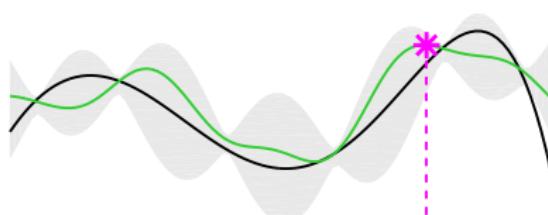


# Active Bayesian Design of Experiments via Posterior Sampling

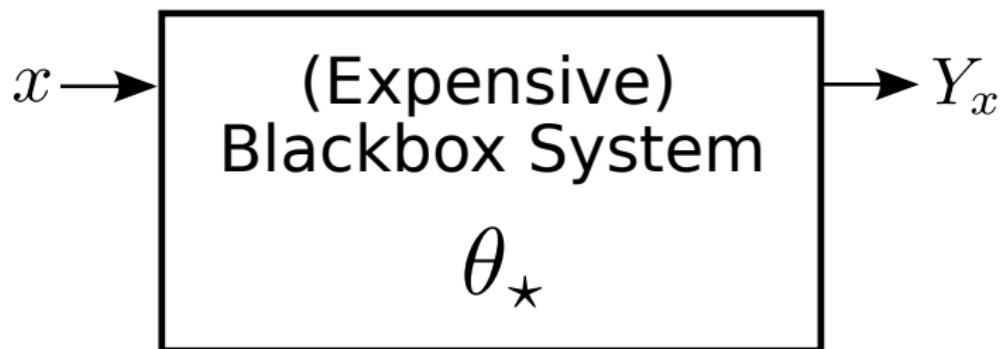


**Kirthevasan Kandasamy**  
Carnegie Mellon University

June 6, 2018  
Machine Learning in Science & Engineering Conference  
Pittsburgh, PA

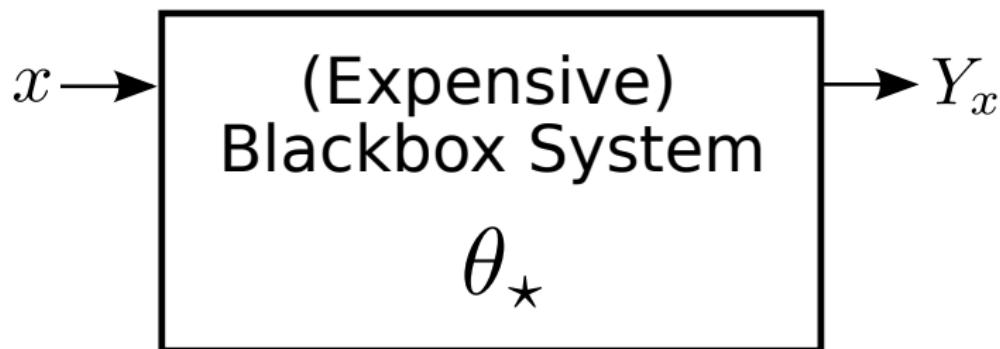
slides: [www.cs.cmu.edu/~kkandasa](http://www.cs.cmu.edu/~kkandasa)

## Design of Experiments



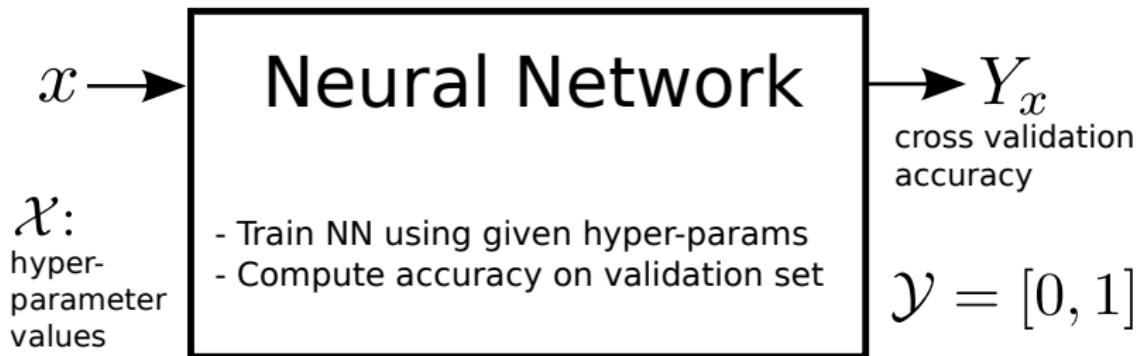
- ▶ Choose an experiment  $x \in \mathcal{X}$ .
- ▶ Obtain the result (observation)  $Y_x \sim \mathbb{P}(y|x, \theta_\star)$ .
- ▶  $\theta_\star$  (unknown) completely specifies the system.

## Design of Experiments



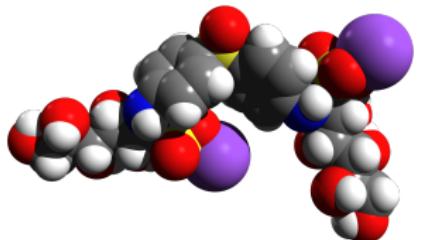
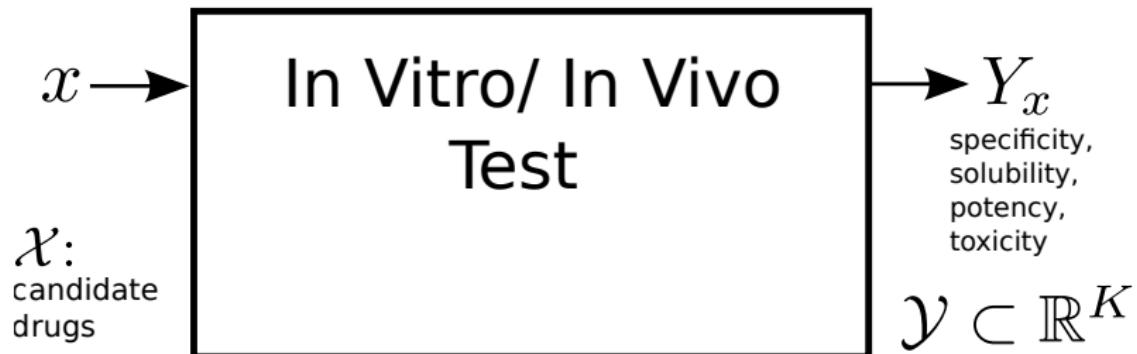
- ▶ Choose an experiment  $x \in \mathcal{X}$ .
- ▶ Obtain the result (observation)  $Y_x \sim \mathbb{P}(y|x, \theta_*)$ .  
 $\theta_*$  (unknown) completely specifies the system.
- ▶ Repeat in an adaptive sequence to collect data  
 $D_n = \{(x_t, Y_{x_t})\}_{t=1}^n$ .
- ▶ Typically some “goal” in mind.

