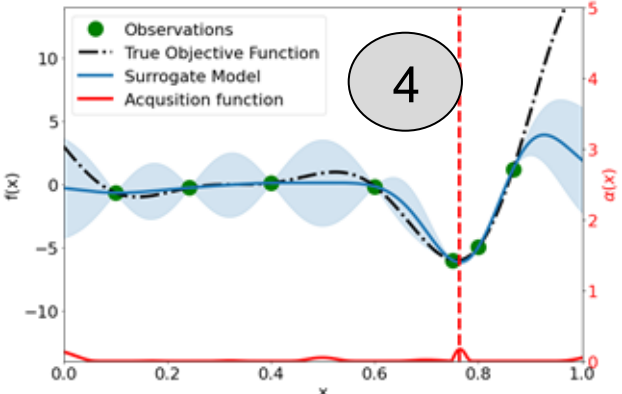
Demo BO loop



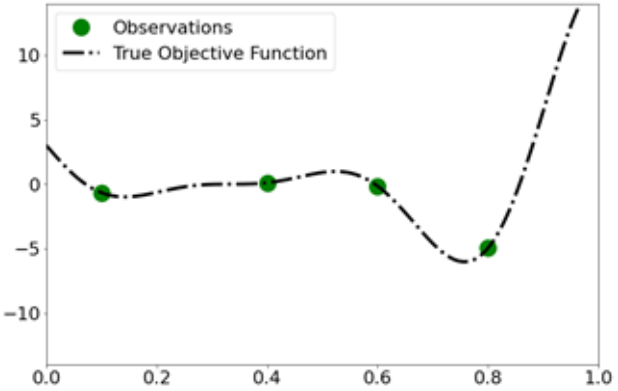



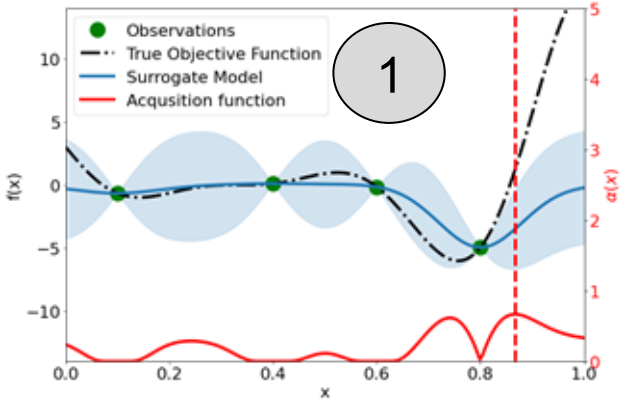



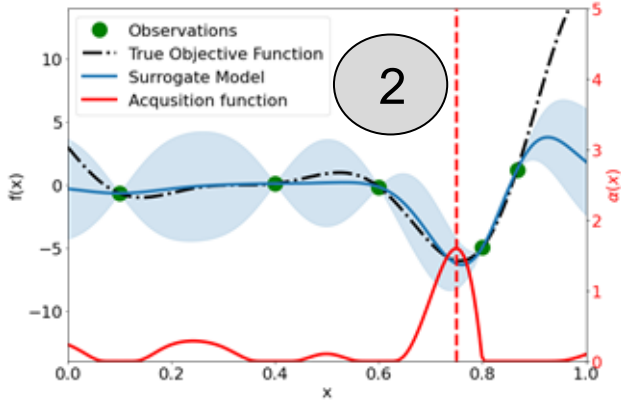



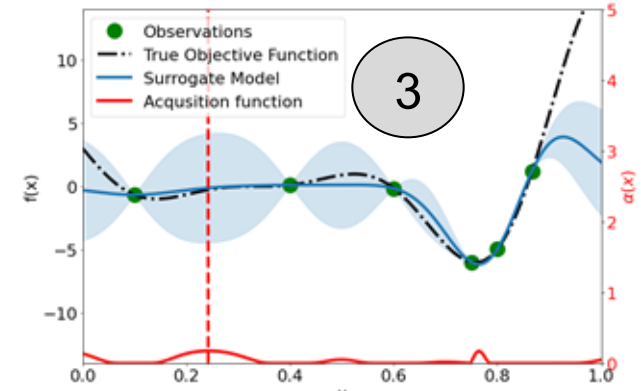
# Automated decision-making

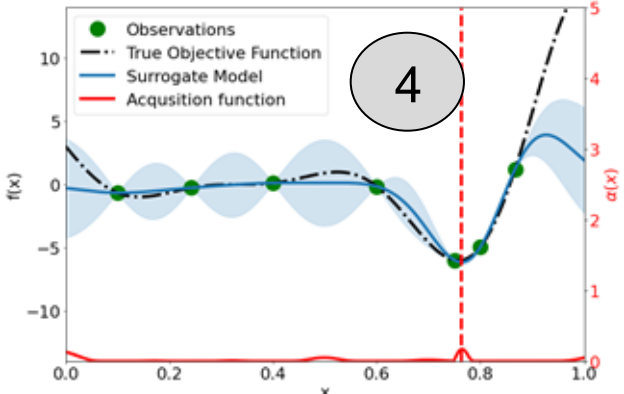
Demo BO loop

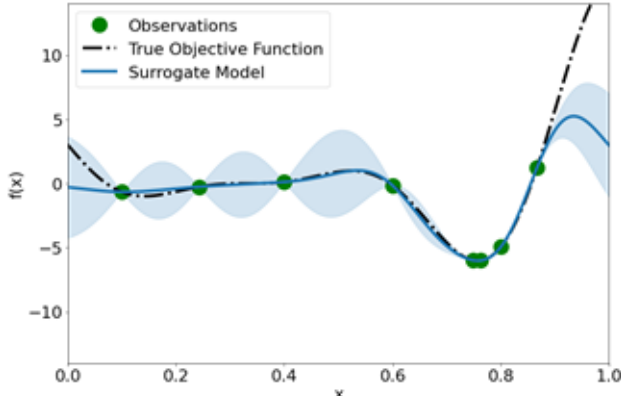


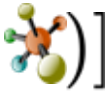
# Utility functions

- $U_f($  $)$: what is the utility of evaluating  (if it will return $f$ )

# Utility functions

- $U_f($  $)$: what is the utility of evaluating  (if it will return $f$ )
  - $f^\star$ Is best so far
  - Has there been an improvement? $U_f($  $) = \mathbb{1}_{(f>f^\star)}$
  - How big was the improvement? $U_f($  $) = \max(f - f^\star, 0)$

# Expected Utility Functions for GPs

- $\alpha($  $) = \mathbb{E}_f[U_f($  $)]$ : what utility is predicted by my model of $f$

$$f \sim \mathcal{N}(\mu, \sigma^2)$$

# Expected Utility Functions for GPs

- $\alpha(\text{  }) = \mathbb{E}_f[U_f(\text{  })]$ : what utility is predicted by my model of $f$

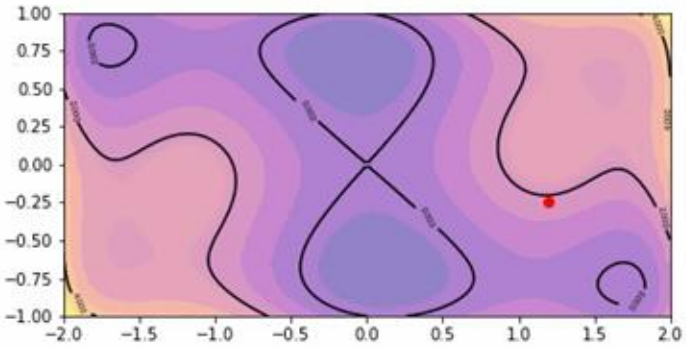
$$f \sim \mathcal{N}(\mu, \sigma^2)$$

- What the probability of improvement? $\alpha_{\text{PI}}(\text{  }) = \mathbb{E}_f\left[\mathbb{1}_{(f>f^\star)}\right]$
- How much improvement do we expect? $\alpha_{\text{EI}}(\text{  }) = \mathbb{E}_f[\max(f - f^\star, 0)]$

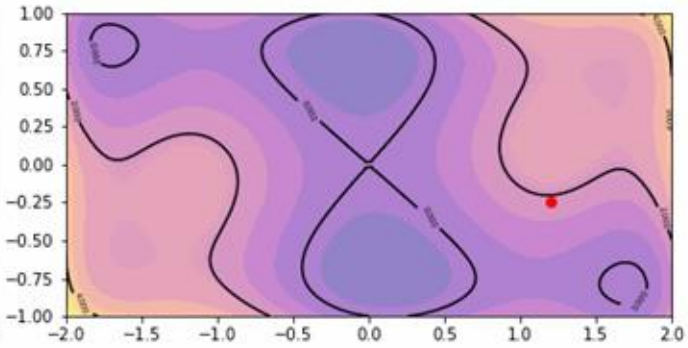
# BO Demo 2

Bayesian optimization is a global optimizer

Bayesian optimization (global)

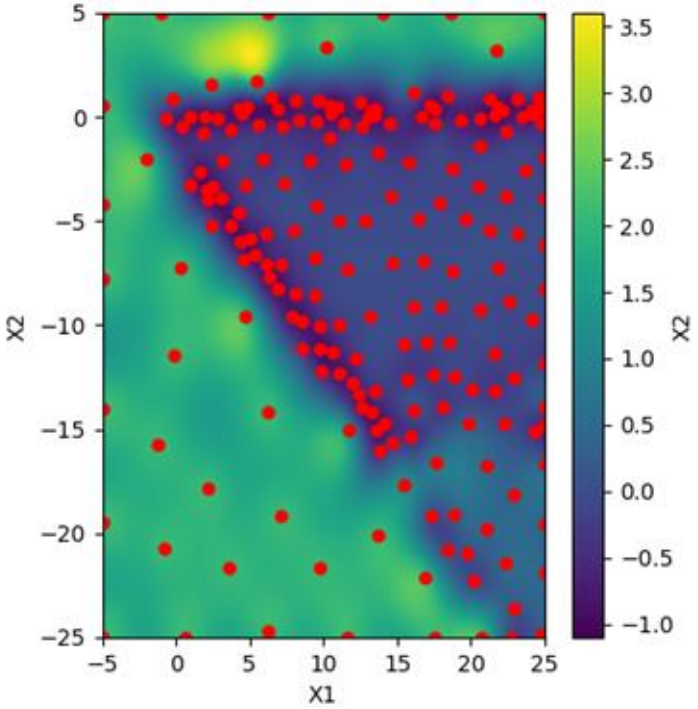



Gradient descent (local)



# BO Demo 3

Efficient coverage of the search space



# So why Bayesian Optimization?

- BO performs **global** optimization
- We do not need gradients or noiseless observations
- BO can optimize under a **limited evaluation budget**
  - Simulating performance of a car engine (mins)
  - Training a large ML model (hours)
  - Synthesising a new molecule (weeks)
  - Testing performance of a wind turbine in real world (months

Increasing cost

**BO: clever modelling rather than brute force!**

# So why Bayesian Optimization?

- BO pe **bal** optim
- We do not ne
- B
  - Sim
  - ng a large
  - Synthesising a ale
  - Testing performance of a wind n real world (

ng cost

A great new(ish) :
**https://bayesoptbook.com/**

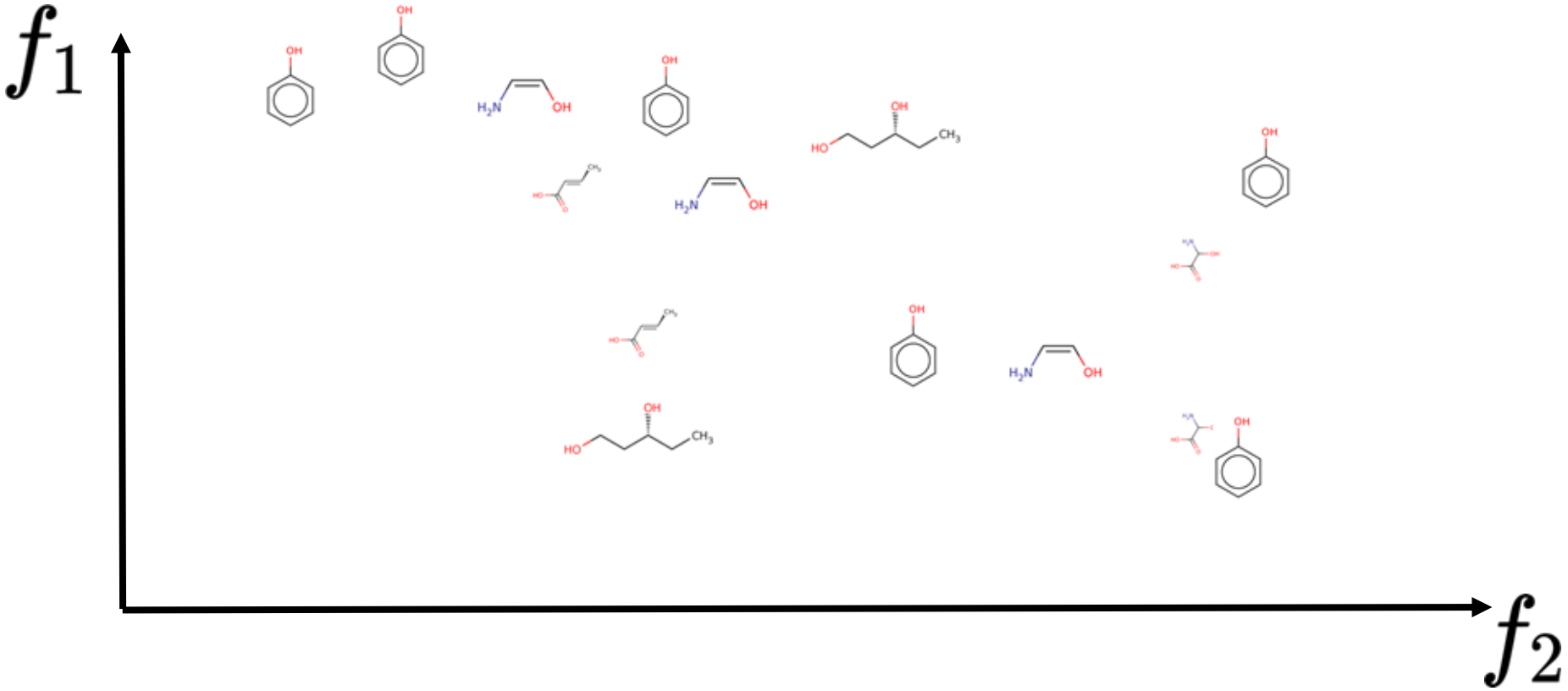
**BO: clever modelli rather than brute force!**

# BO can do it all

- Constraints?
- Multiple objectives?
- Multiple fidelities?
- Batch experiments?

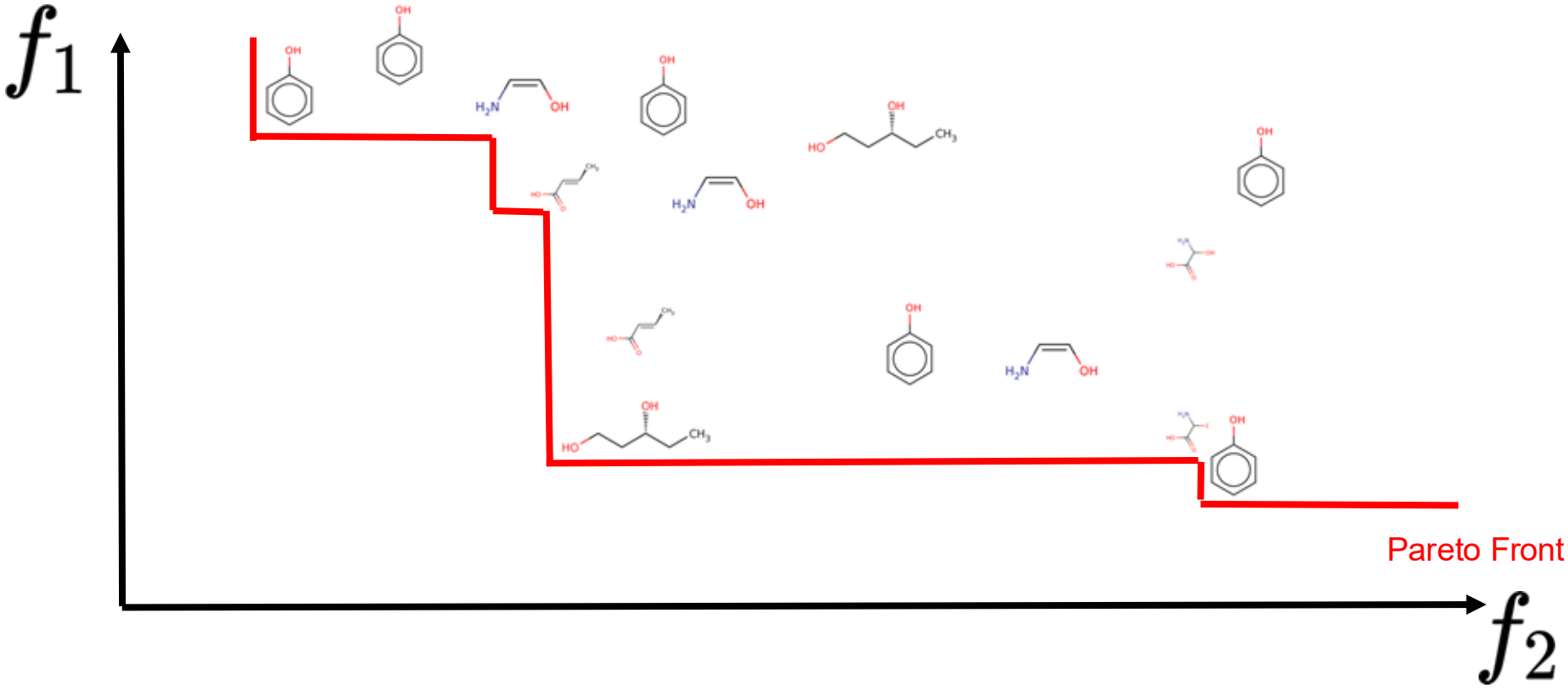
# Multi-objective Optimisation

>1 competing objectives



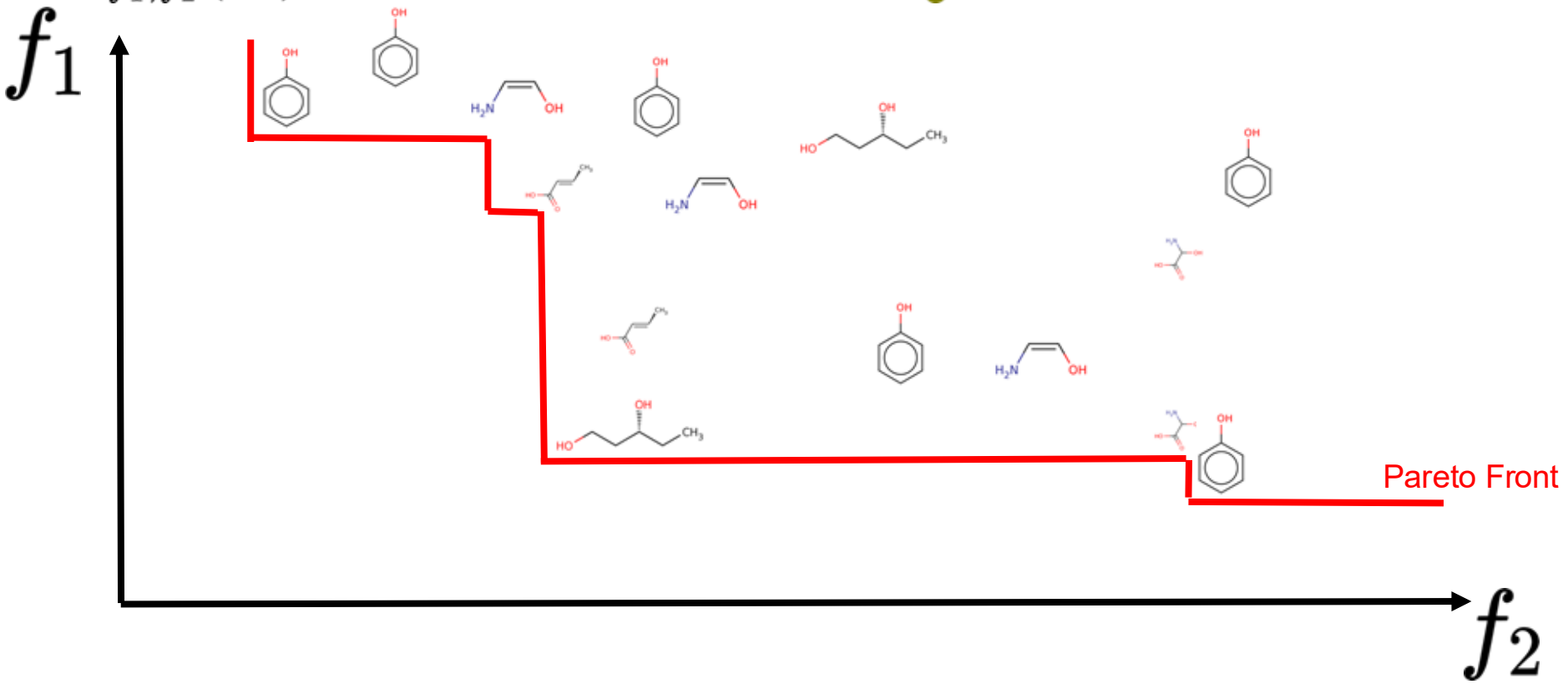
# Multi-objective Optimisation

What do we care about?