# Black-box Optimisation: Model Selection



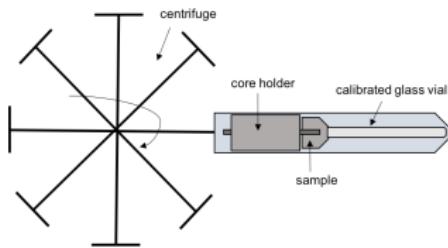
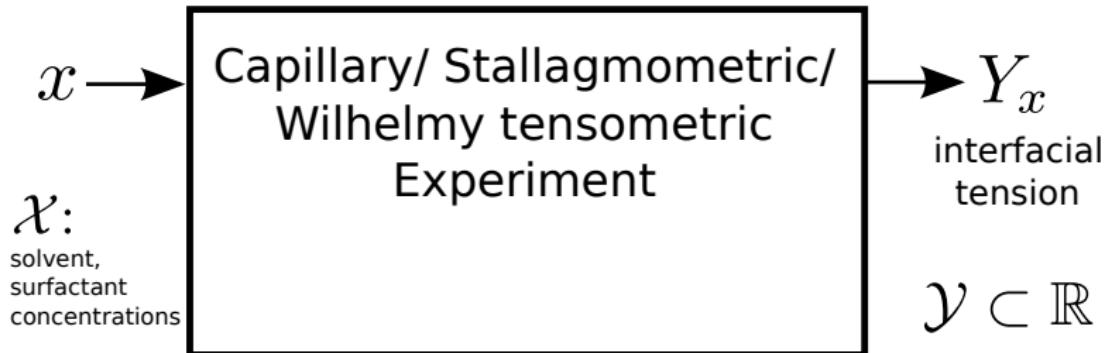
**Goal:** Find hyper-parameters with highest CV accuracy.

# Multi-objective Optimisation: Drug Discovery



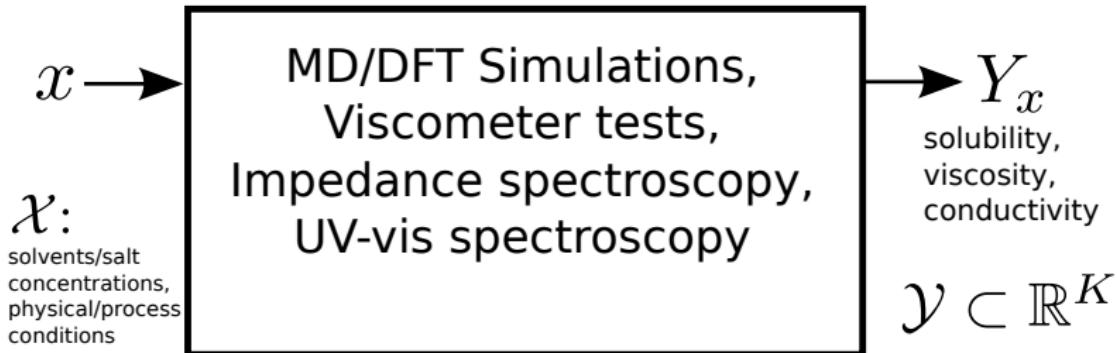
**Goal:** Find drug with “good value” on all objectives.

# Active Learning: Materials Science



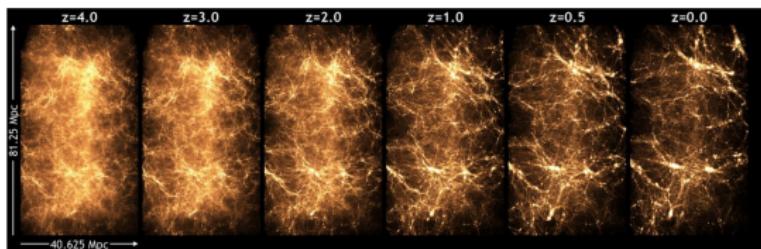
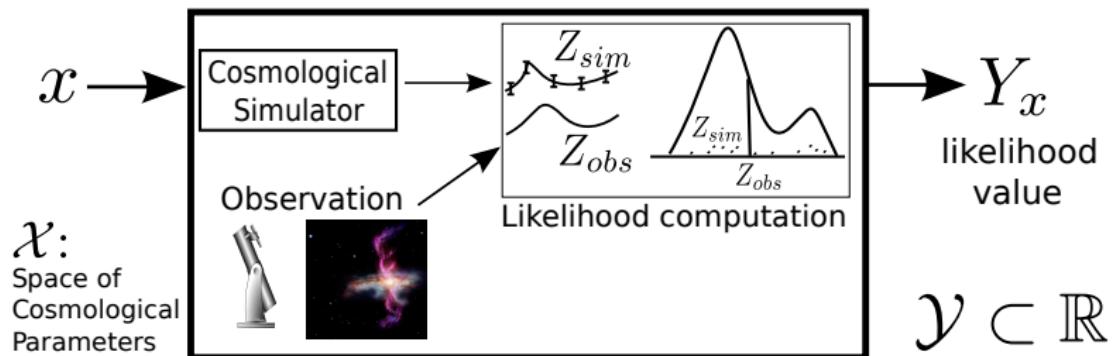
**Goal:** Estimate relation between solution and interfacial tension.

# Multiple Goals: Materials Science



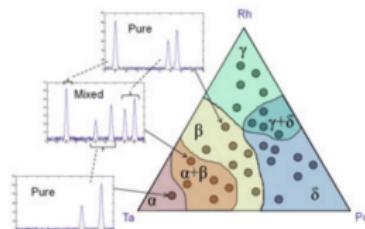
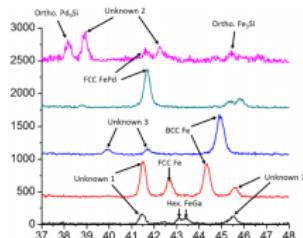
**Goal:** Estimate relation between electrolyte solution and viscosity, while simultaneously optimising conductivity.

# Posterior Estimation: Astrophysics



**Goal:** Estimate posterior for cosmological parameters given data.

# Phase Identification: Materials Science



**Goal:** Identify changes in crystal structure in an alloy.

# Outline

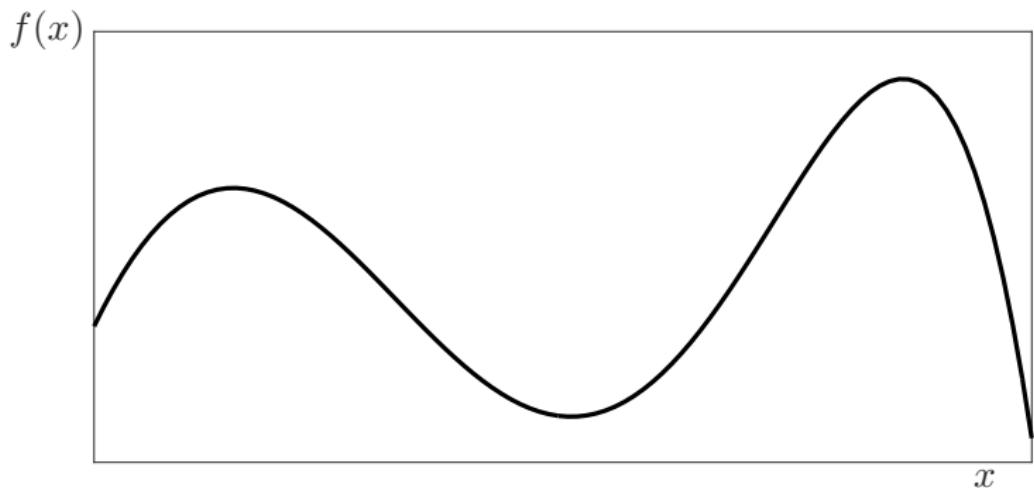
- ▶ Part I: Preliminaries (Black-box Optimisation)
  1. Bayesian Models
  2. Black-box Optimisation via Thompson Sampling
- ▶ Part II: DOE via posterior sampling
- ▶ Part III: Scaling up DOE (back to Black-box Optimisation)
  1. Parallelising experiments
  2. Multi-fidelity experimentation
  3. High dimensional input spaces
  4. Beyond Euclidean/categorical domains

# Outline

- ▶ Part I: Preliminaries (Black-box Optimisation)
  1. Bayesian Models
  2. Black-box Optimisation via Thompson Sampling
- ▶ Part II: DOE via posterior sampling
- ▶ Part III: Scaling up DOE (back to Black-box Optimisation)
  1. Parallelising experiments
  2. Multi-fidelity experimentation
  3. High dimensional input spaces
  4. Beyond Euclidean/categorical domains

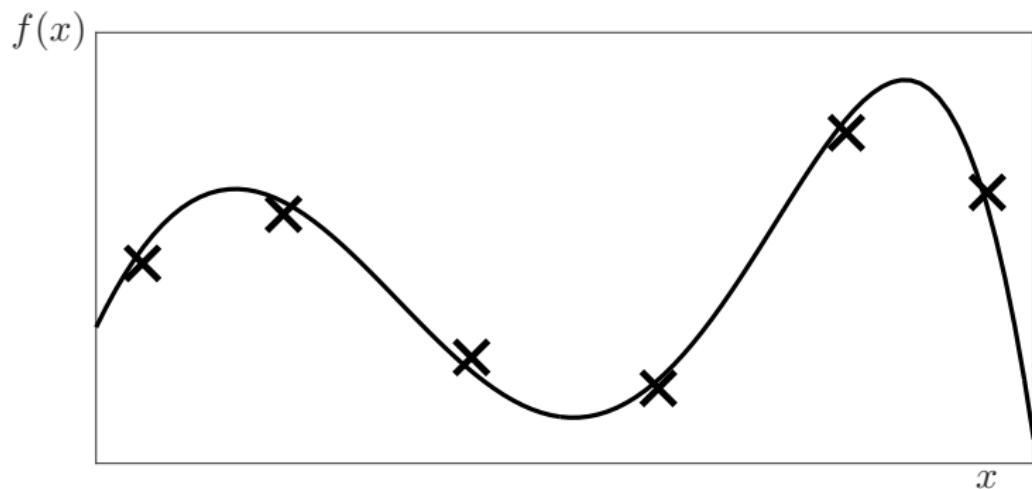
## Black-box Optimisation

$f : \mathcal{X} \rightarrow \mathbb{R}$  is an expensive, black-box function, accessible only via noisy evaluations.



## Black-box Optimisation

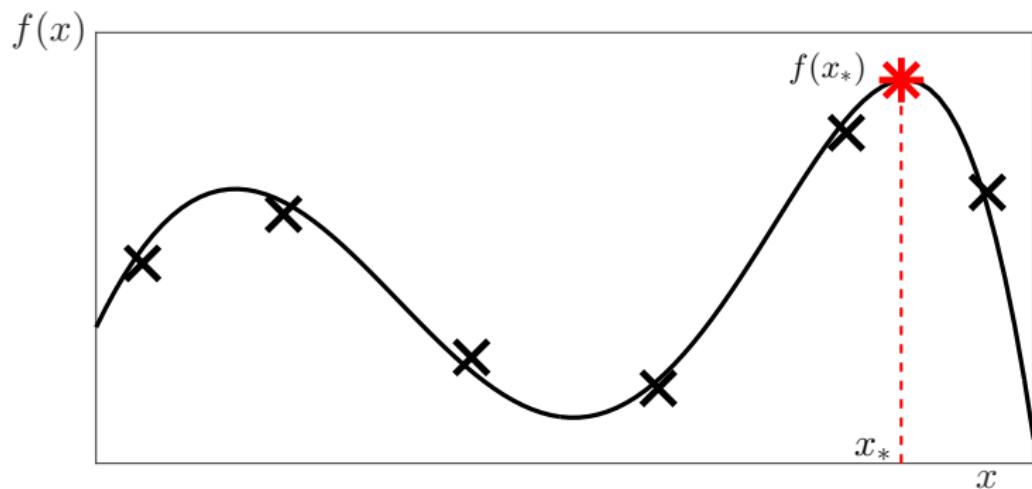
$f : \mathcal{X} \rightarrow \mathbb{R}$  is an expensive, black-box function, accessible only via noisy evaluations.



# Black-box Optimisation

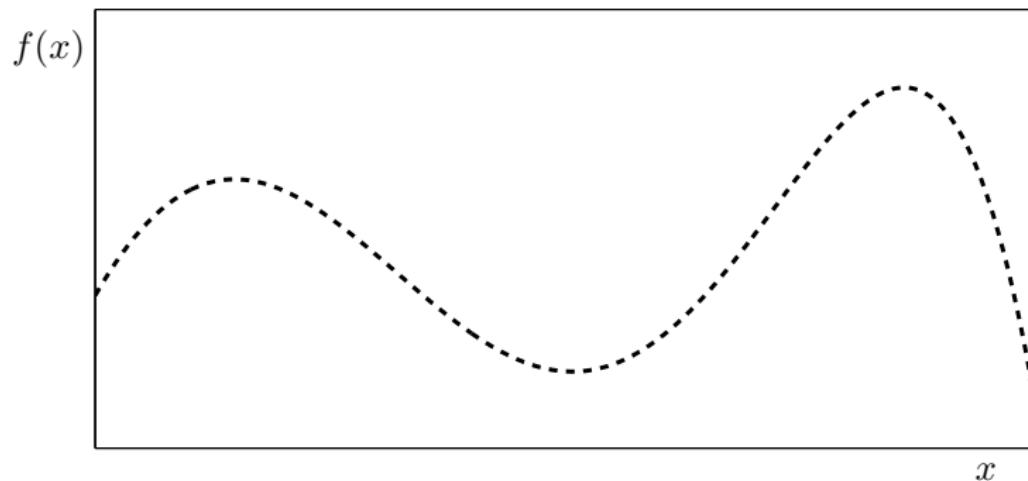
$f : \mathcal{X} \rightarrow \mathbb{R}$  is an expensive, black-box function, accessible only via noisy evaluations.

Let  $x_* = \operatorname{argmax}_x f(x)$ .