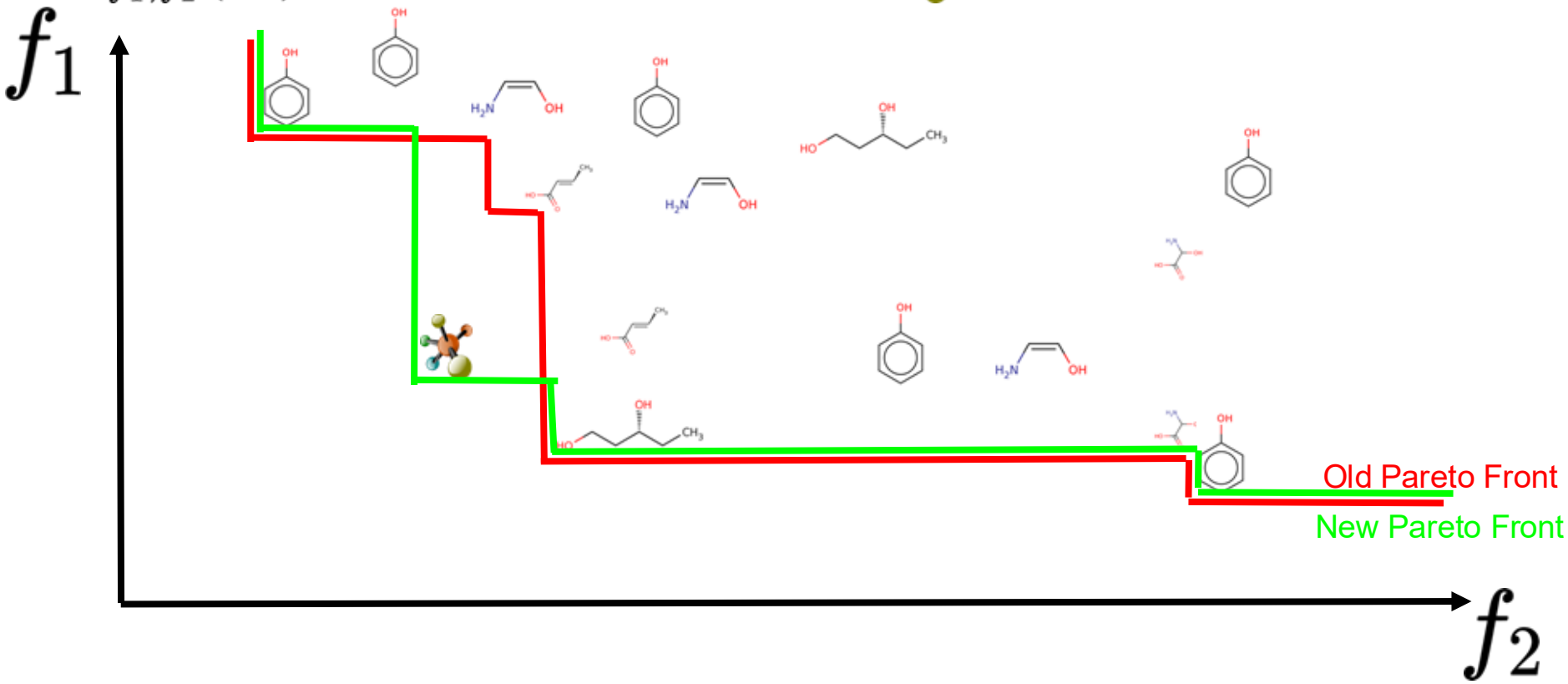
# Multi-objective Optimisation

$U_{f_1,f_2}$(  ): what is the utility of evaluating  if it will return $(f_1, f_2)$



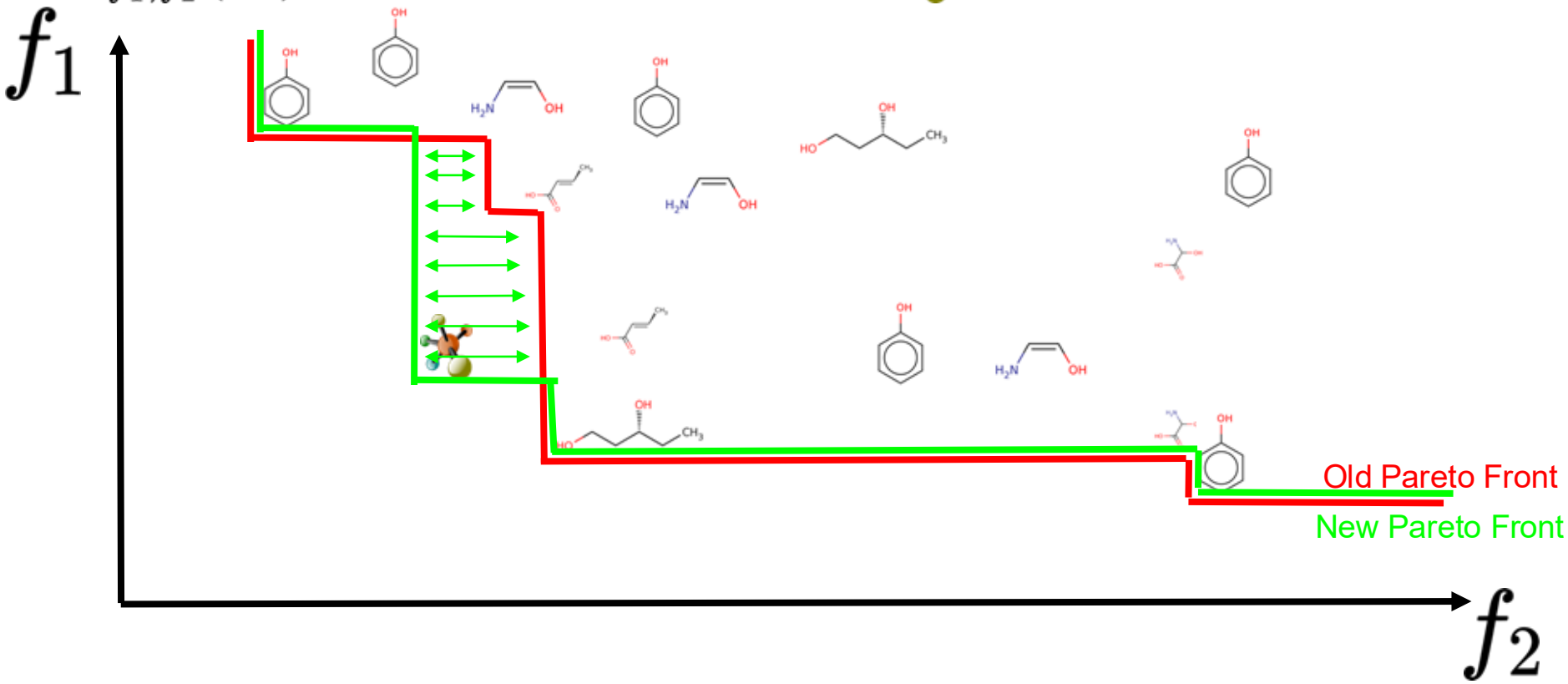
# Multi-objective Optimisation

$U_{f_1,f_2}$(  ): what is the utility of evaluating  if it will return $(f_1, f_2)$



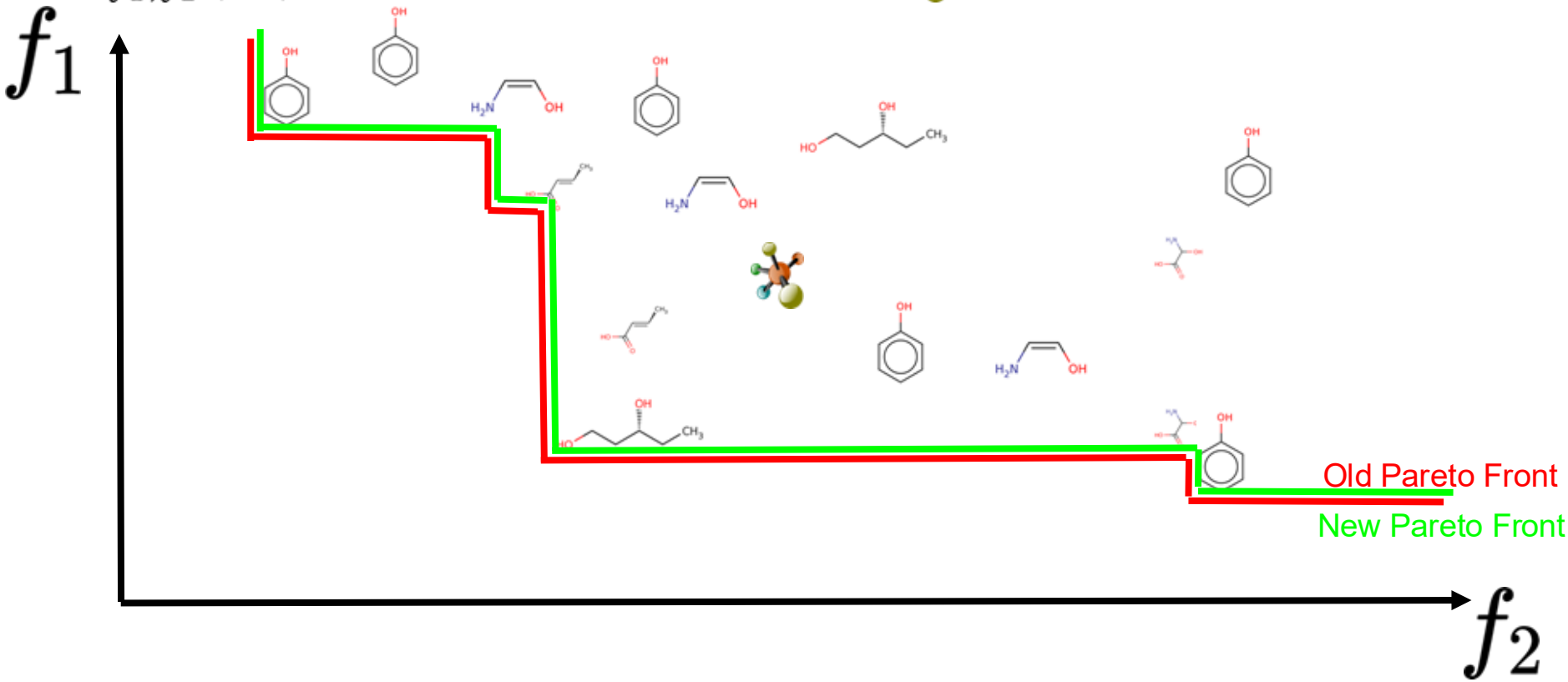
# Multi-objective Optimisation

$U_{f_1,f_2}$(  ): what is the utility of evaluating  if it will return $(f_1, f_2)$



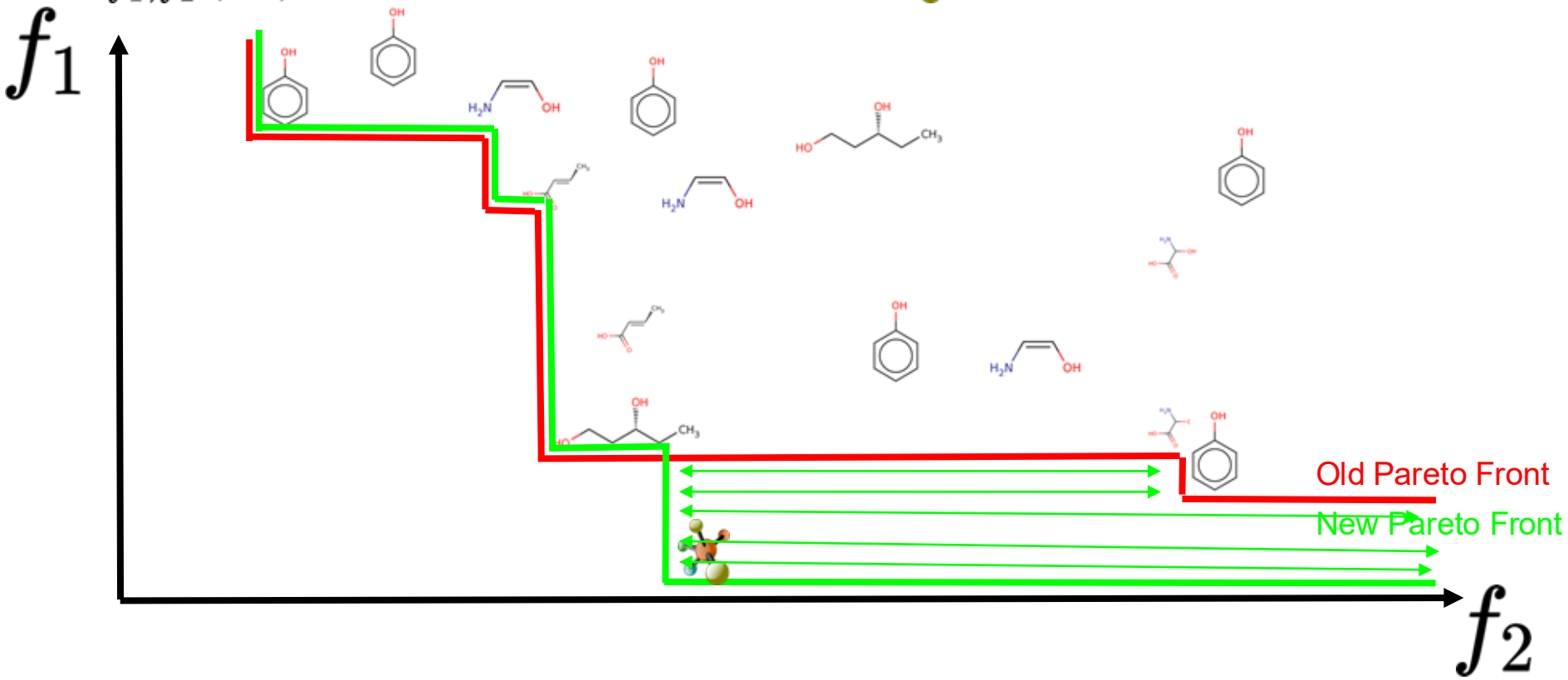
# Multi-objective Optimisation

$U_{f_1,f_2}$(  ): what is the utility of evaluating  if it will return $(f_1, f_2)$



# Multi-objective Optimisation

$U_{f_1,f_2}$(  ): what is the utility of evaluating  if it will return $(f_1, f_2)$



# Multi-objective Optimisation

$U_{f_1,f_2}$(  ): what is the utility of evaluating  if it will return $(f_1, f_2)$

$$\alpha_{\text{EHVI}}(\text{  }) = \mathbb{E}_{f_1,f_2}(U_{f_1,f_2}(\text{  }))$$

$$f_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$$
$$f_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$$

# Multi-objective Optimisation

$U_{f_1,f_2}($  $)$: what is the utility of evaluating  if it will return $(f_1, f_2)$

$$\alpha_{\text{EHVI}}(\text{  }) = \mathbb{E}_{f_1,f_2}(U_{f_1,f_2}(\text{  }))$$

$$f_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$$
$$f_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$$

$$\alpha_{\text{EHVI}}(\ \{\text{  }_i, \text{  }_j\}\ ) = ???$$

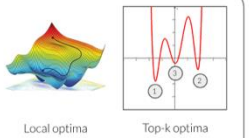
# Active learning

**BO is just one of the  things we can do.....**

# Active learning

Sequentially choose data that reduces uncertainty in the thing we care about

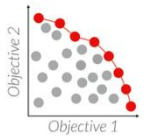
**Optimization Variants**
(global, local, top-k optima)




Applications

- *ML model training*
- *HPO / NAS*
- *Systems tuning*

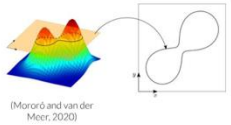
**Multi-objective Opt.**
(Pareto frontiers)




Applications

- *Process optimization*
- *Portfolio optimization*
- *Laboratory equipment / machines*

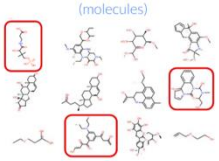
**Level Set Estimation**
(superlevel set, sublevel set)




Applications

- *Catalyst design*
- *Active learning / weak supervision*
- *Environmental monitoring*

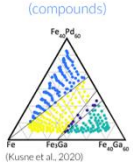
**Search**
(subset matching criteria)




Applications

- *Drug discovery*
- *Fraud detection*
- *Targeted opinion polling*

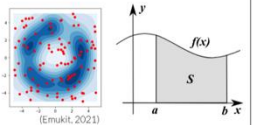
**Phase Identification**
(boundaries, partitions)




Applications

- *High throughput materials design/discovery*

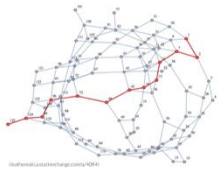
**Quadrature, Integration**
(integrals, expectations)




Applications

- *Probabilistic modeling* (marginal distributions, normalization constants)
- *Estimating center of mass* (and centroids)

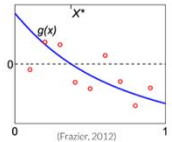
**Graph Algorithms**
(shortest paths)



Applications

- *Transportation networks*
- *Shipping networks*
- *Social networks*

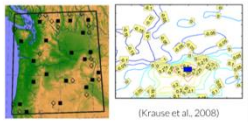
**Root Finding, Bisection**
(roots)




Applications

- *Target localization* (airborne radar)
- *Edge detection* (computer vision)
- *Biology / microbiology* (phase shifts)

**Sensor Placement**
(function value at locations)

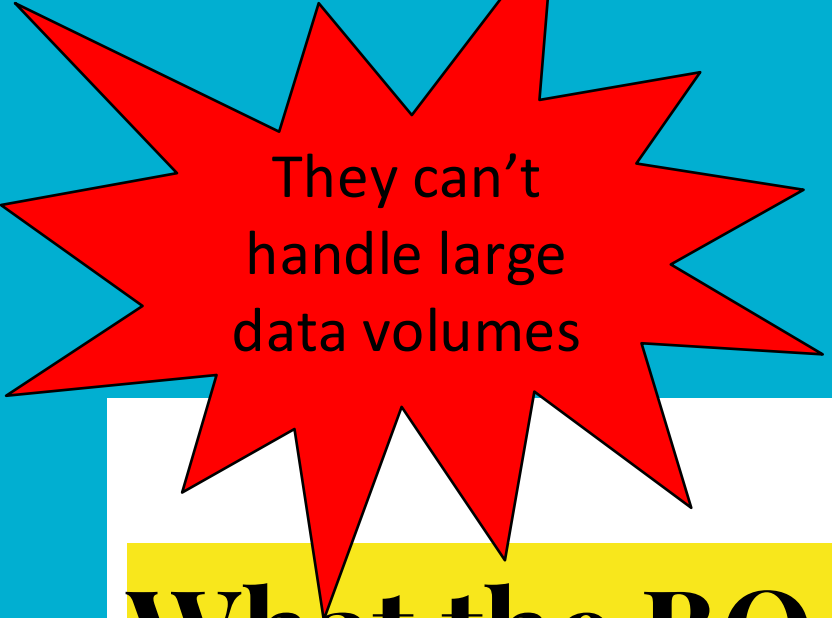



Applications

- *Water distribution systems*
- *Outbreak detection in networks*
- *Weather monitoring*