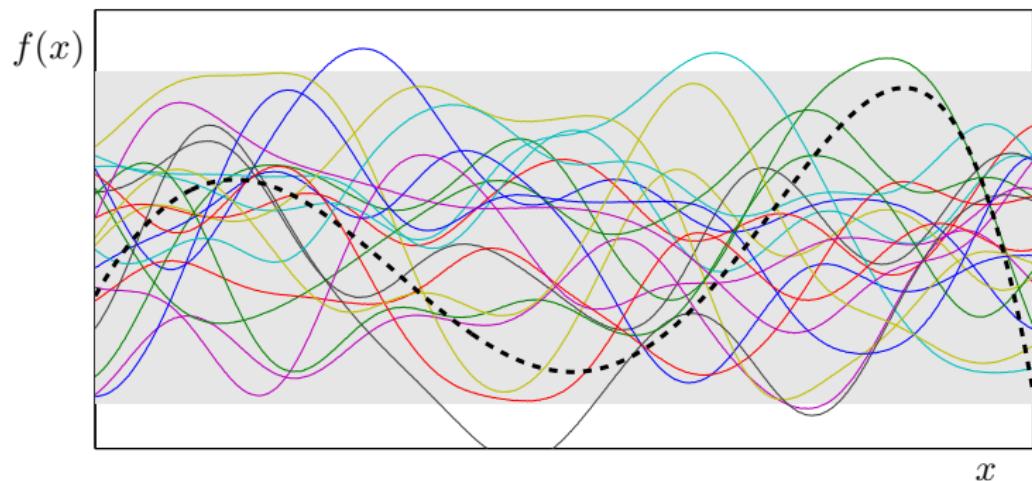
# Bayesian Models for $f$

Functions with no observations



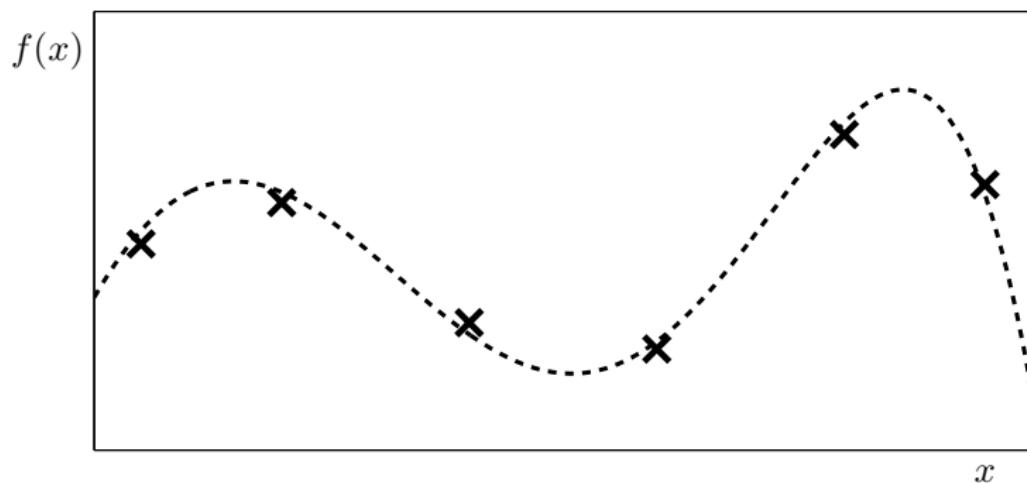
# Bayesian Models for $f$

Prior  $\mathbb{P}(\theta_*)$



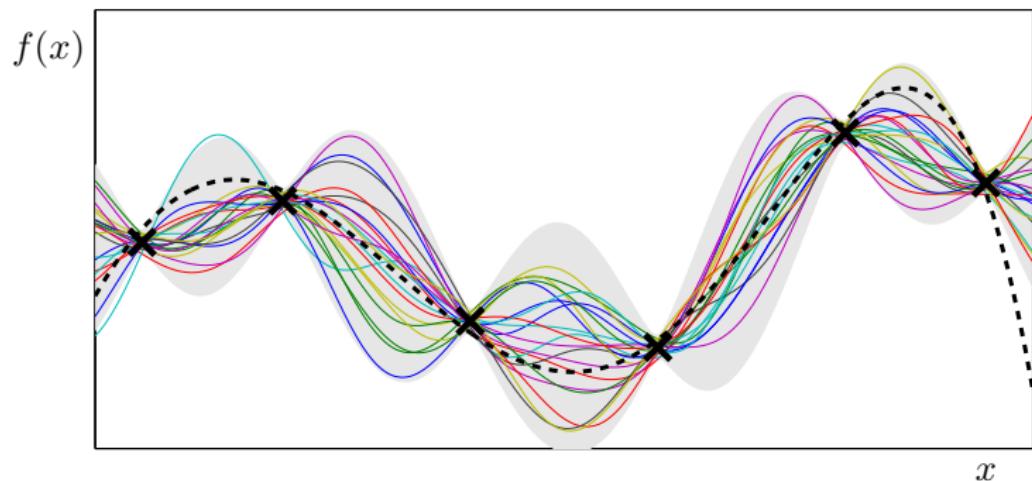
# Bayesian Models for $f$

Observations



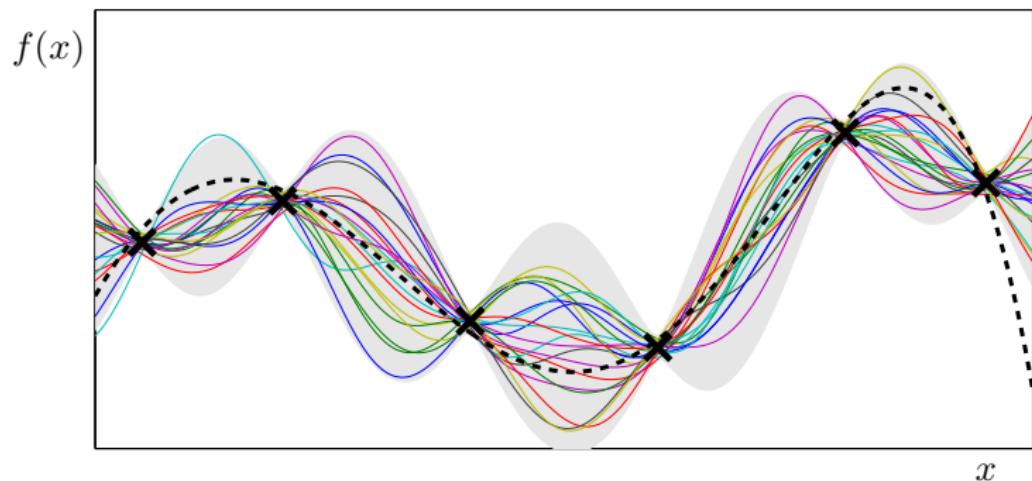
# Bayesian Models for $f$

Posterior given observations  $\mathbb{P}(\theta_\star|D_t)$



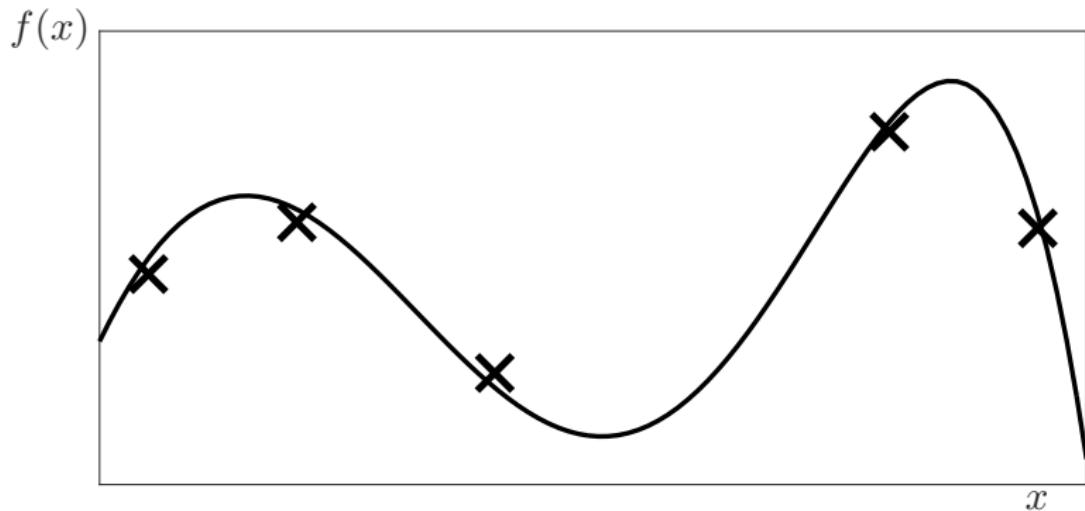
# Bayesian Models for $f$

Posterior given observations  $\mathbb{P}(\theta_\star | D_t)$



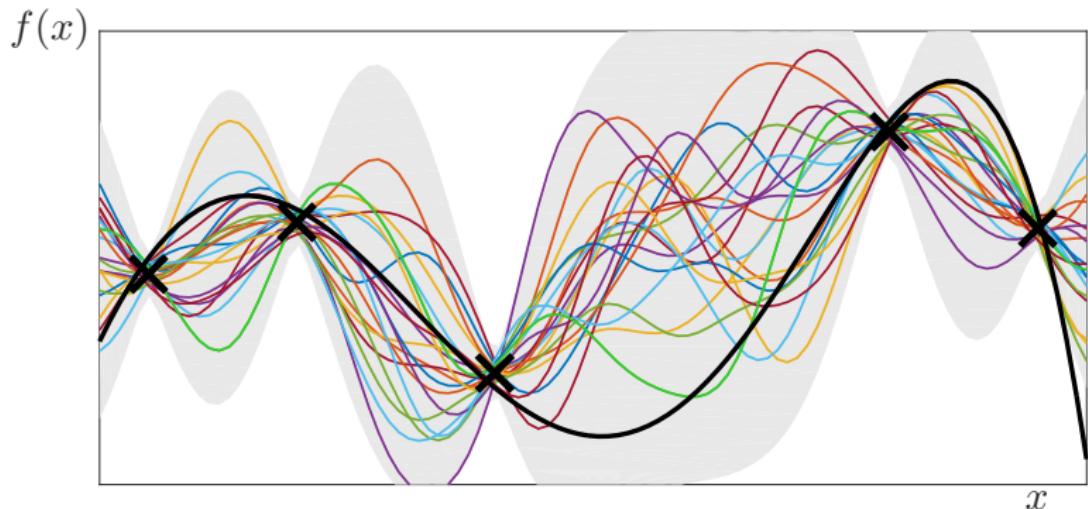
# Thompson Sampling for Black-box Optimisation

(Thompson, 1933)



# Thompson Sampling for Black-box Optimisation

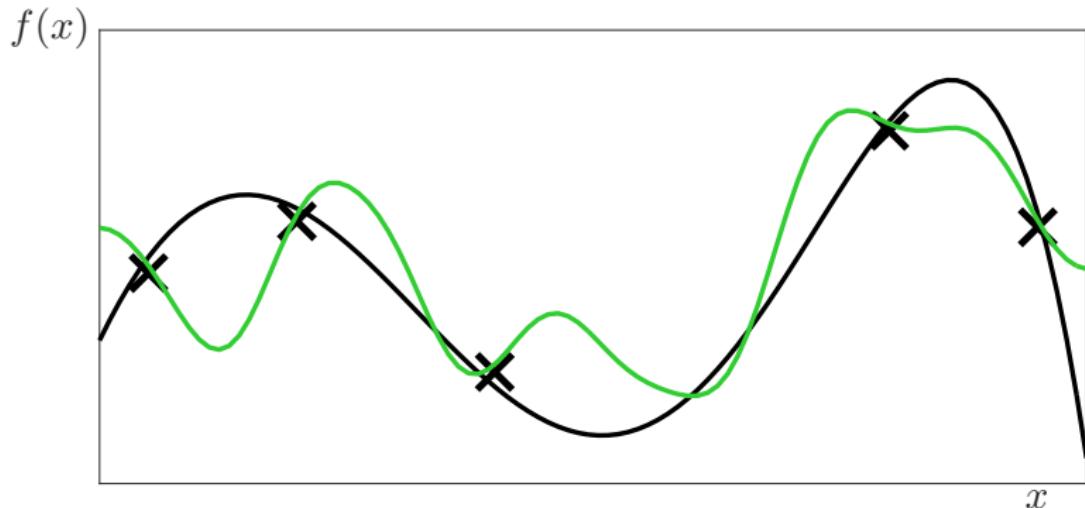
(Thompson, 1933)



- 1) Construct posterior  $\mathbb{P}(\theta_*|D_t)$ .

# Thompson Sampling for Black-box Optimisation

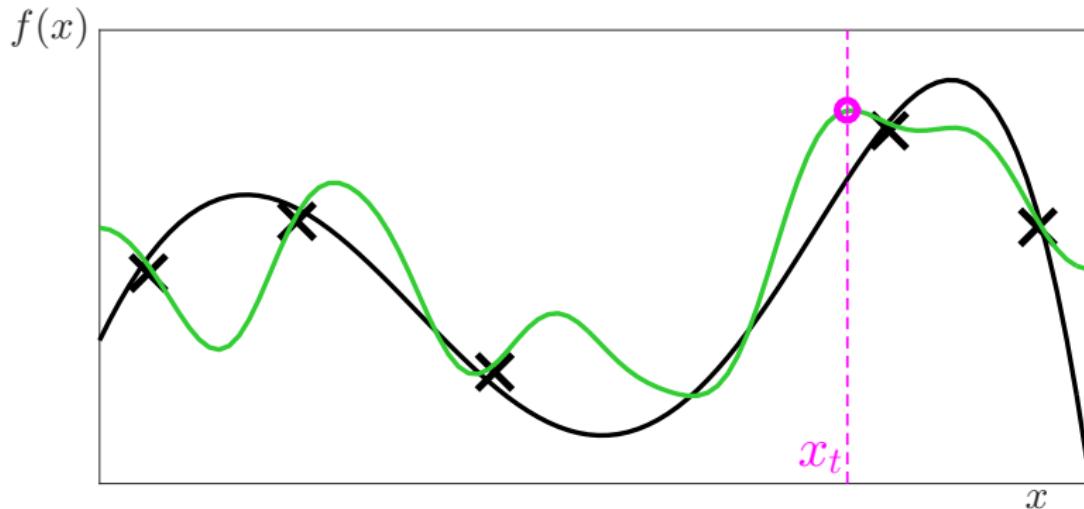
(Thompson, 1933)



- 1) Construct posterior  $\mathbb{P}(\theta_*|D_t)$ .
- 2) Draw sample  $g$  from posterior.

# Thompson Sampling for Black-box Optimisation

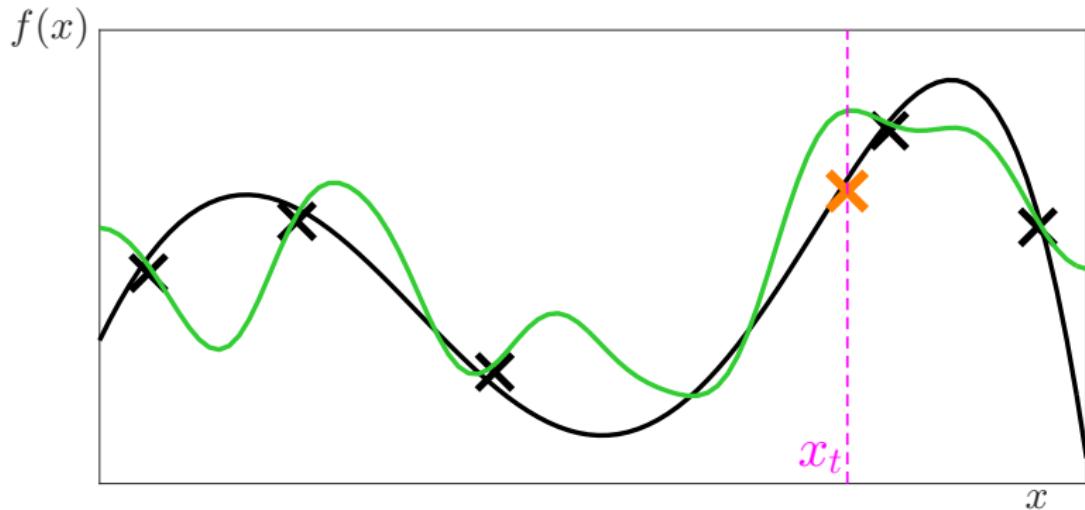
(Thompson, 1933)



- 1) Construct posterior  $\mathbb{P}(\theta_* | D_t)$ .
- 2) Draw sample  $g$  from posterior.
- 3) Choose  $x_t = \operatorname{argmax}_x g(x)$ .