Bayesian Algorithm Execution
https://willieneis.github.io/bax-website/

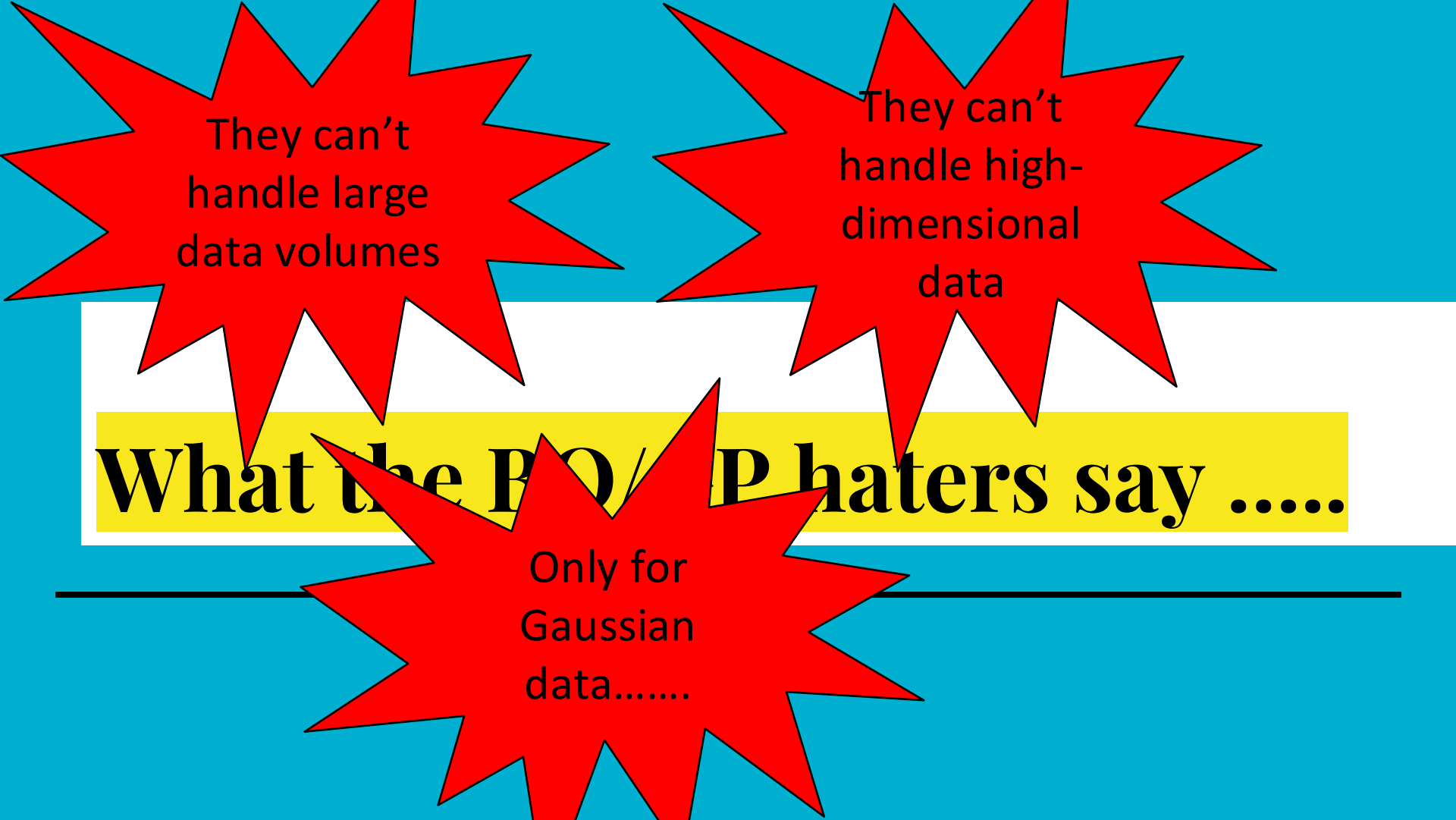
# What the BO/GP haters say .....




# What the BO/GP haters say .....

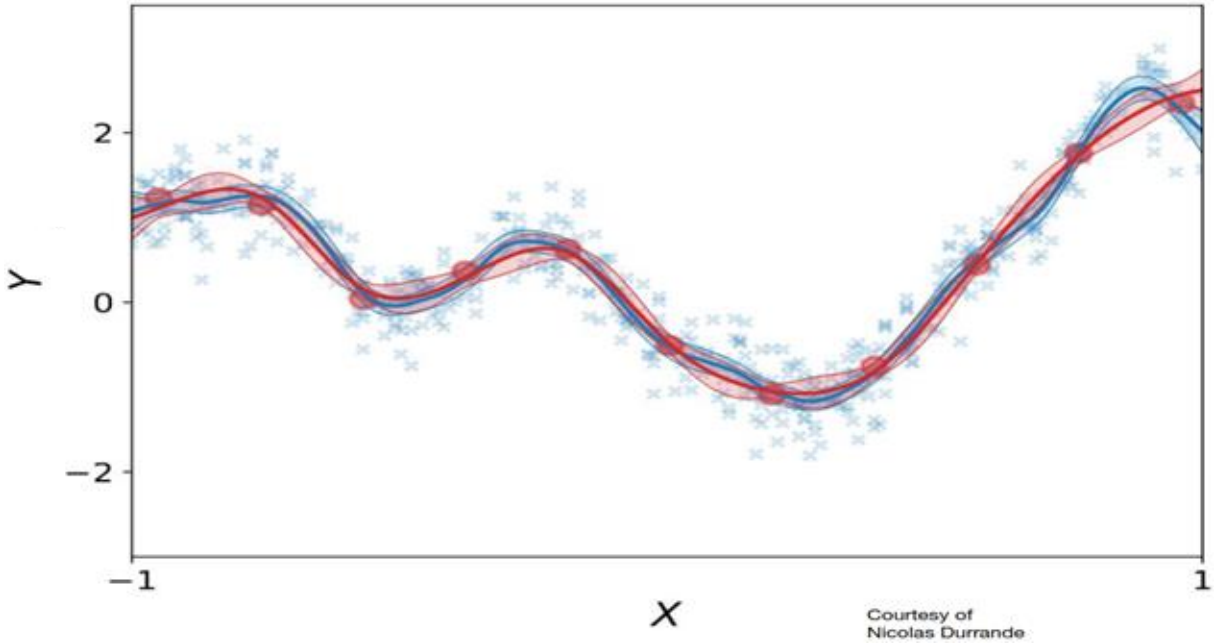


# What the BO/GP haters say .....


They can’t handle large data volumes
They can’t handle high-dimensional data
What the BQ/ P haters say …..
Only for Gaussian data…….

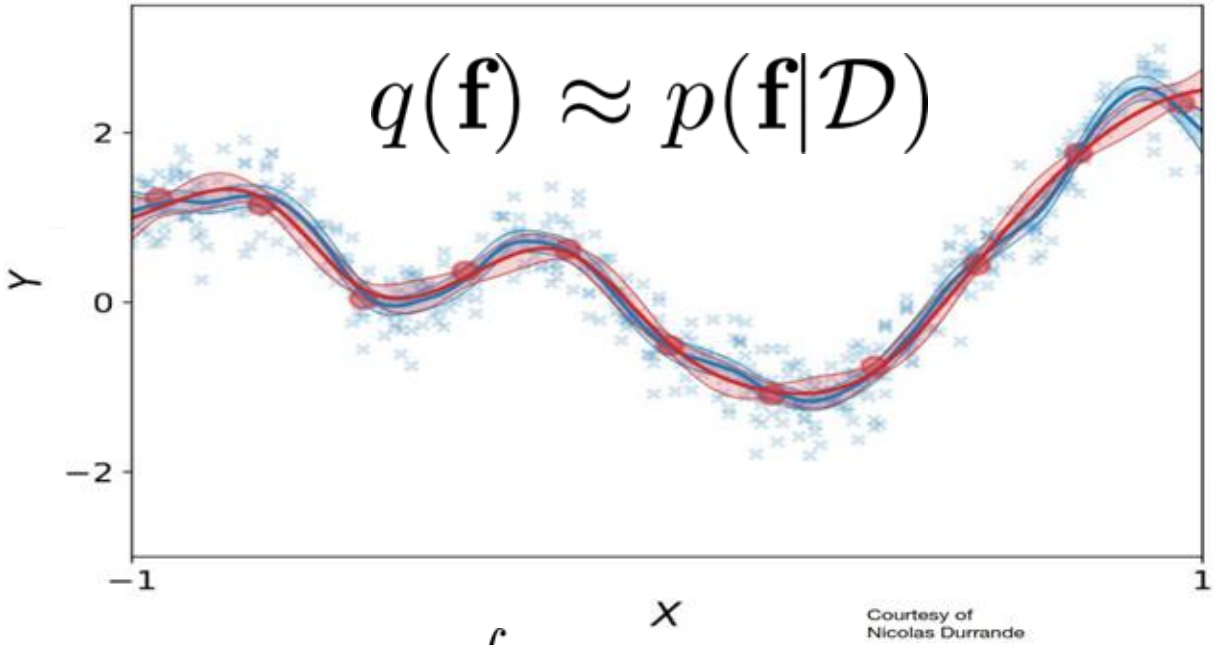
# SVGPs: Gaussian processes for big data

Use Sparse Gaussian Processes (replace N data with M inducing points)



# SVGPs: Gaussian processes for big data

Use Sparse Variational Gaussian Processes (replace N data with M inducing points)



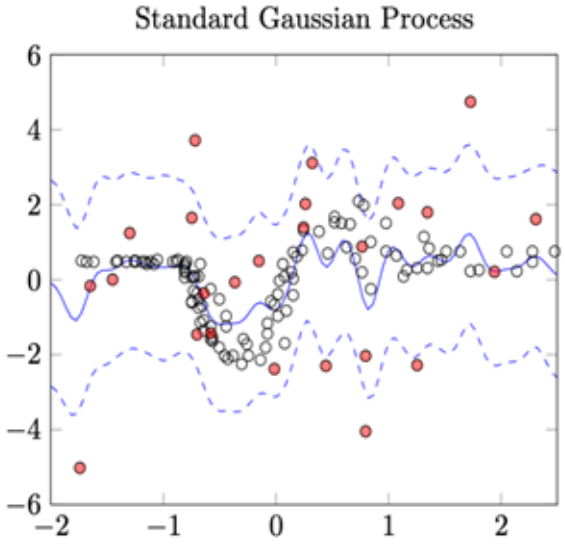

Courtesy of Nicolas Durrande

$$\mathrm{ELBO}(q(\mathbf{f})) = \int q(\mathbf{f}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} - \mathcal{KL}(q(\mathbf{f}), p(\mathbf{f}))$$

# SVGPs: Non-gaussian data

$$y_i \sim \mathcal{N}(f(\mathbf{x}_i), \sigma^2)$$

Saul et al. 2016

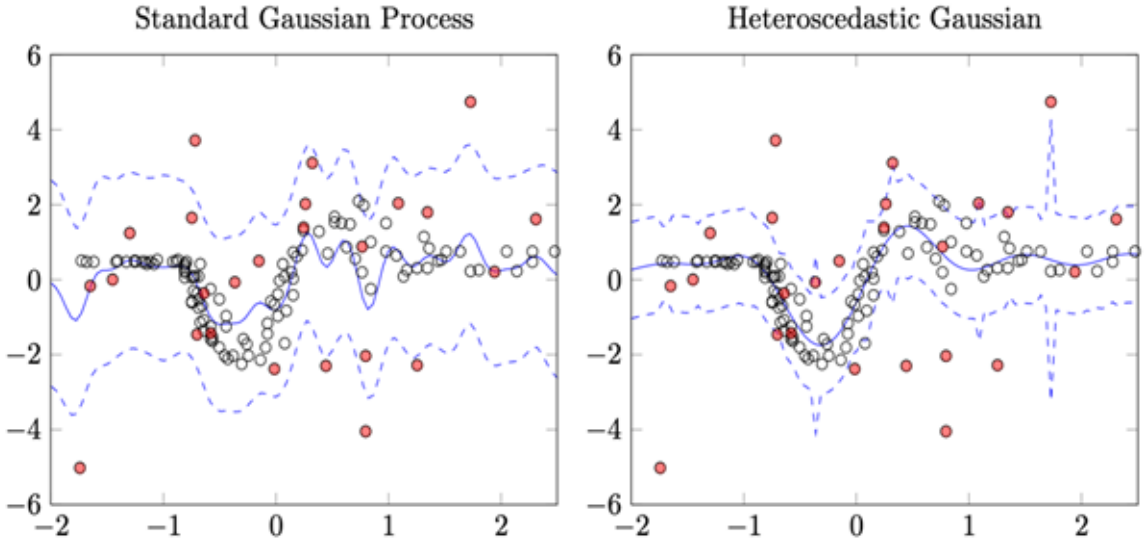


# SVGPs: Non-gaussian data

$$y_i \sim \mathcal{N}(f(\mathbf{x}_i), \sigma^2)$$

$$y_i \sim \mathcal{N}(f_0(\mathbf{x}_i), e^{f_1(\mathbf{x}_i)})$$

Saul et al. 2016



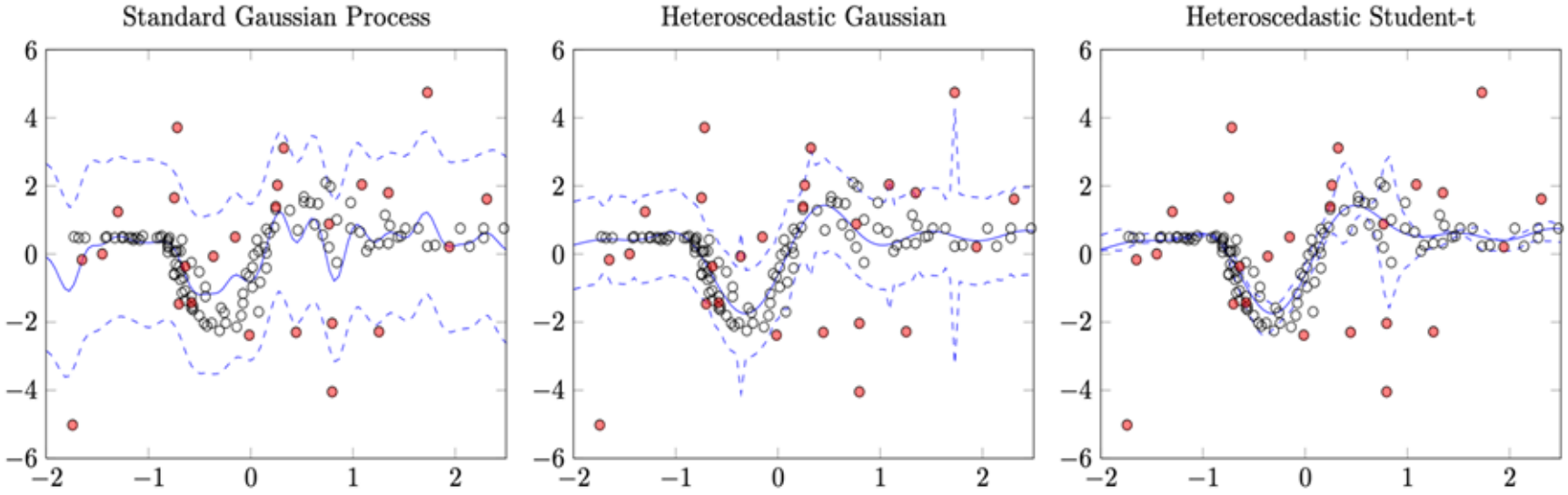
# SVGPs: Non-gaussian data

$$y_i \sim \mathcal{N}(f(\mathbf{x}_i), \sigma^2)$$

$$y_i \sim \mathcal{N}(f_0(\mathbf{x}_i), e^{f_1(\mathbf{x}_i)})$$

$$y_i \sim \mathcal{S}t(f_0(\mathbf{x}_i), e^{f_1(\mathbf{x}_i)}, \nu)$$

Saul et al. 2016




# Beware the curse of dimensionality ....


Hvafner et al. 2025

Doumont et al. 2026

# Gaussian processes are great in high-dim!

RBF kernels are not!

Hvafner et al. 2025

Doumont et al. 2026

# Gaussian processes are great in high-dim!

RBF kernels are not!

$$l \propto \sqrt{D}$$

$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{||\mathbf{x}-\mathbf{y}||^2}{2l^2}}$$

Hvafner et al. 2025

Doumont et al. 2026

# Gaussian processes are great in high-dim!

RBF kernels are not!

$$l \propto \sqrt{D}$$

$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{||\mathbf{x}-\mathbf{y}||^2}{2l^2}}$$

$$= \prod_{i}^{d} k_i(x_i, y_i)$$

Hvafner et al. 2025

Doumont et al. 2026

# Gaussian processes are great in high-dim!

RBF kernels are not!

$$l \propto \sqrt{D}$$

$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{||\mathbf{x}-\mathbf{y}||^2}{2l^2}}$$

$$= \prod_{i}^{d} k_i(x_i, y_i)$$

AND

Hvafner et al. 2025

Doumont et al. 2026

# Gaussian processes are great in high-dim!

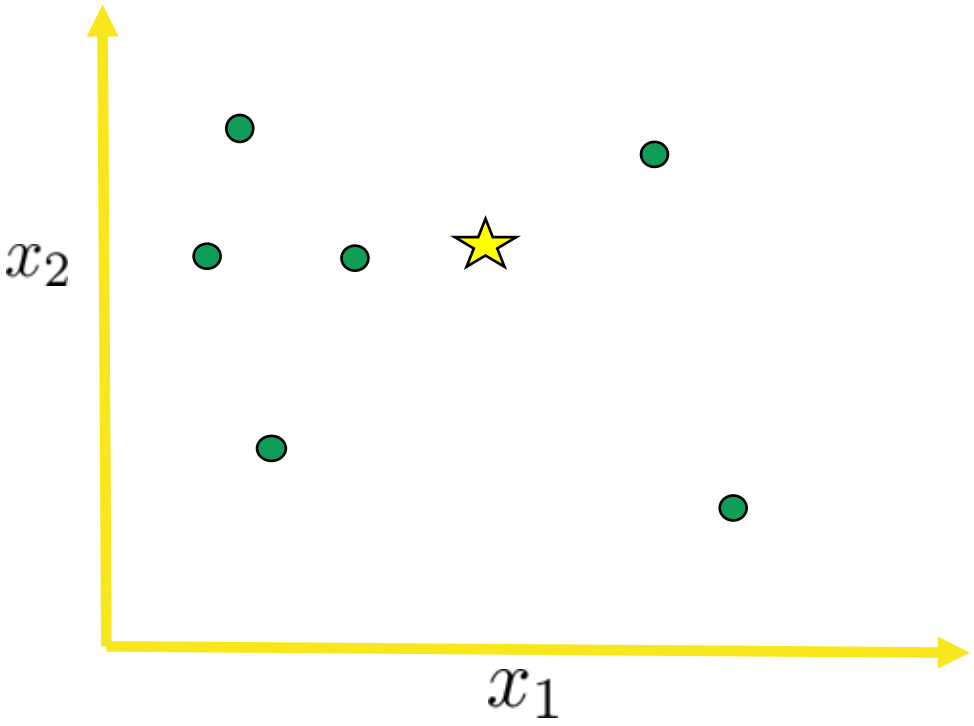
RBF kernels are not!

$$l \propto \sqrt{D}$$

$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{||\mathbf{x}-\mathbf{y}||^2}{2l^2}}$$

$$= \prod_i^d k_i(x_i, y_i)$$

AND



Hvafner et al. 2025

Doumont et al. 2026

# Gaussian processes are great in high-dim!

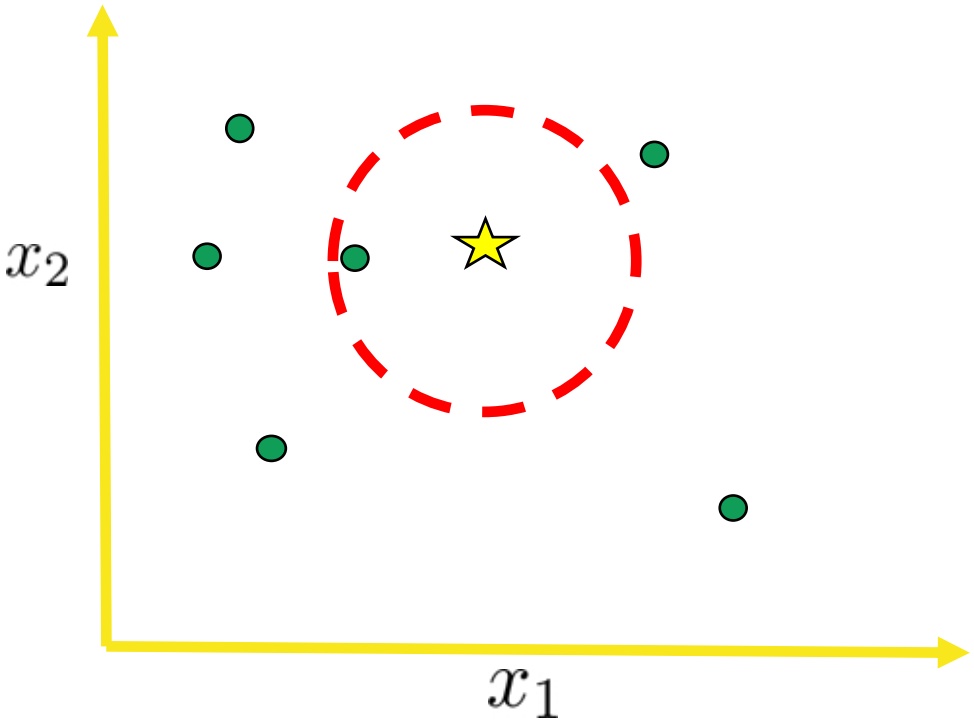
RBF kernels are not!

$$l \propto \sqrt{D}$$

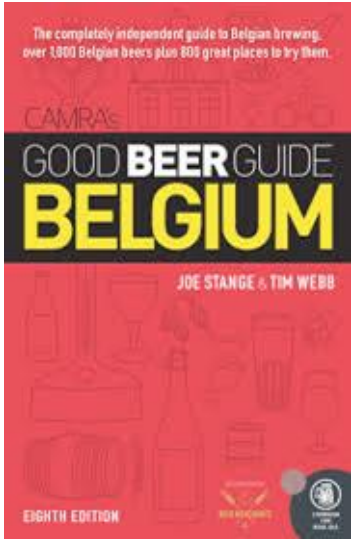
$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{||\mathbf{x}-\mathbf{y}||^2}{2l^2}}$$

$$= \prod_i^d k_i(x_i, y_i)$$

AND

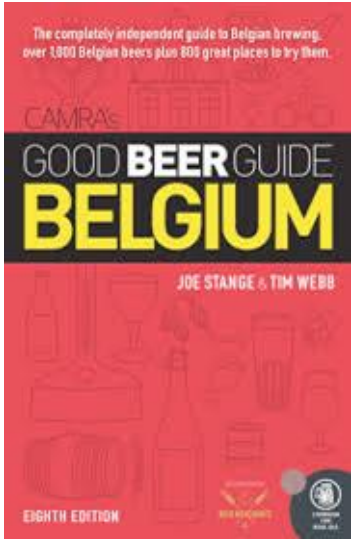


# Gaussian processes are great in high-dim!



# Gaussian processes are great in high-dim!

- Type of fermentation (wild yeast?)
- Ingredients (orange peel?, coriander??????)
- Strength
- Brewed by a monk?
- Barrel-aged?



# Additive rather than multiplicative kernels?

$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{||\mathbf{x}-\mathbf{y}||^2}{2l^2}}$$

$$= \prod_{i}^{d} k_i(x_i, y_i)$$

AND

# Additive rather than multiplicative kernels?

$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{||\mathbf{x}-\mathbf{y}||^2}{2l^2}}$$
$$= \prod_i^d k_i(x_i, y_i)$$

AND

$$k_1(\mathbf{x}, \mathbf{y}) = \sum_i^d k_i(x_i, y_i)$$

OR

# Additive rather than multiplicative kernels?

$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{||\mathbf{x}-\mathbf{y}||^2}{2l^2}}$$

$$= \prod_{i}^{d} k_i(x_i, y_i)$$

AND

$$k_1(\mathbf{x}, \mathbf{y}) = \sum_{i}^{d} k_i(x_i, y_i)$$

OR

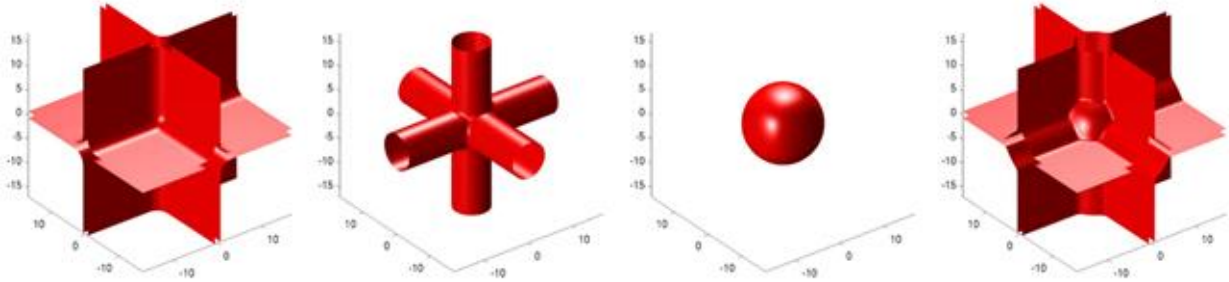
$$k_2(\mathbf{x}, \mathbf{y}) = \sum_{i<j}^{d} k_i(x_i, y_i) k_j(x_j, y_j$$

(Duvenaud et al 2011)

# Additive Gaussian Processes

$$k(x,y) = k_0 + \sum k_i(x_i, y_i) + \sum_{i<j} k_i(x_i, y_i) k_j(x_j . y_j)$$

0 1 2



1st order interactions
$k_1 + k_2 + k_3$

2nd order interactions
$k_1 k_2 + k_2 k_3 + k_1 k_3$

3rd order interactions
$k_1 k_2 k_3$
(Squared-exp kernel)

All interactions

(Additive kernel)

(Duvenaud et al 2011)

# Additive Gaussian Processes

$$k(x,y) = \overset{0}{k_0} + \sum \overset{1}{k_i}(x_i, y_i) + \sum_{i<j} \overset{2}{k_i}(x_i, y_i) k_j(x_j . y_j)$$

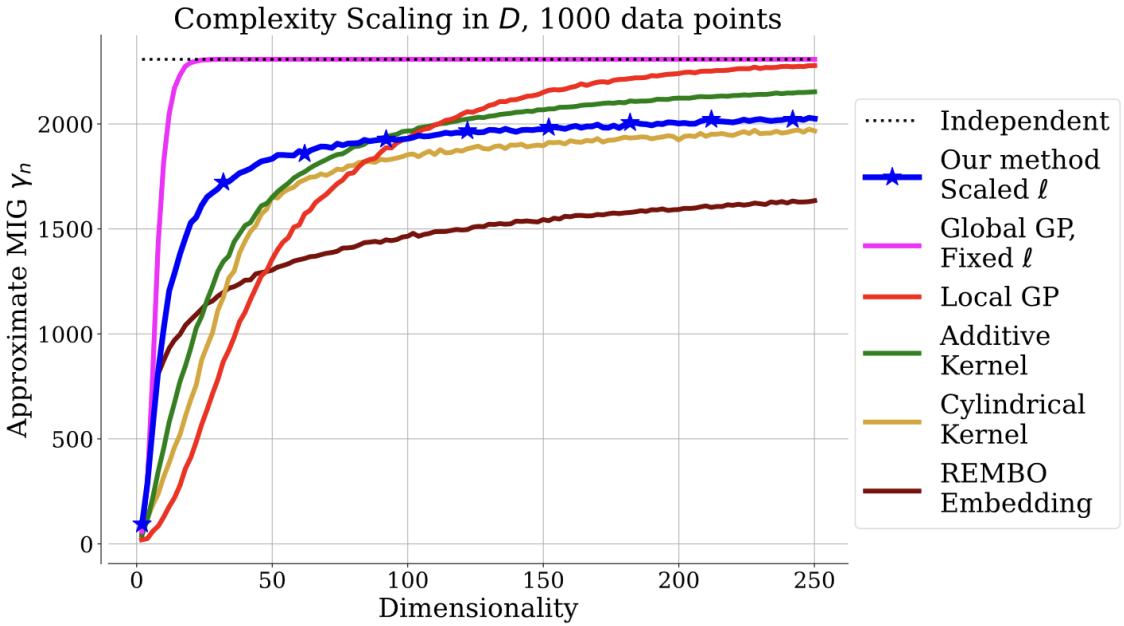
$$\Updownarrow$$

$$f(\mathbf{x}) = f_0 + \sum f_i(x_i) + \sum_{i<j} f_{ij}(x_i, x_j)$$

Hvafner et al. 2025

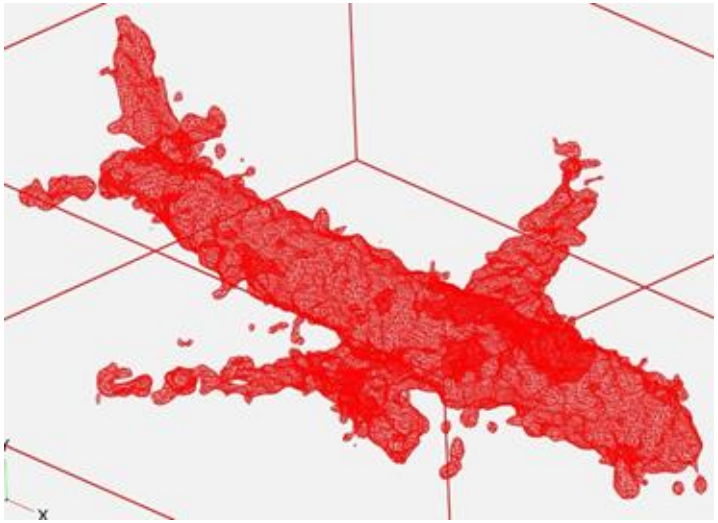
# High-dimensional Gaussian Processes

$$I(y_{\boldsymbol{X}}, f_{\boldsymbol{X}}) = \frac{1}{2} \log |\boldsymbol{I} + \sigma_{\varepsilon}^{-2} \mathbf{K}|,$$

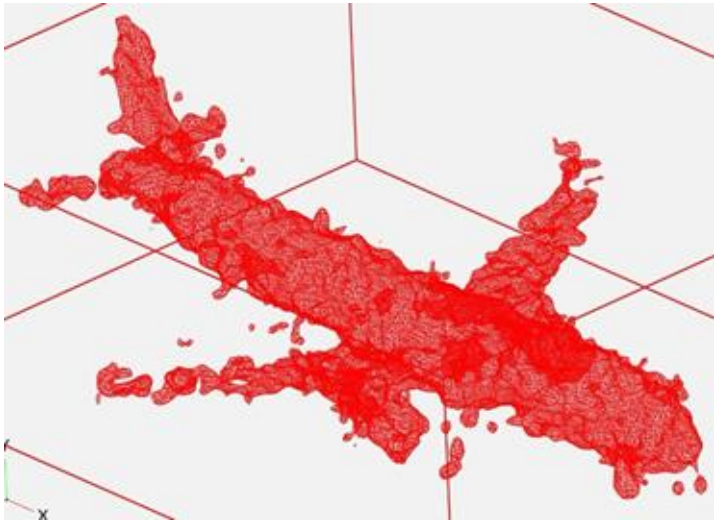


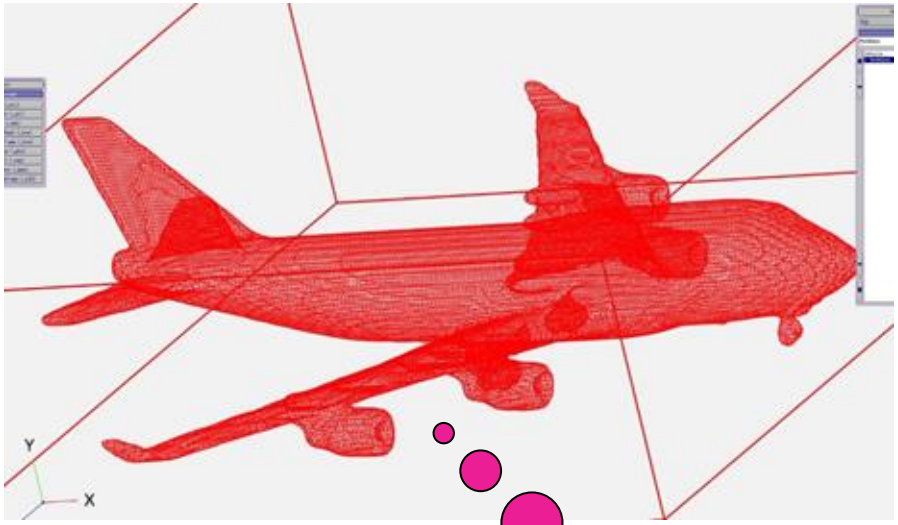
# Bayesian optimization in the age of GenAI