# Thompson Sampling for Black-box Optimisation

(Thompson, 1933)

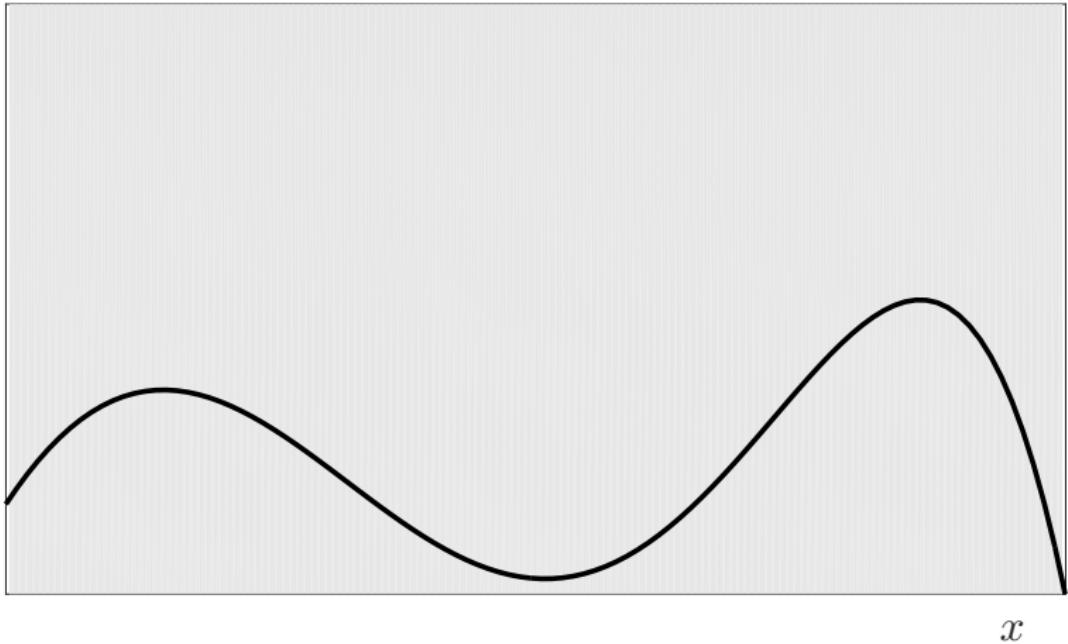


- 1) Construct posterior  $\mathbb{P}(\theta_* | D_t)$ .
- 2) Draw sample  $g$  from posterior.
- 3) Choose  $x_t = \text{argmax}_x g(x)$ .
- 4) Evaluate  $f$  at  $x_t$ .

# Thompson Sampling for Black-box Optimisation

(Thompson, 1933)

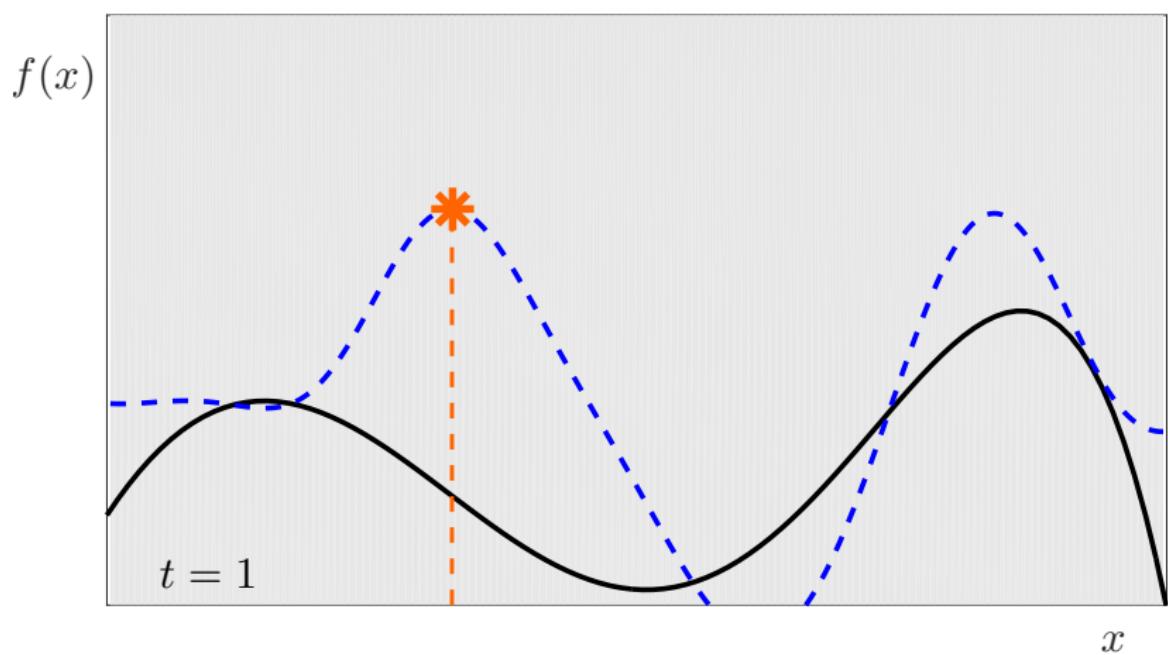
$$f(x)$$



$x$

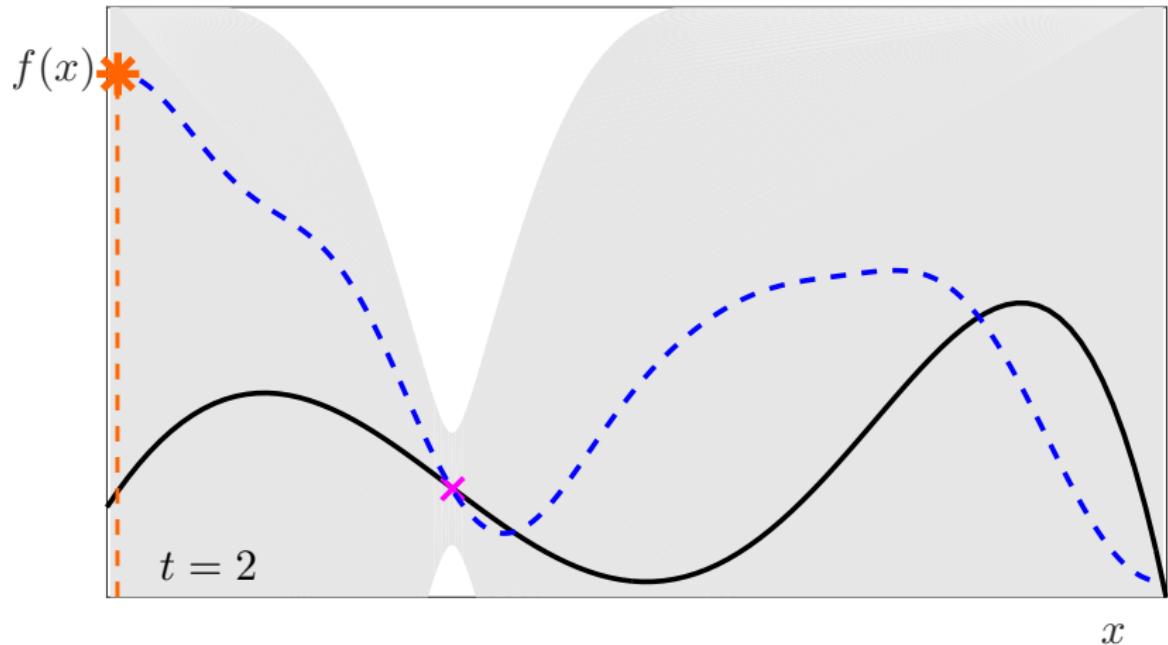
# Thompson Sampling for Black-box Optimisation

(Thompson, 1933)