You know it when you see it



You know it when you see it

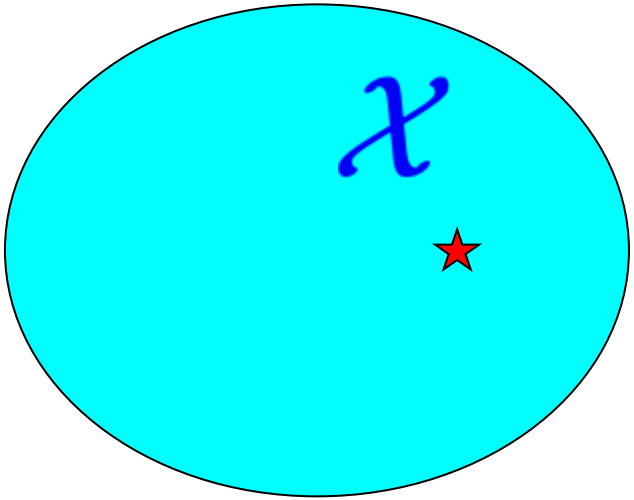




- Vanishingly small support
- Can't learn *is_plane* function

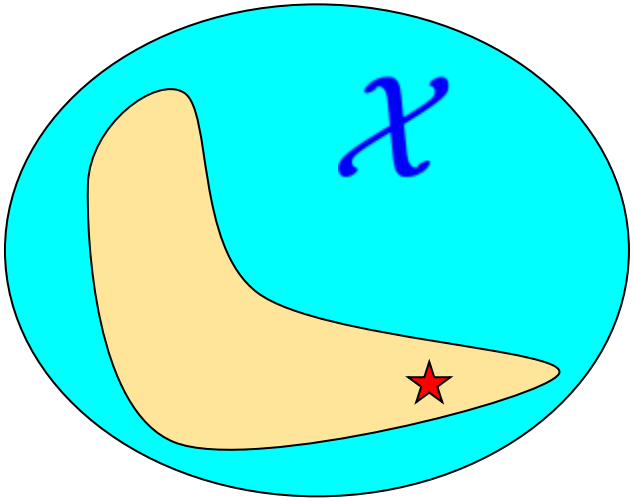
# You know it when you see it

# Standard BO




$$\max_{x \in \mathcal{X}} f(x)$$

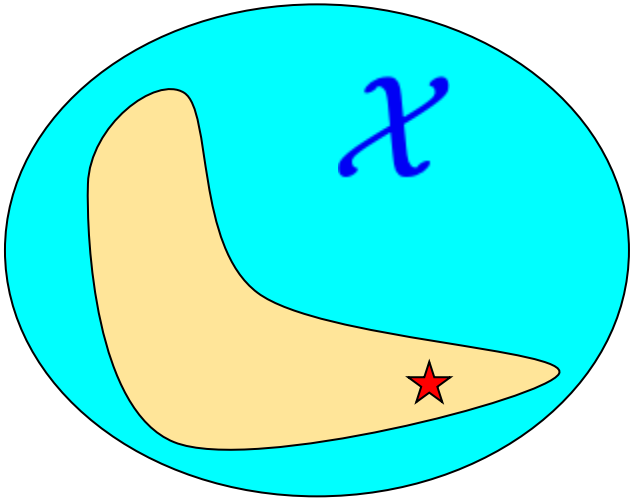
# Standard BO



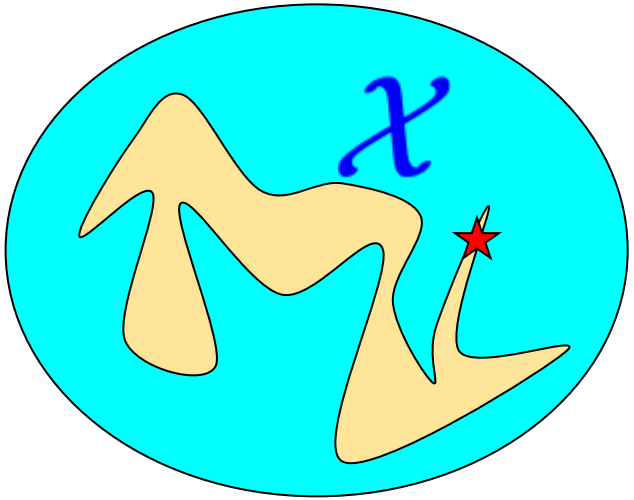

$$\max_{x \in \mathcal{X}} f(x)$$

$$: P(C(x) = 0) > t$$

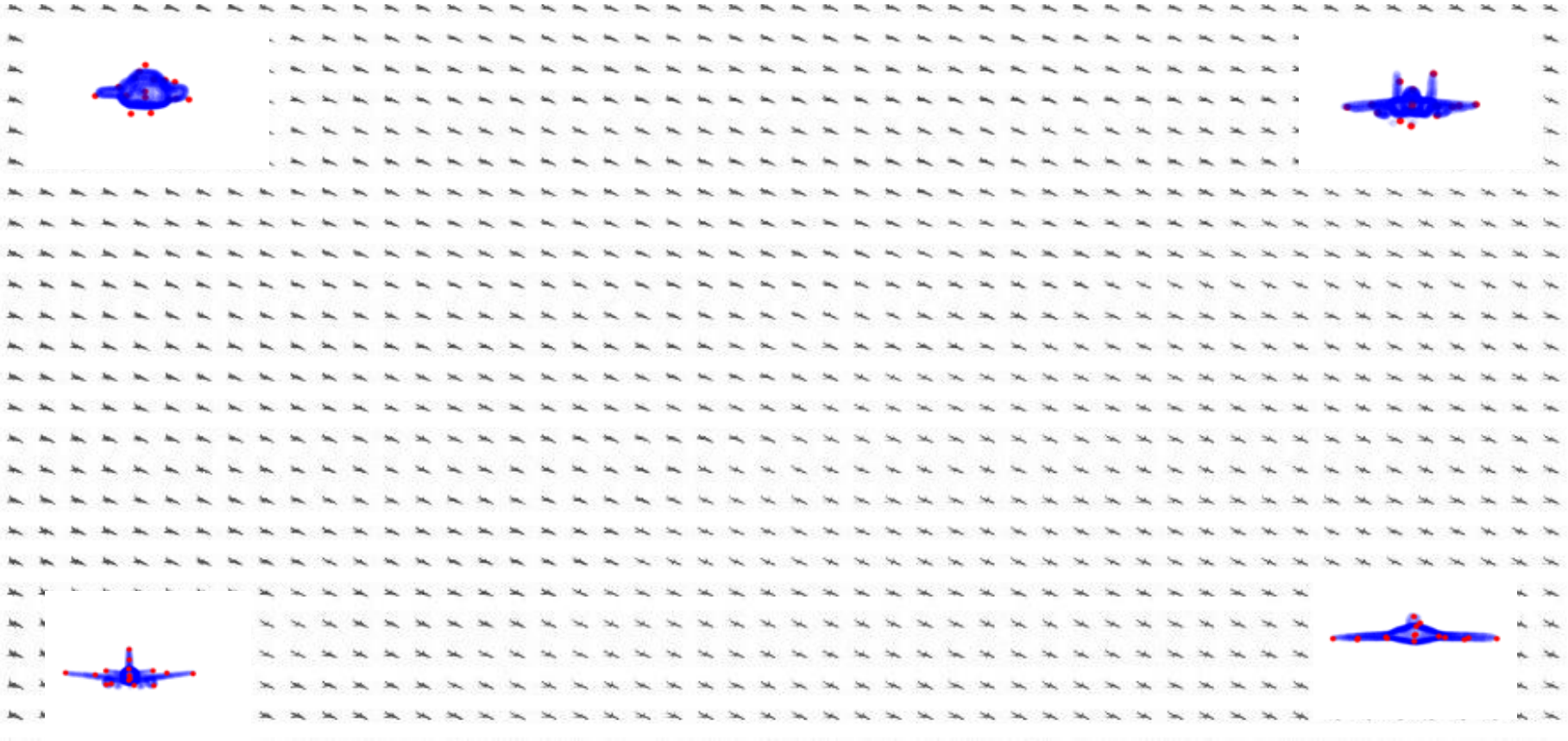
# Standard BO v.s. Structured problems




$$\max_{x \in \mathcal{X}} f(x)$$
$$: P(C(x) = 0) > t$$

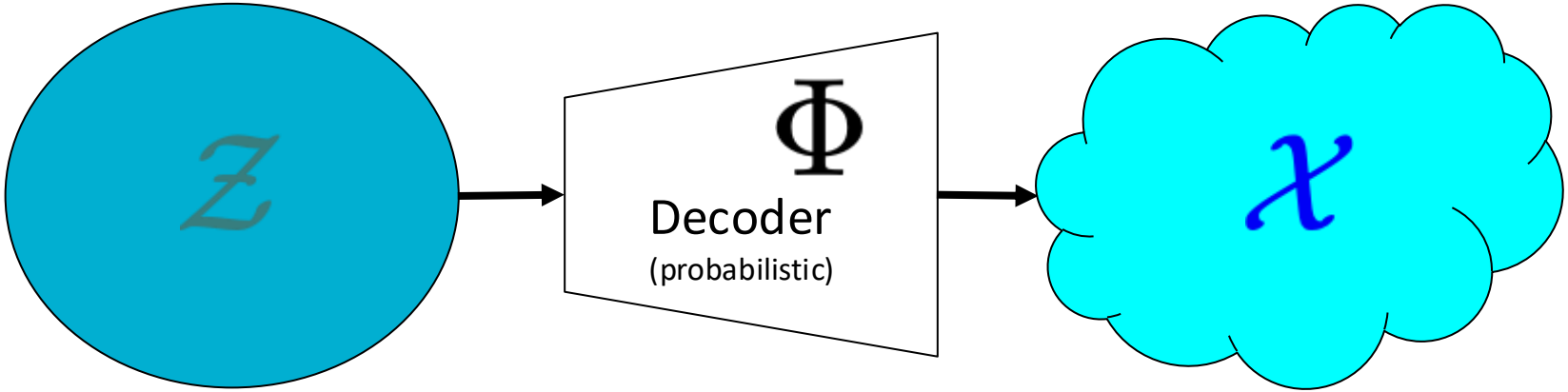



$$\max_{x \in \mathcal{X}} f(x)$$
$$: P(C(x) = 0) > t$$



# Generative AI is cool right?

# Priors over "**Highly Structured**" spaces



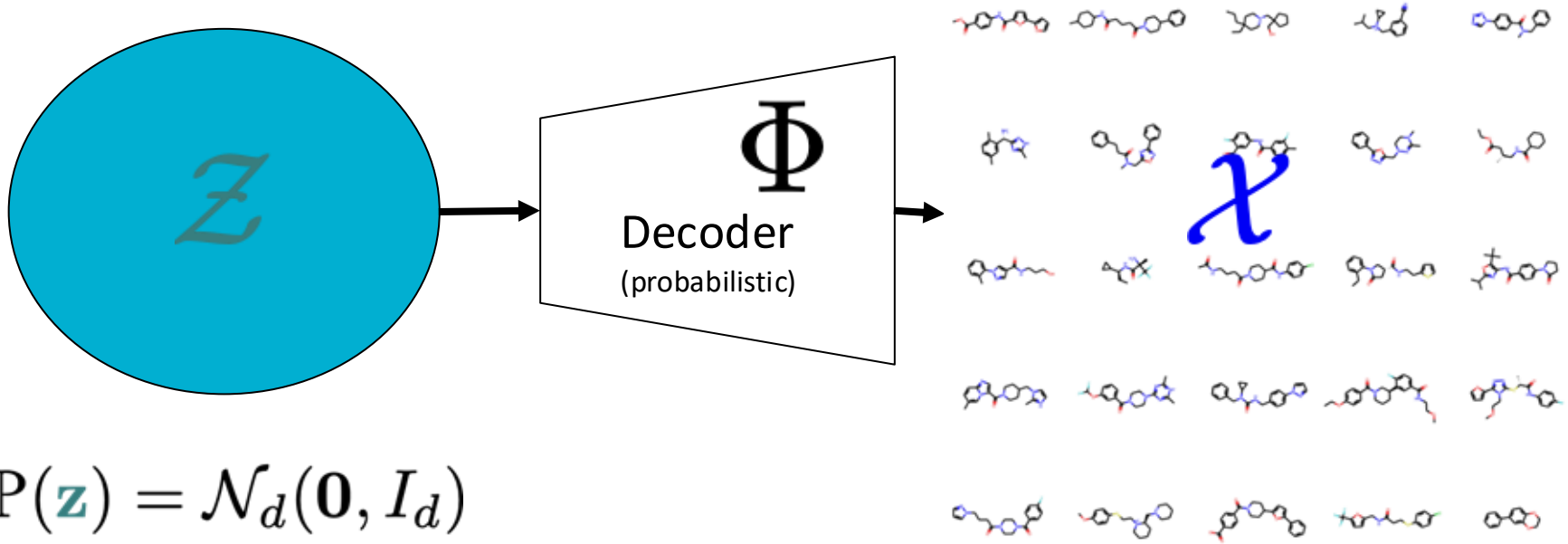

$$\mathbb{P}(\mathbf{z}) = \mathcal{N}_d(\mathbf{0}, I_d)$$

$$\mathbb{P}(\mathbf{x}) = ???$$

VAE / Diffusion / Normalising Flows e.t.c on unlabeled data

Richards et al. 2022

# Priors over "**Highly Structured**" molecule spaces




$$\mathbb{P}(\mathbf{z}) = \mathcal{N}_d(\mathbf{0}, I_d)$$

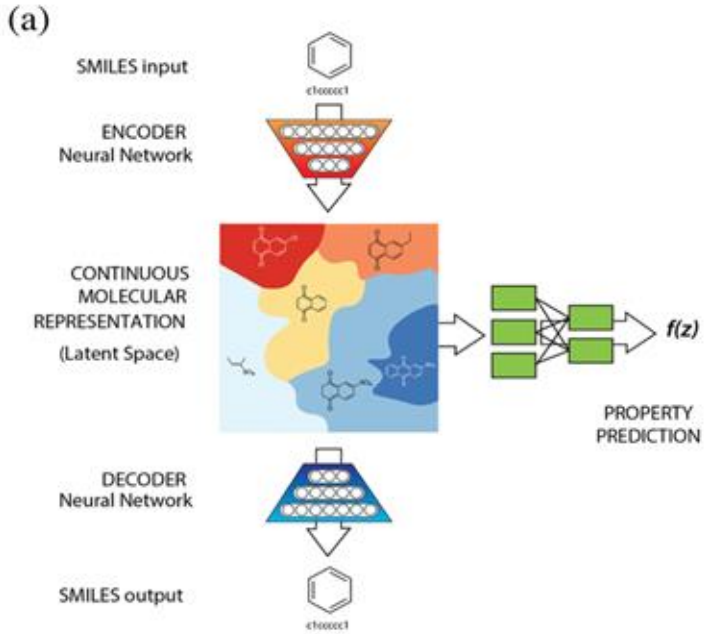
# How do we “condition” GenAI models?

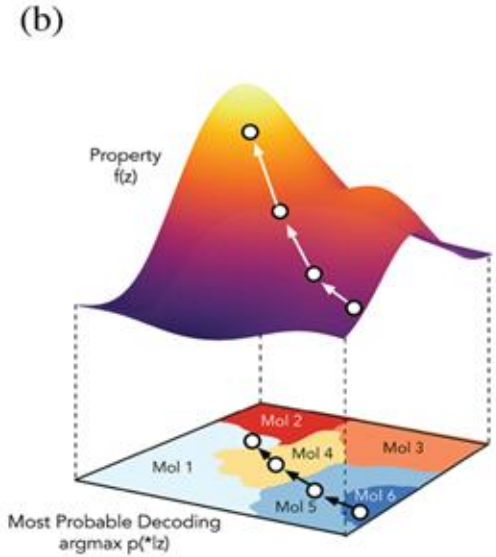
- Train conditional model (expensive, requires specificity)
- Fine-tuning on the fly (still expensive / fragile)
- Model-specific (e.g. guidance for flow-matching)
- **BO in the latent space (exploit locality)**

# LATENT SPACE BO

Search space = latent space

$$\mathcal{X} = \mathcal{Z}$$



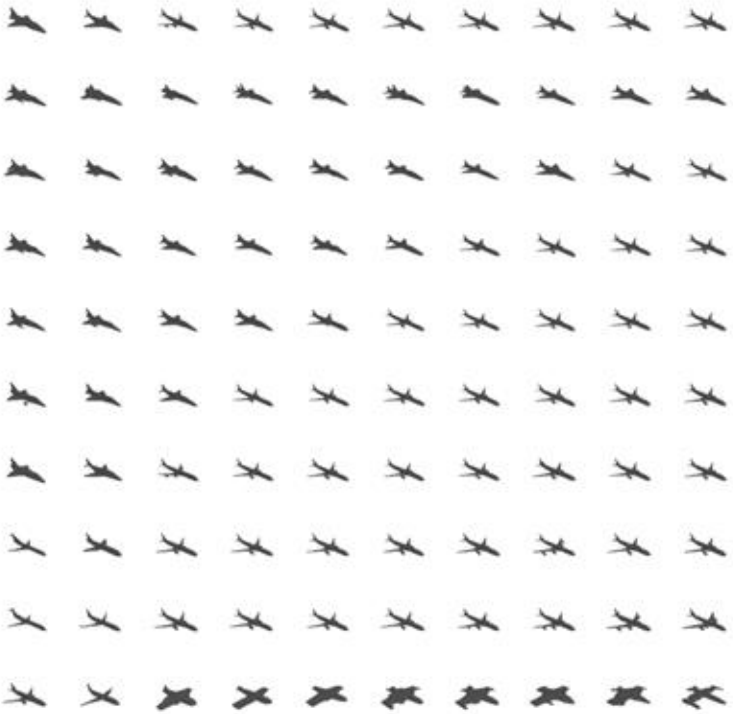



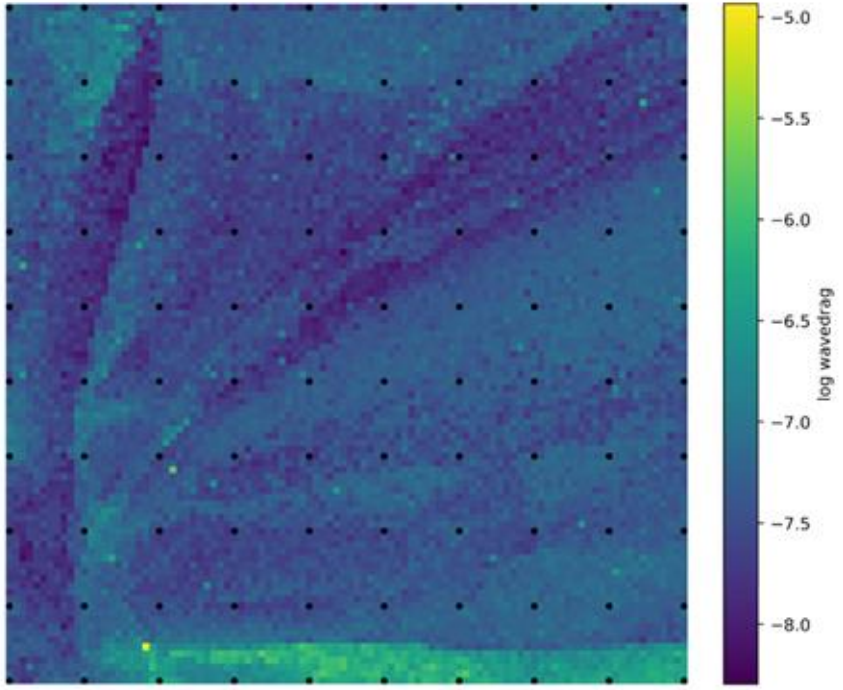

$$\tilde{f} : \mathcal{Z} \rightarrow \mathbb{R}$$

Continuous variable representation for:
- Interpolation
- Optimization
- Exploration

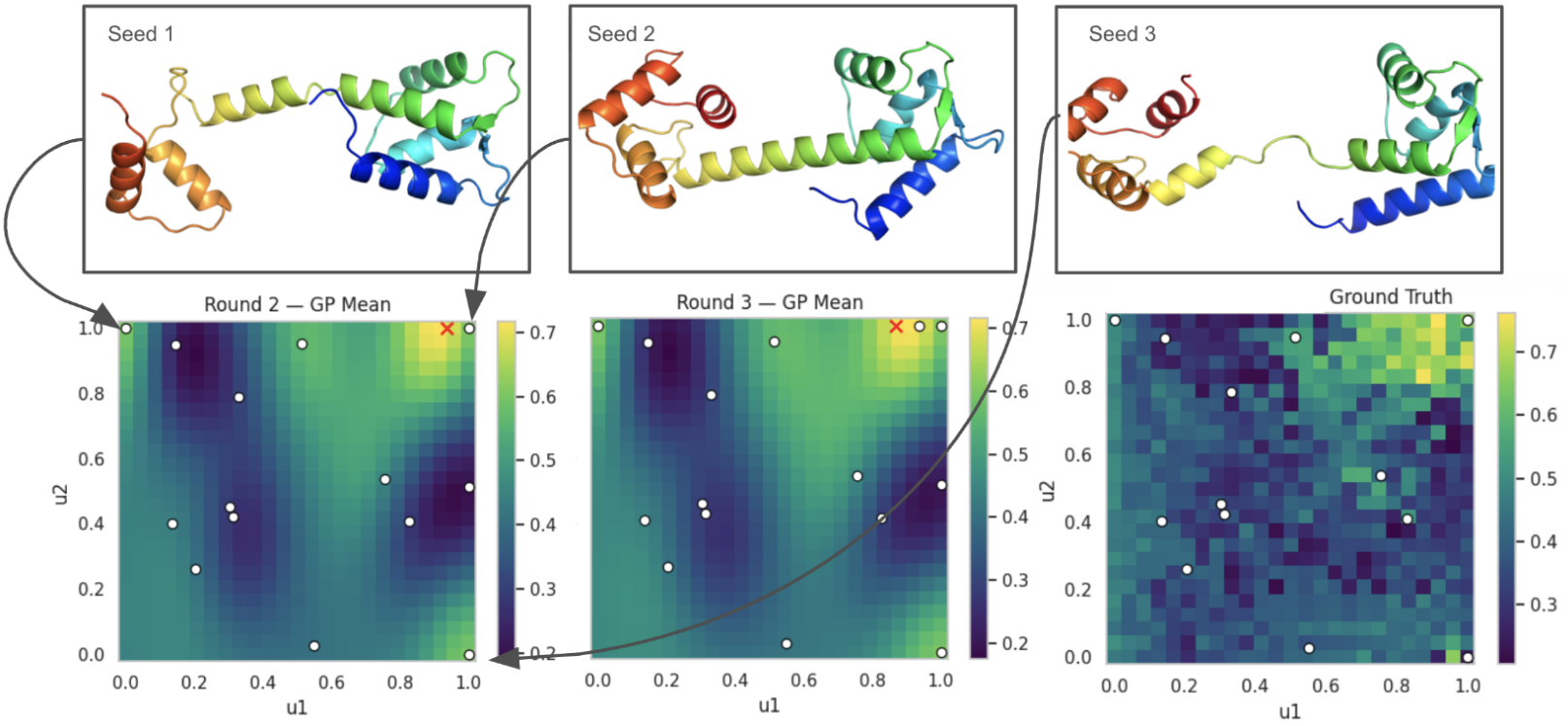
Gomez-Bombarelli et al. 2018

# Not just for VAEs

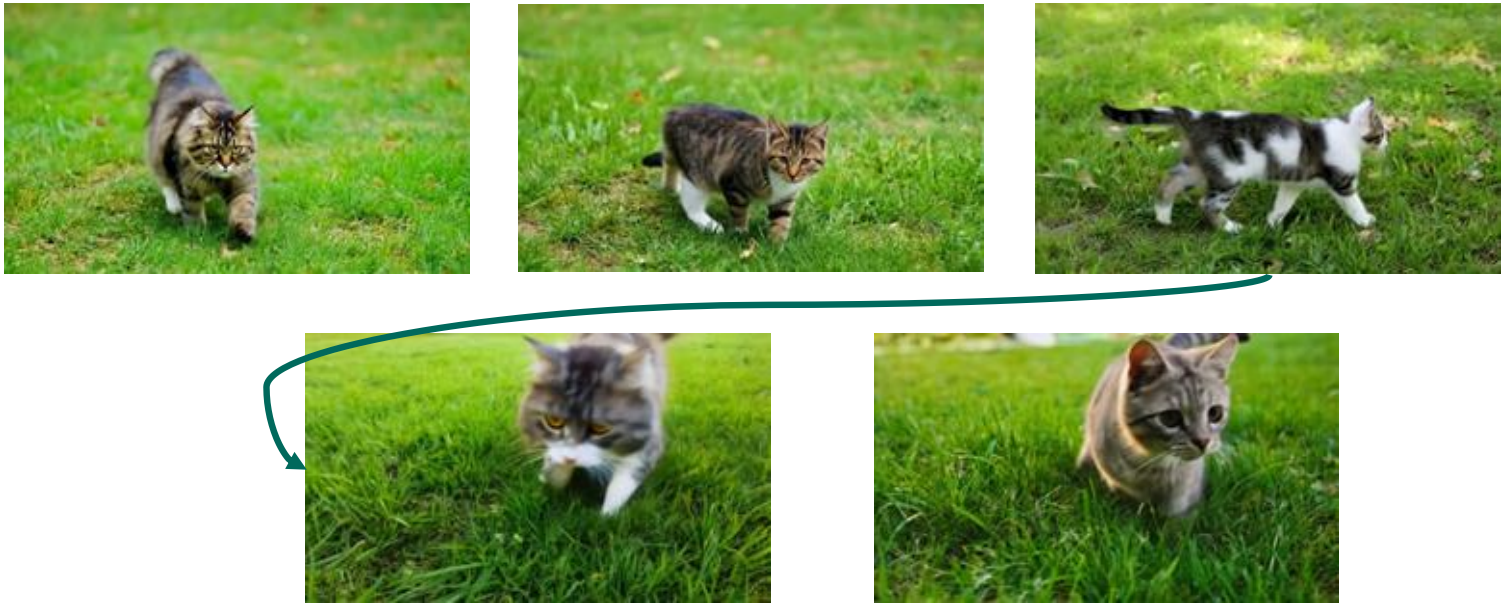




# Not just for VAEs

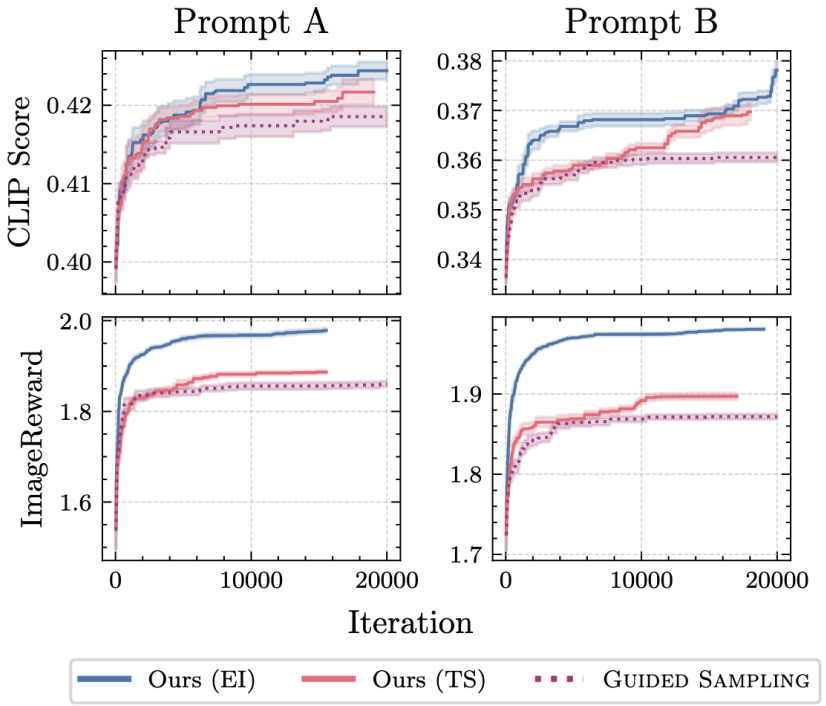


# Not just for VAEs



# Latent space BO is orthogonal to guidance


Prompt B: "Three cats and one dog sitting on the grass."



# BO inside stable diffusion

Ours (EI) | Guided Sampling

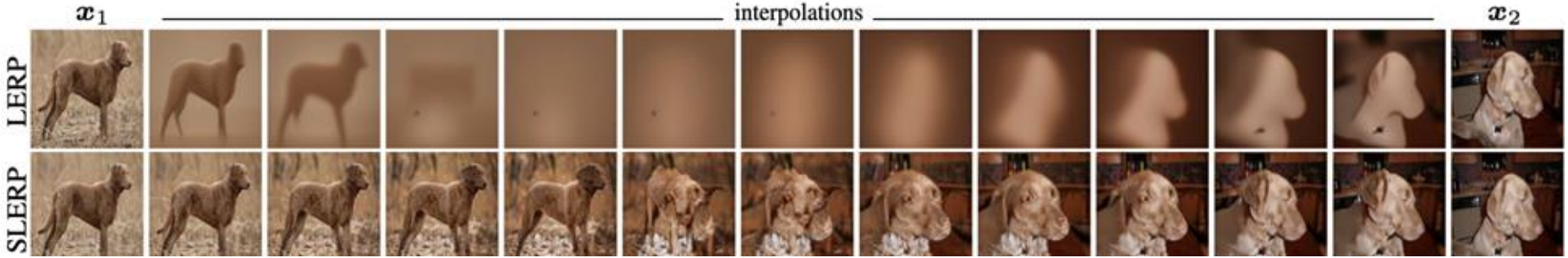

Prompt B: "Three cats and one dog sitting on the grass."

Ours (EI) | Guided Sampling

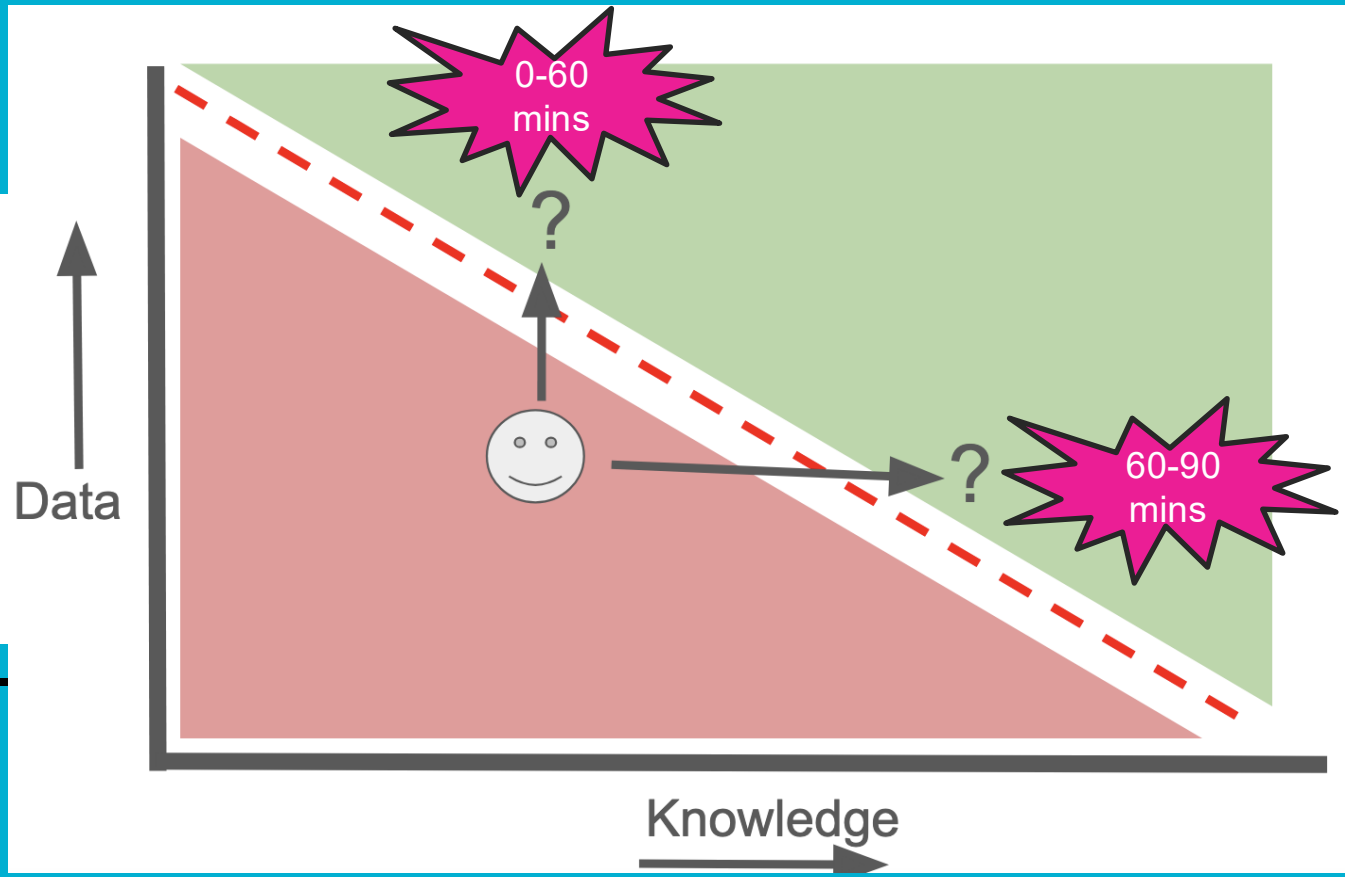

Prompt A: "Photo of an athlete cat explaining it's latest scandal at a press conference to journalists."

# Why is Latent space BO hard?

- High-dimensionality (N<<D)
- Not all latent encode nicely (typicality)
- Must make fast decisions

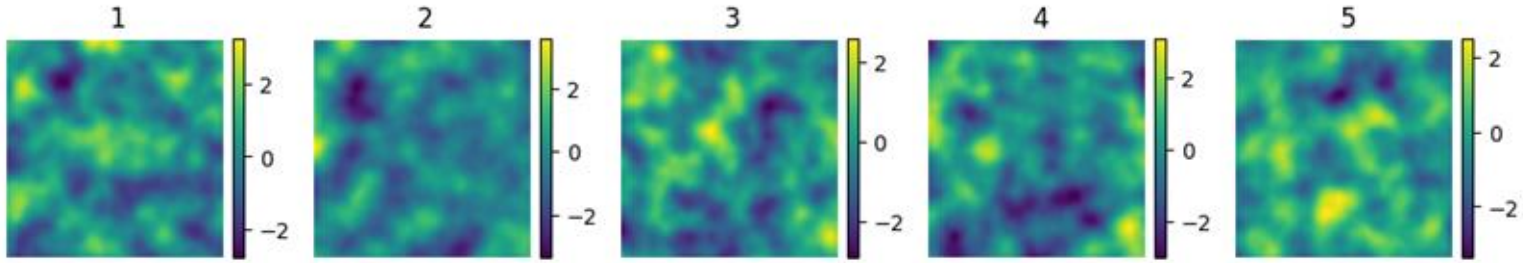


# Generative modelling for Gaussian processes


0-60
mins
?
Data
?
60-90
mins
Knowledge

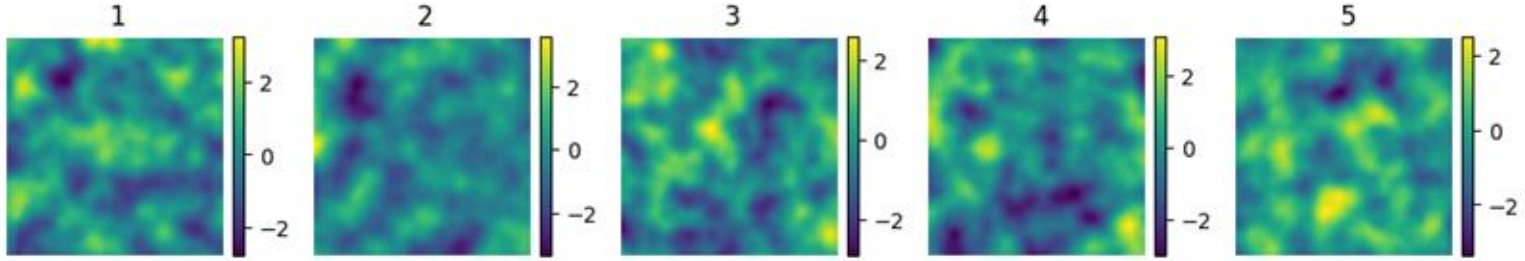
# Teaser

Matern

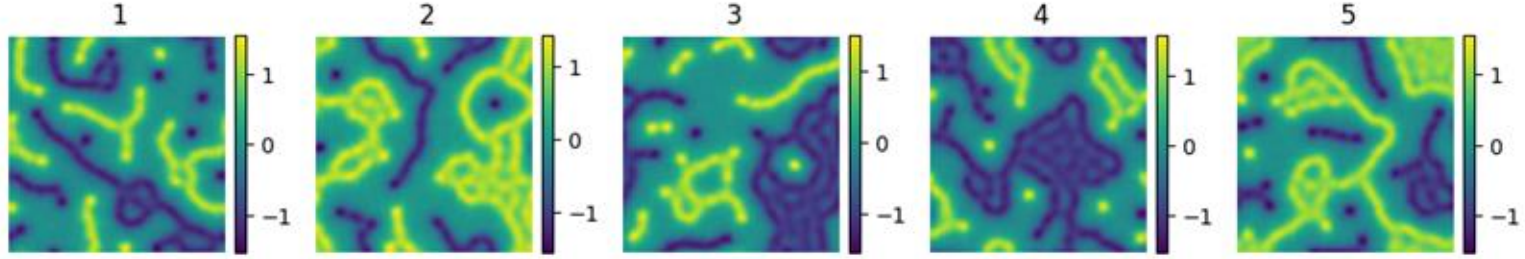


# Teaser

Matern




Matern + Weird ODE



# 1) Why are generative models so good now?

Variational Autoencoder

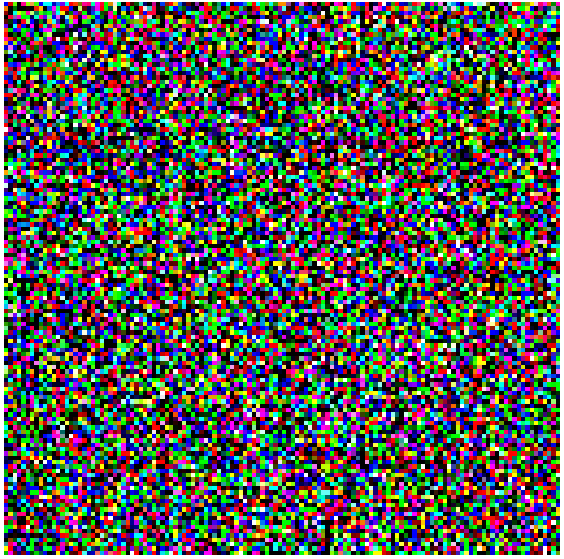
Diffusion

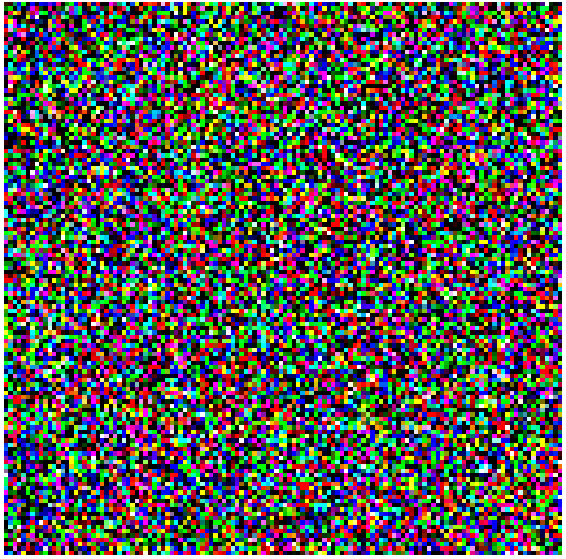
# 1) Why are generative models so good now?

Variational Autoencoder (1 step)

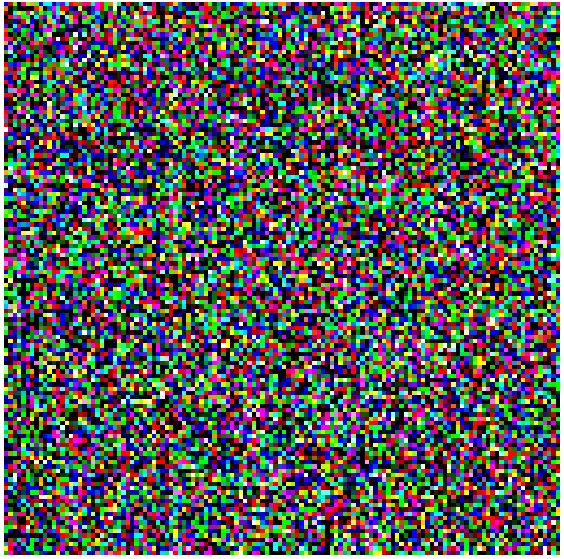
Diffusion (many steps)

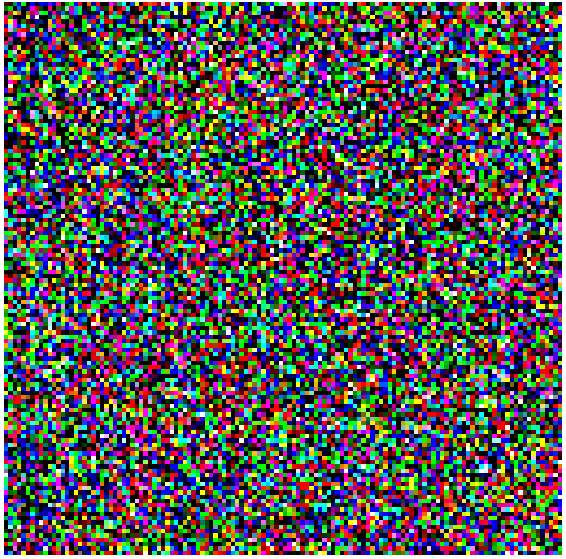
# Is this enough?

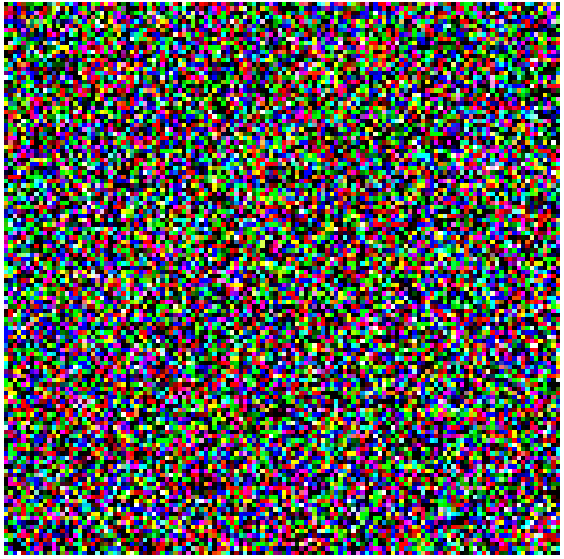




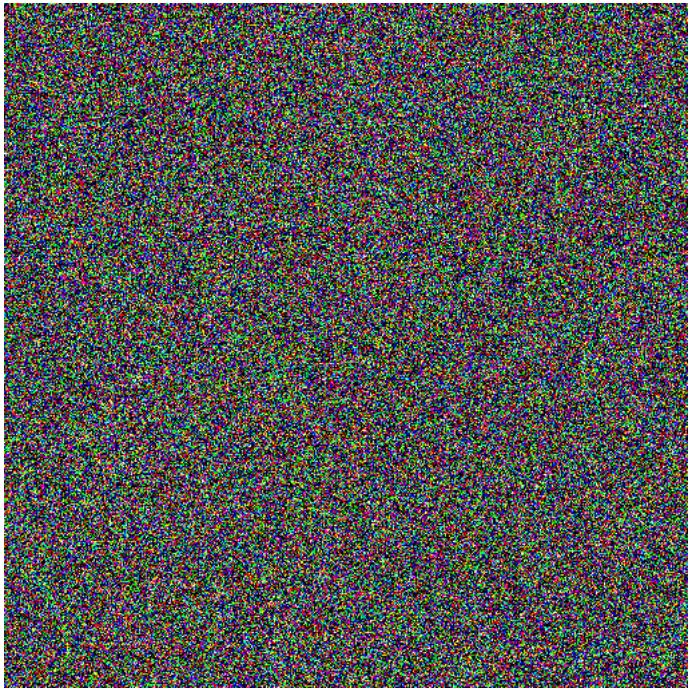
# Is this enough?

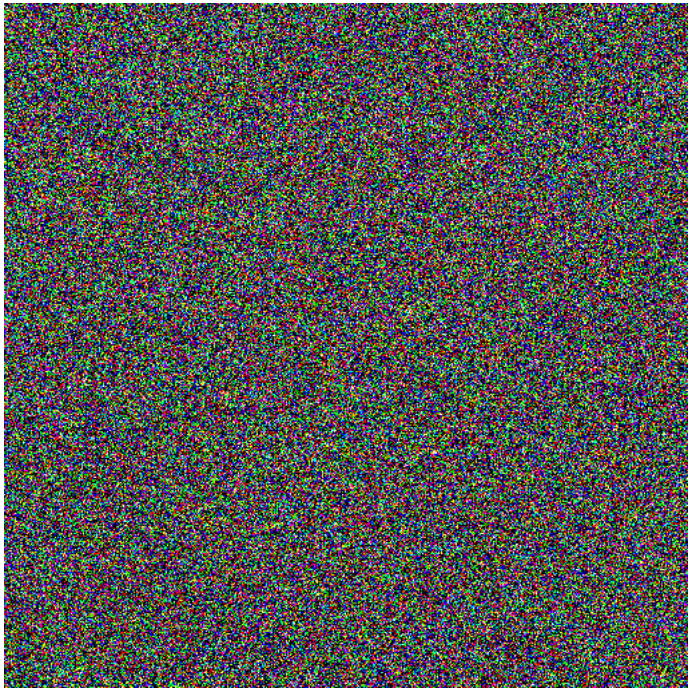






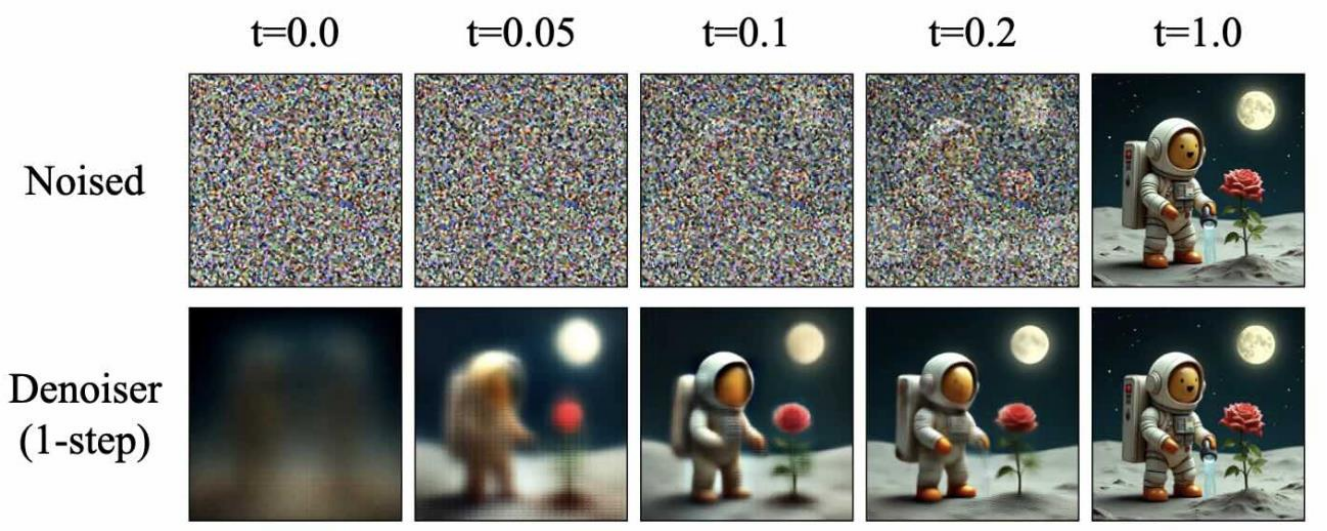
# We need controlled generation (“conditional”)





Chung et al 2023, Sabour et al 2026.