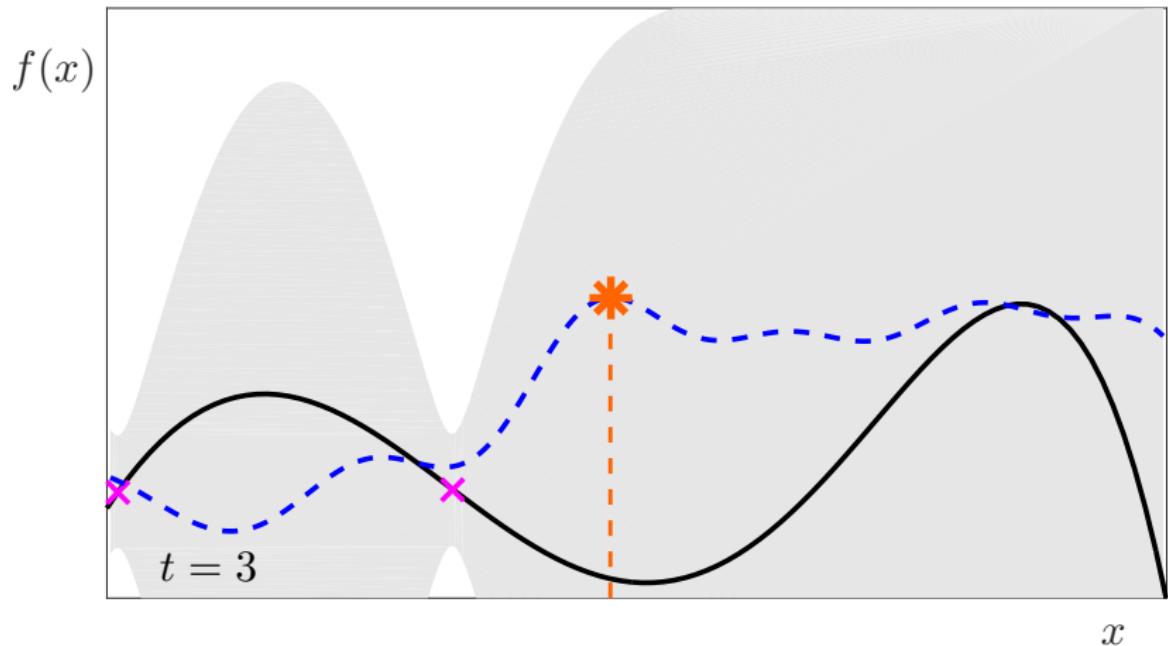
# Thompson Sampling for Black-box Optimisation

(Thompson, 1933)



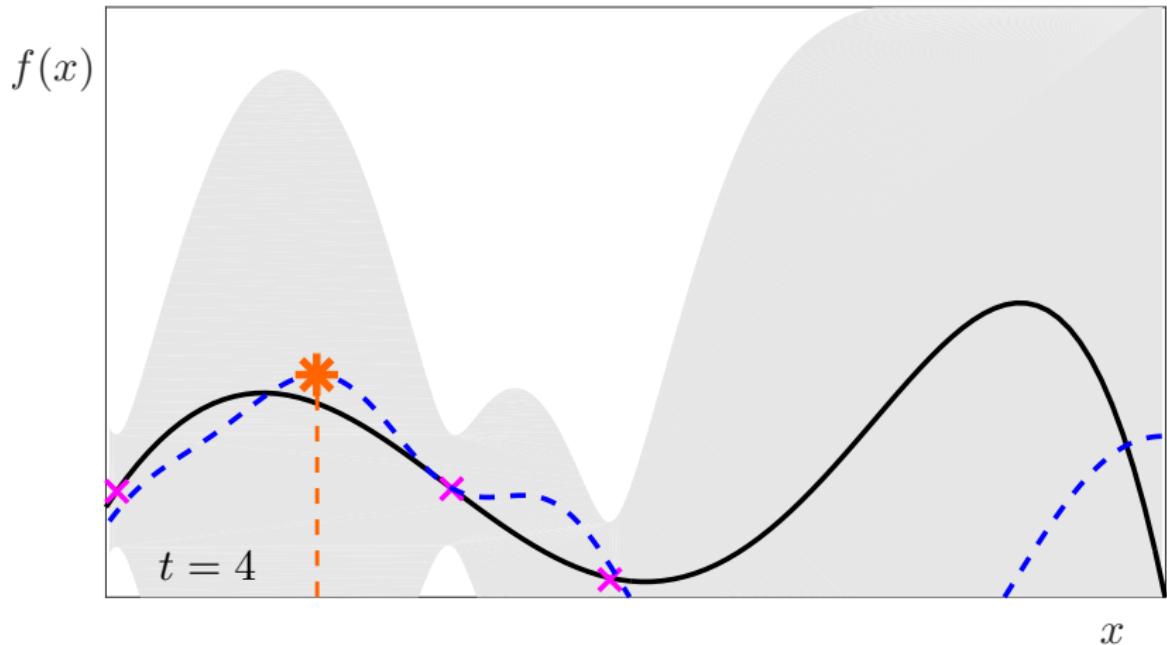
# Thompson Sampling for Black-box Optimisation

(Thompson, 1933)



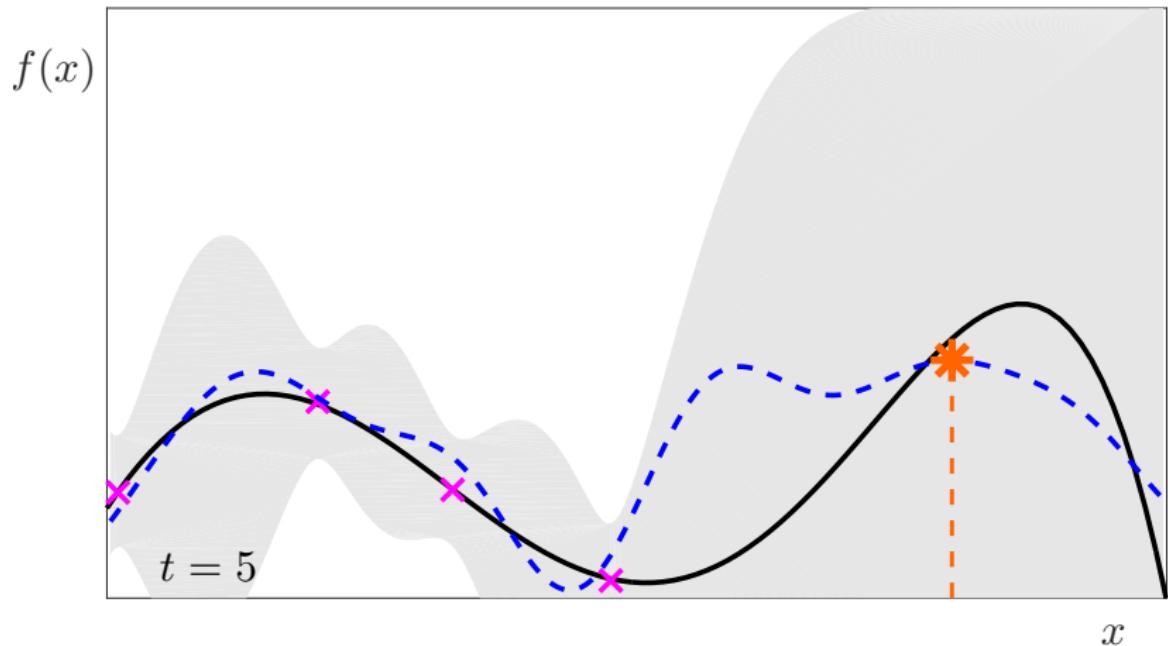
# Thompson Sampling for Black-box Optimisation

(Thompson, 1933)