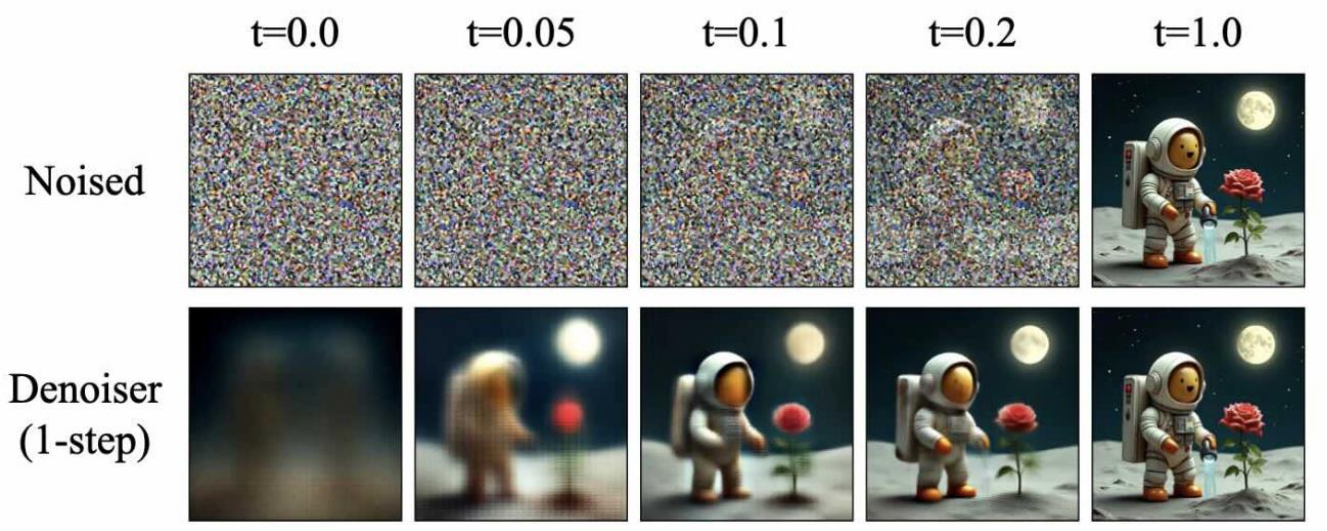
# 2) Why are generative models so good now ?

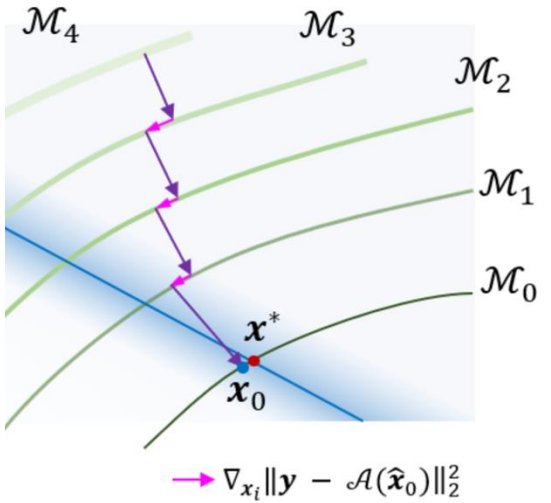



Can roughly look ahead

Chung et al 2023, Sabour et al 2026.

# 2) Why are generative models so good now ?



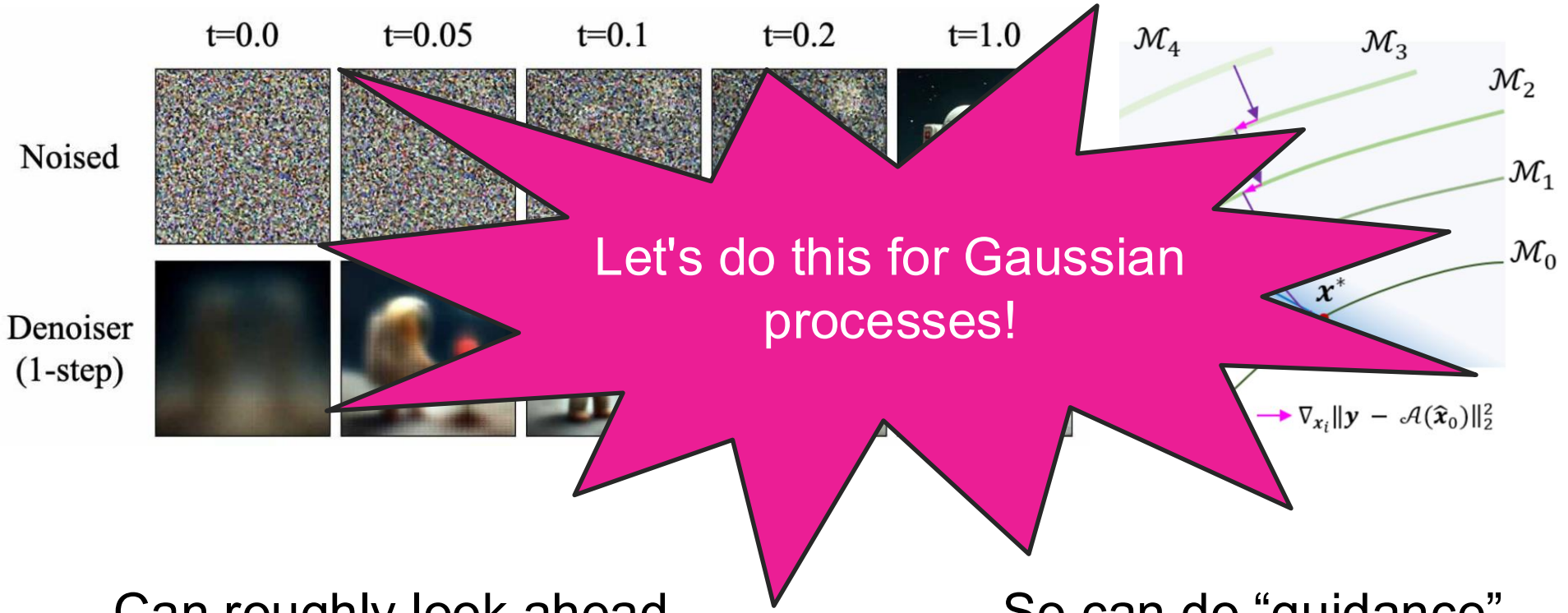




Can roughly look ahead

So can do “guidance”

Chung et al 2023, Sabour et al 2026.

# 2) Why are generative models so good now ?




Can roughly look ahead

So can do “guidance”

# Overview

1. An iterative (but over-the-top) way to sample from a GP

# Overview

1. An iterative (but over-the-top) way to sample from a GP

2. Guidance from genAI allows conditioning

# Overview

1. An iterative (but over-the-top) way to sample from a GP

2. Guidance from genAI allows conditioning

3. **Sampling from GP + any additional constraints via solving an ODE**

# Overview

1. An iterative (but over-the-top) way to sample from a GP

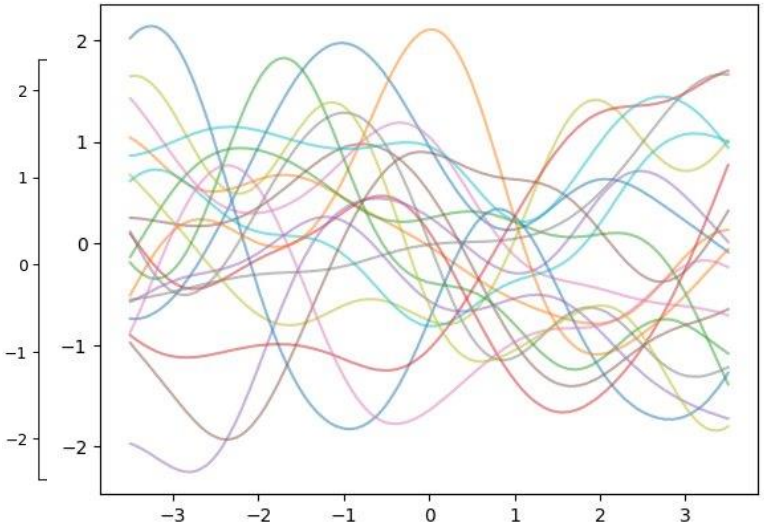
2. Guidance from genAI allows conditioning

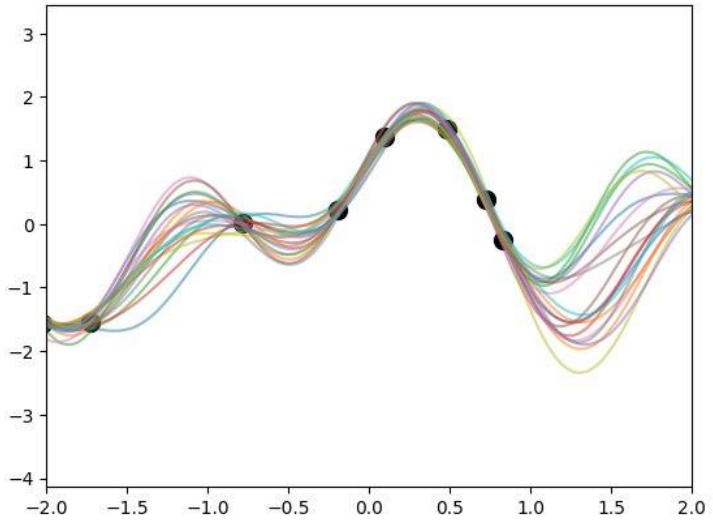
3. **Sampling from GP + any additional constraints via solving an ODE**

No neural networks required!

# A Gaussian Process Primer

$$\mathbf{f}_0 \sim \mathcal{N}(\mathbf{m}_*, \mathbf{K}_{**})$$



# Sampling a Gaussian Process (one step)

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\mathbf{f}_0 \leftarrow \mathbf{m}_* + \mathbf{K}_{**}^{1/2}\mathbf{z}$$

Borovitskiy et al 2023.

# Sampling a Gaussian Process (one step)

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\mathbf{f}_0 \leftarrow \mathbf{m}_* + \mathbf{K}_{**}^{1/2}\mathbf{z}$$

## Sampling a Gaussian Process in many steps (ours)

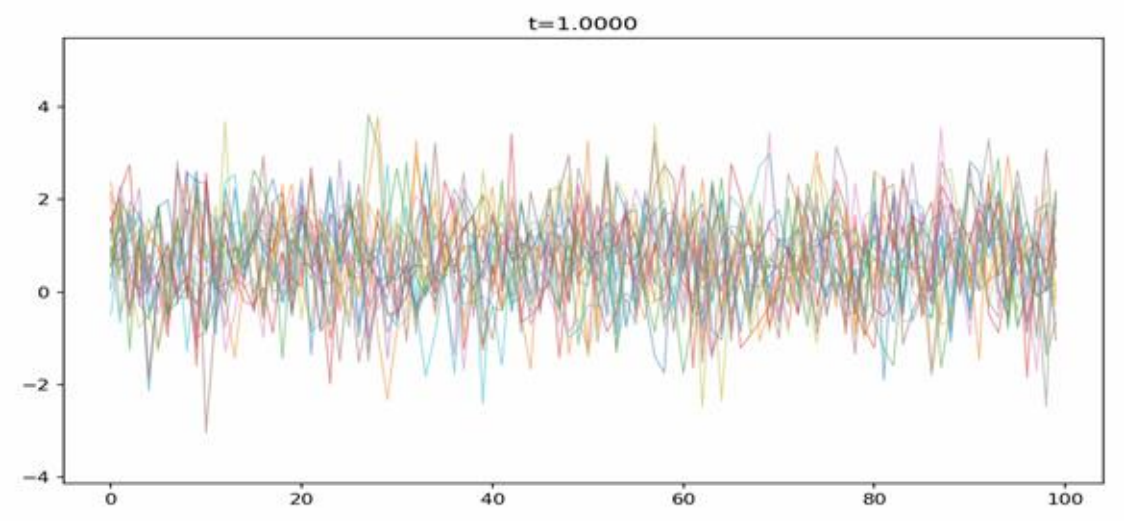
$$\boldsymbol{f}_t = \Phi(t, \boldsymbol{z}) := \mathbf{b}(t) + \mathbf{A}(t)^{1/2}\boldsymbol{z}, \quad t \in [0, 1]$$

---

$$\mathbf{A}(t) := \alpha^2(t)\mathbf{K}_{**} + (1 - \alpha^2(t))\mathbf{I}_m \qquad \mathbf{b}(t) := \alpha(t)\mathbf{m}_*$$

# Sampling a Gaussian Process in many steps (ours)

$$\boldsymbol{f}_t = \Phi(t, \boldsymbol{z}) := \mathbf{b}(t) + \mathbf{A}(t)^{1/2}\boldsymbol{z}, \quad t \in [0, 1]$$



# Sampling a Gaussian Process in many steps via an ODE

$$\frac{\mathrm{d}\boldsymbol{f}_t}{\mathrm{d}t} = \boldsymbol{v}(\boldsymbol{f}_t, t) := -\frac{1}{2}\beta(t)\mathbf{A}(t)^{-1}\mathbf{b}(t) - \frac{1}{2}\beta(t)\left(\mathbf{I}_m - \mathbf{A}(t)^{-1}\right)\boldsymbol{f}_t, \quad t \in [0, 1],$$

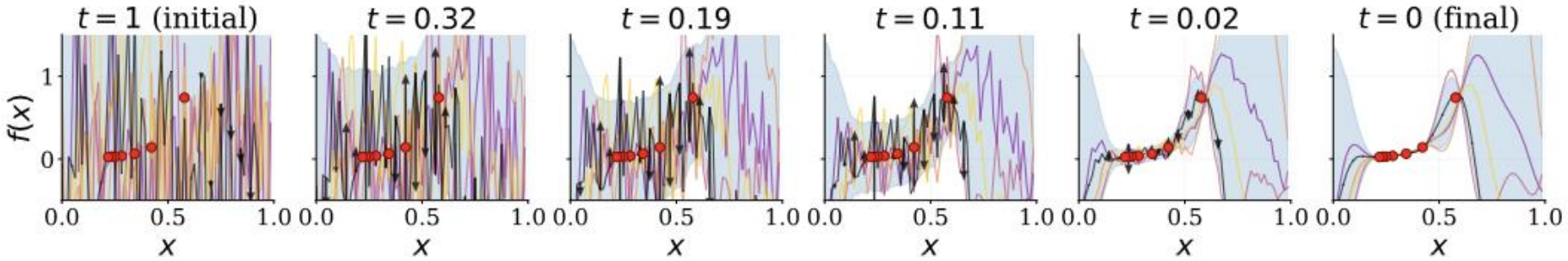
---

$$\beta(t) = -2\frac{\alpha'(t)}{\alpha(t)}$$

# Sampling a Gaussian Process in many steps via an ODE

$$\frac{\mathrm{d}\boldsymbol{f}_t}{\mathrm{d}t} = \boldsymbol{v}(\boldsymbol{f}_t, t) := -\frac{1}{2}\beta(t)\mathbf{A}(t)^{-1}\mathbf{b}(t) - \frac{1}{2}\beta(t)\left(\mathbf{I}_m - \mathbf{A}(t)^{-1}\right)\boldsymbol{f}_t, \quad t \in [0, 1],$$

$$f_1 \sim \mathcal{N}(0, I) \longrightarrow f_0 \sim \mathcal{N}(0, K)$$



# This is a stupid thing to do

- Same order complexity in N …
- More expensive than standard sampling ….
- ODE is an additional source of error …

**But……..**

# Perhaps not a stupid thing to do

- We have an iterative generation process
- We have a 1-step denoiser (via flow map)
- Let's apply principles of guidance (training-free guidance)

# FLOWGP: Sampling GPs under non-linear and non-Gaussian conditioning

$$p(\boldsymbol{f}_0 \mid \mathcal{D}, \mathcal{C}) \propto \underbrace{p(\boldsymbol{f}_0)\, p(\mathcal{D} \mid \boldsymbol{f}_0)}_{\propto\, p(\boldsymbol{f}_0 \mid \mathcal{D})}\, p(\mathcal{C} \mid \boldsymbol{f}_0),$$

- $\mathcal{D}$ denotes information that admits a linear closed-form Gaussian update
- $\mathcal{C}$ denotes non-linear / non-Gaussian contributions

# FLOWGP: Sampling GPs under non-linear and non-Gaussian conditioning

$$p(\boldsymbol{f}_0 \mid \mathcal{D}, \mathcal{C}) \propto \underbrace{p(\boldsymbol{f}_0)\, p(\mathcal{D} \mid \boldsymbol{f}_0)}_{\propto\, p(\boldsymbol{f}_0 \mid \mathcal{D})}\, p(\mathcal{C} \mid \boldsymbol{f}_0),$$

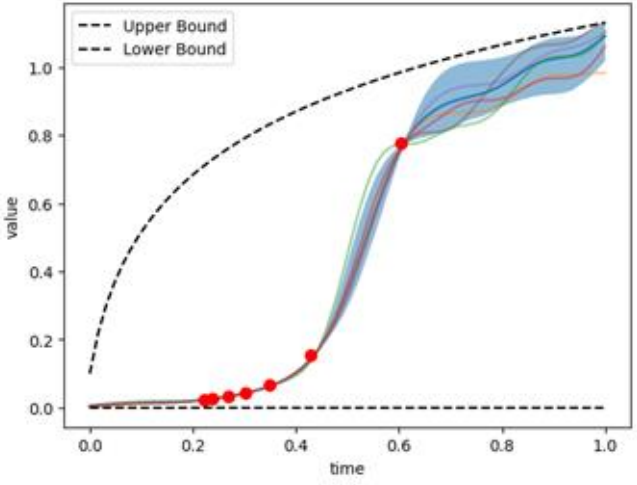
- $\mathcal{D}$ denotes information that admits a linear closed-form Gaussian update
- $\mathcal{C}$ denotes non-linear / non-Gaussian contributions

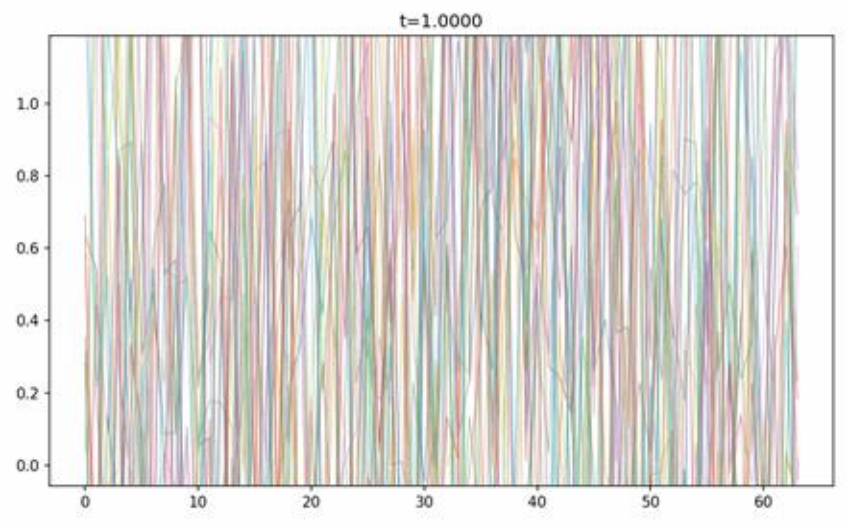
$$\boxed{\frac{\mathrm{d}\boldsymbol{f}_t}{\mathrm{d}t} = \boldsymbol{v}(\boldsymbol{f}_t, t \mid \mathcal{D}) - \frac{1}{2}\beta(t)\nabla_{\boldsymbol{f}_t} \log p(\mathcal{C} \mid \boldsymbol{f}_t, \mathcal{D})}$$

# Example conditions

$$\mathcal{C} = \{\text{is } f_0 \text{ monotonic and bounded}\}$$

$$\mathcal{C} = \{\text{does } f_0 \text{ satisfy an ODE}\}$$



# Accurate approximation of the conditional score

$$p_t(\mathcal{C}|f_t) = \int p_t(\mathcal{C}|f_0)p(f_0|f_t)df_0$$

likelihood

tractable!!!

## Accurate approximation of the conditional score (nice for us)

$$p_t(\mathcal{C}|f_t) = \int p_t(\mathcal{C}|f_0)p(f_0|f_t)df_0$$

likelihood

tractable!!!

Unlike diffusion/flow-matching, our joint distribution is tractable

# Sampling from a GP (whitened)

$$f \leftarrow L^T f$$

"The most efficient thing to do with a computer is nothing"

# Sampling from a GP (whitened)

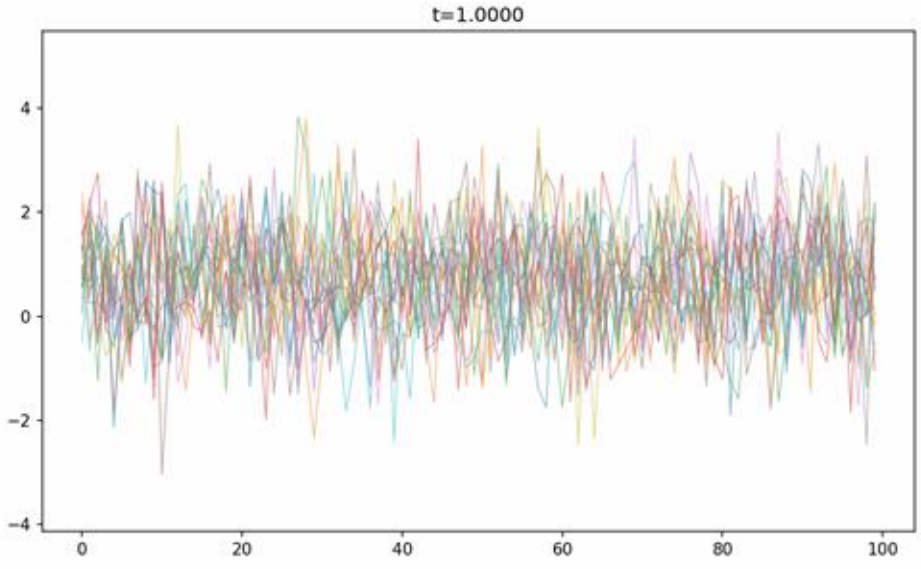
$$f \leftarrow L^T f$$

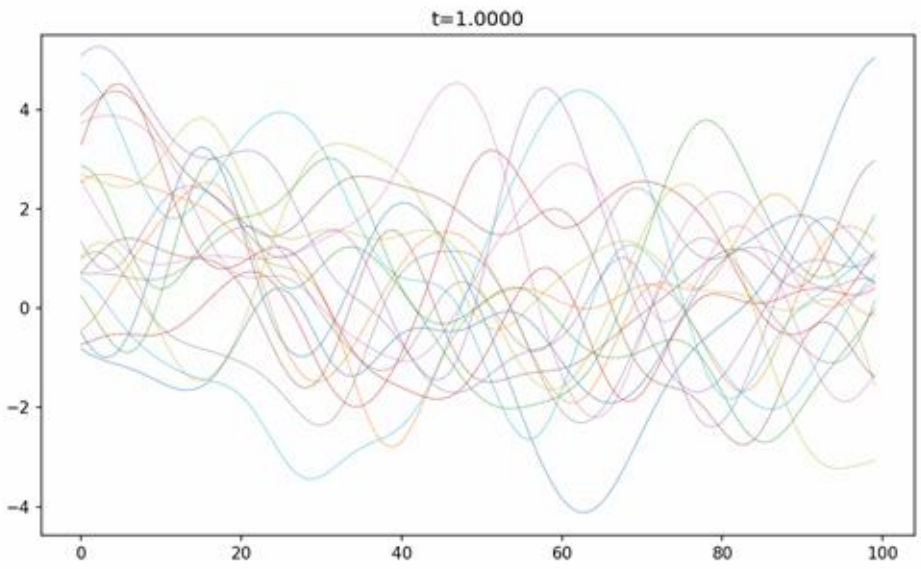
$$\frac{df_t}{dt} = \cancel{v_{\text{uncond}}(f_t, t)} + \frac{1}{2}\beta(t)\nabla_{f_t} \log p_t(\mathcal{C}|f_t)$$
$$= 0$$

"The most efficient thing to do with a computer is nothing"

# Sampling from a GP (whitened)

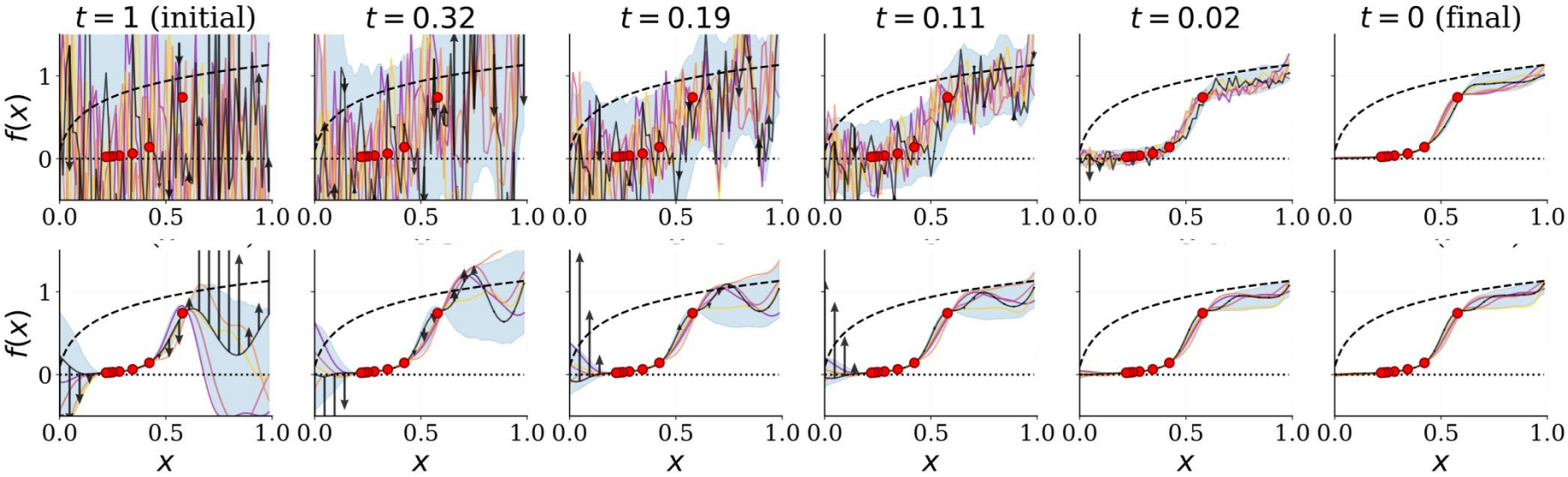
$$f \leftarrow L^T f$$



# Sampling from a GP (whitened)

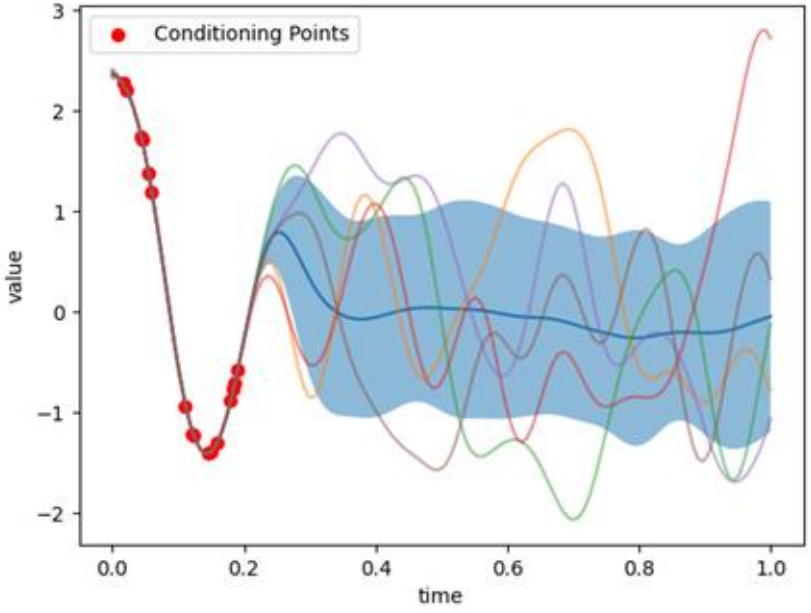
$$f \leftarrow L^T f$$



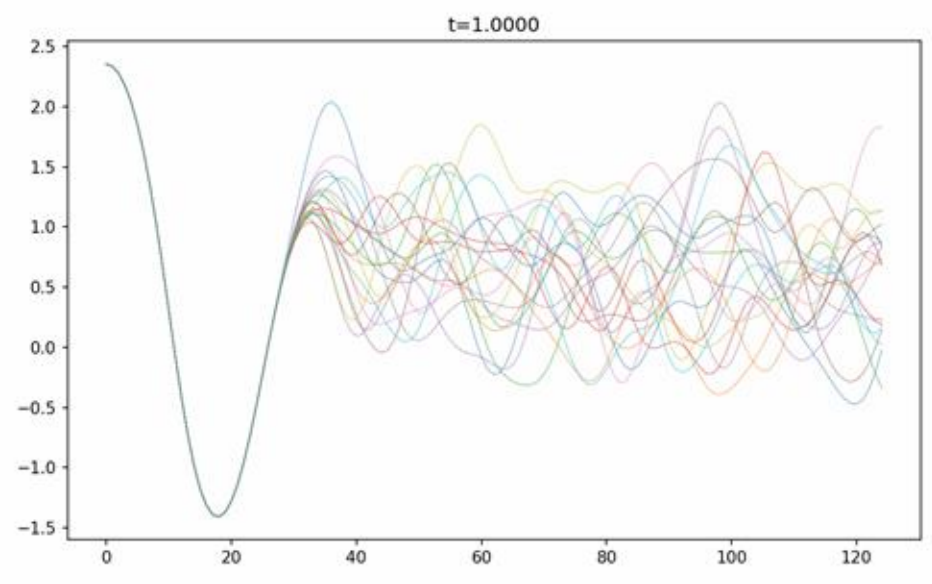
# Some examples...

# Nonlinear damped pendulum

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

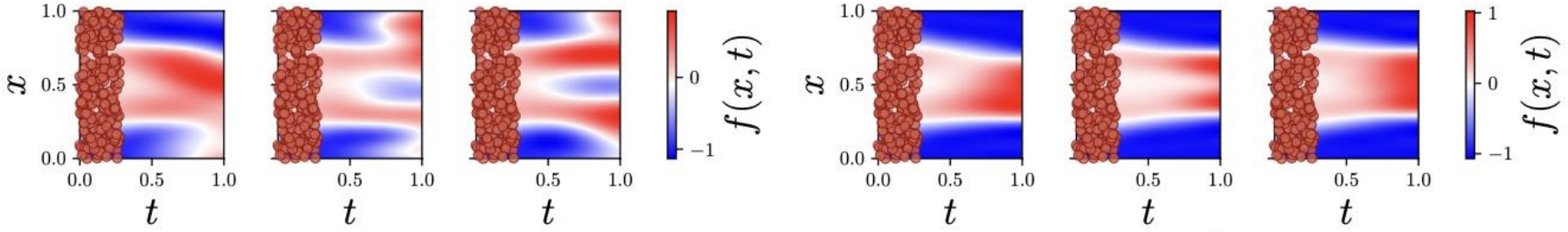



RBF GP

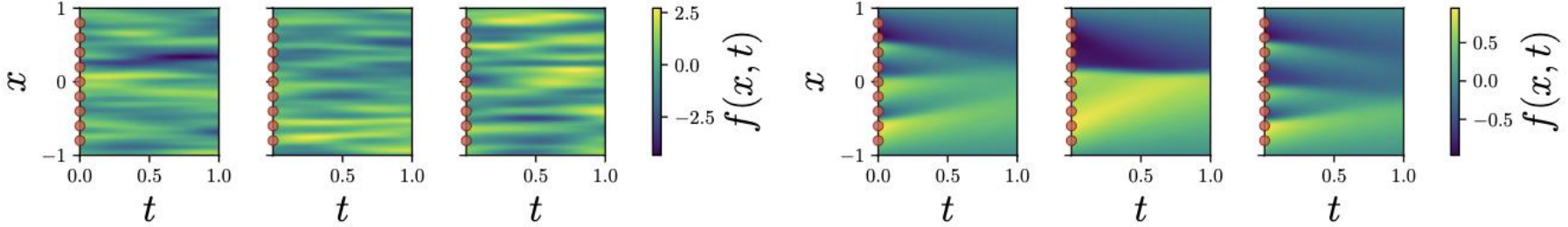



RBF GP +

# 2d physics



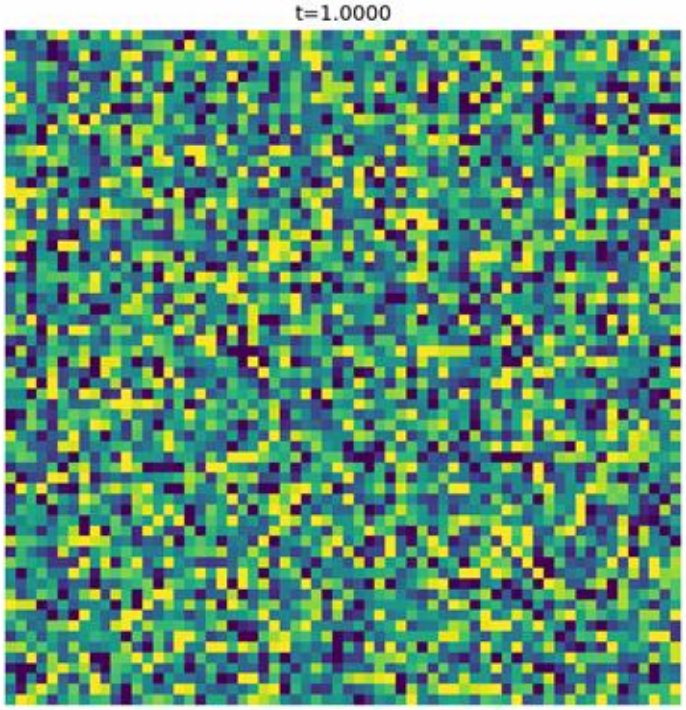

**(b)** FLOWGP enforcing an Allen–Cahn equation constraint: $\frac{\partial f}{\partial t} = \varepsilon \frac{\partial^2 f}{\partial x^2} + 5f - 5f^3$



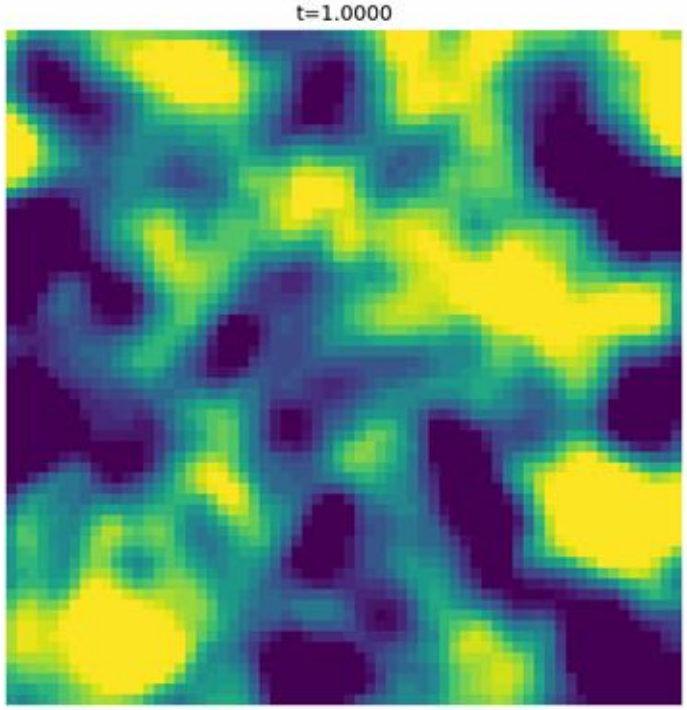

**(c)** FLOWGP enforcing a Burgers' equation constraint: $\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = \nu \frac{\partial^2 f}{\partial x^2}$

## Stationary Allen-Cahn

$$af - bf^3 - c\Delta f = 0$$




Non-whitened

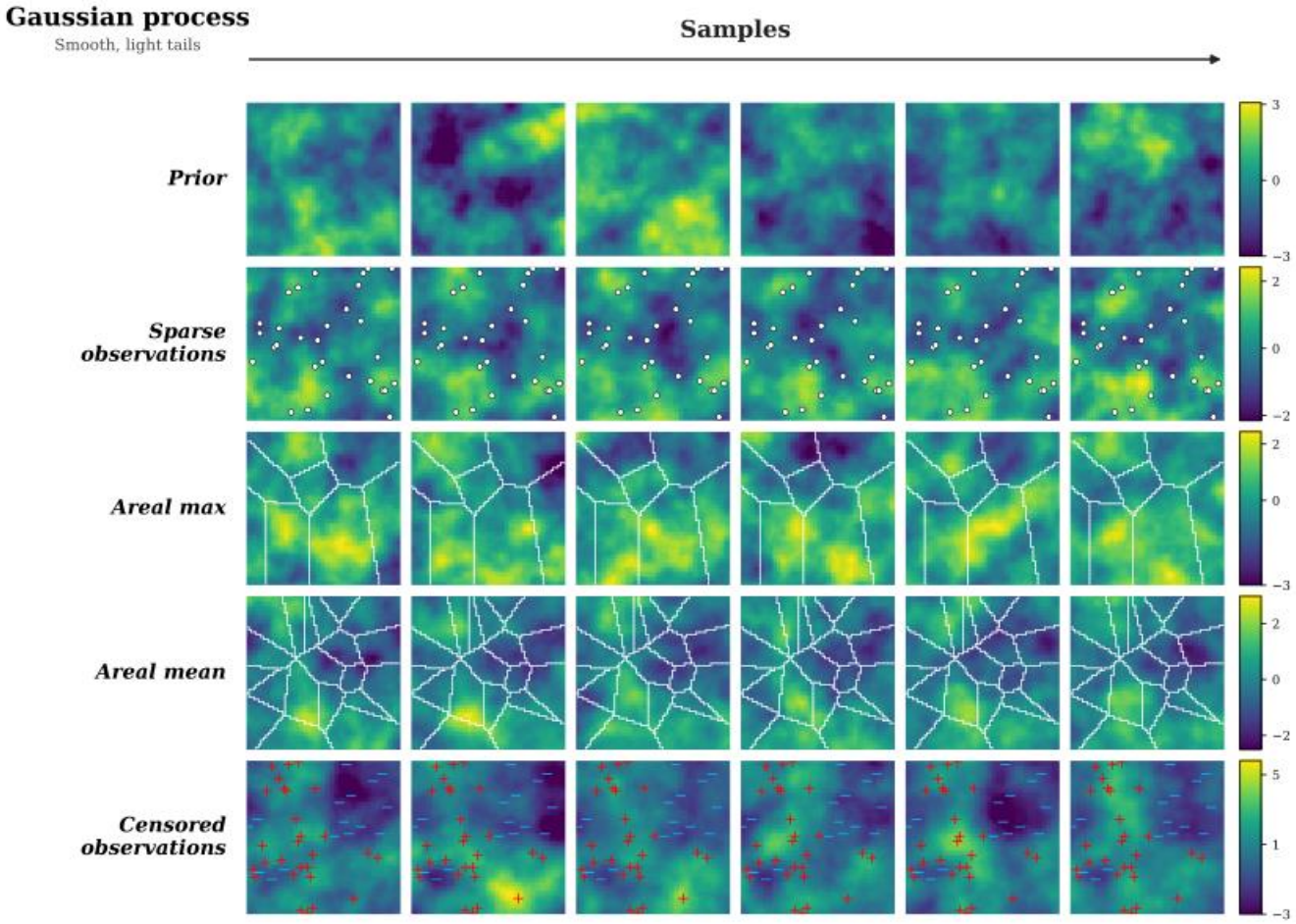



whitened


# Any questions?

henry.moss@lancaster.ac.uk for collaborations / PhD / Postdoc opportunities

# Gaussian Process in Spatial Stats



# Cauchy Convolution Process in Spatial Stats