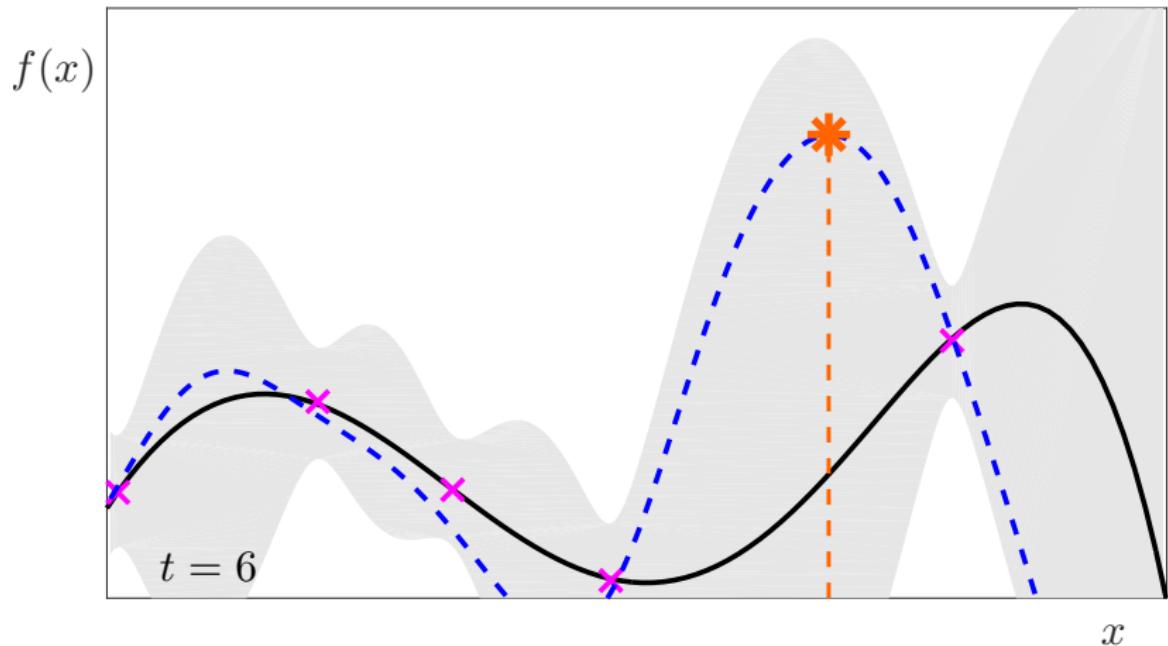
# Thompson Sampling for Black-box Optimisation

(Thompson, 1933)



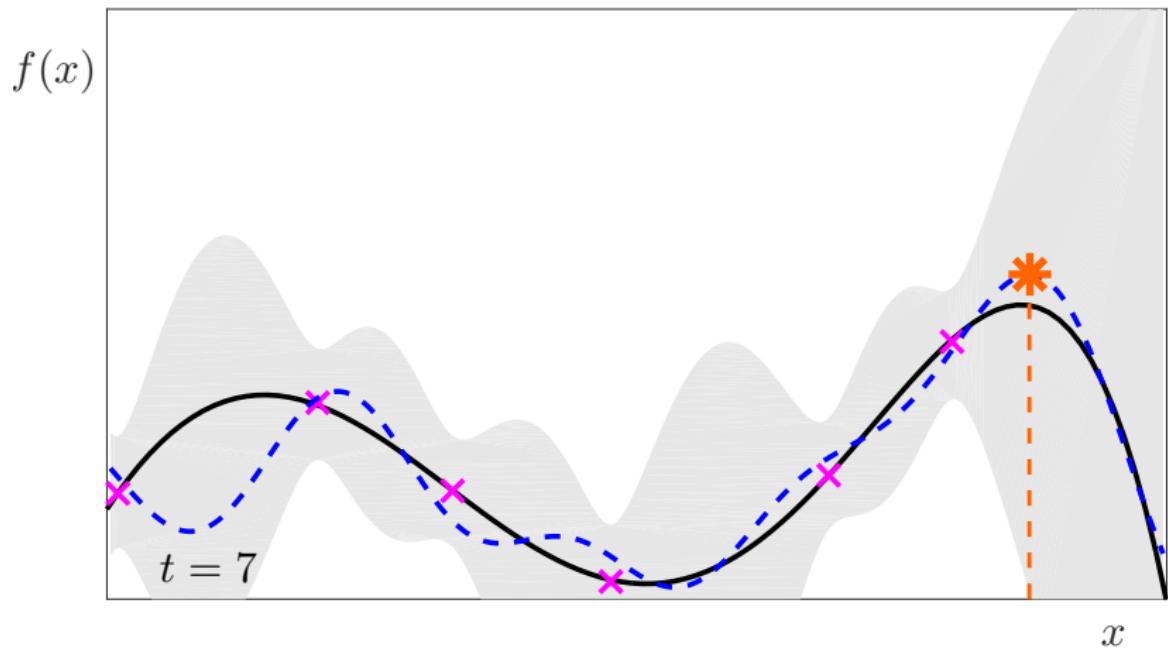
# Thompson Sampling for Black-box Optimisation

(Thompson, 1933)



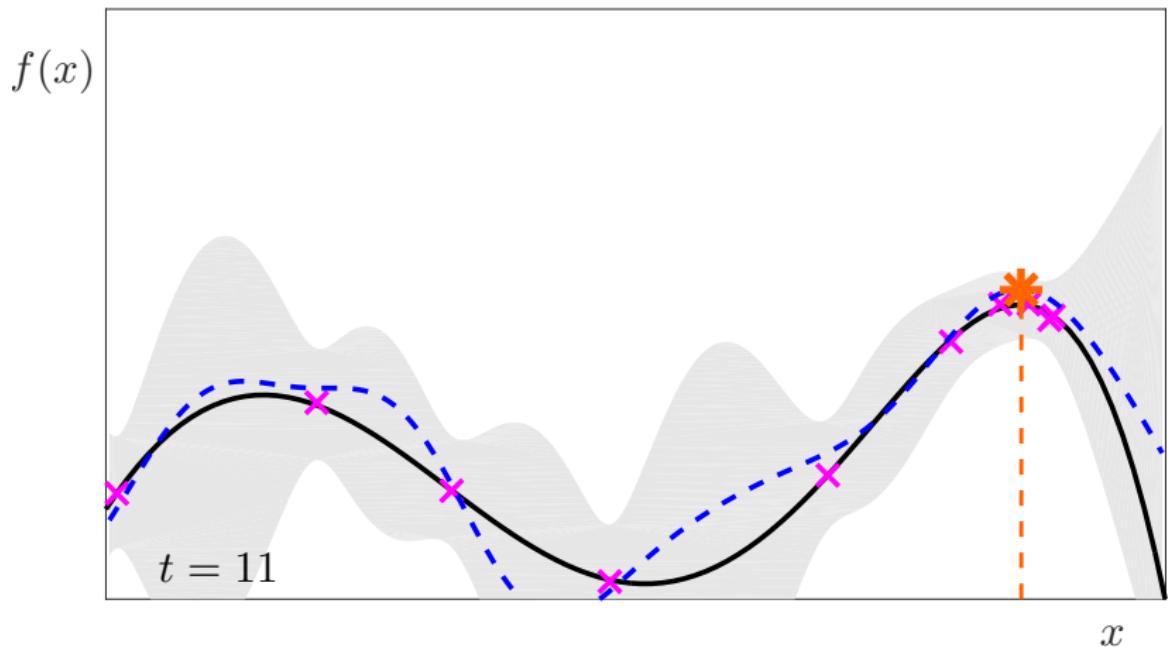
# Thompson Sampling for Black-box Optimisation

(Thompson, 1933)



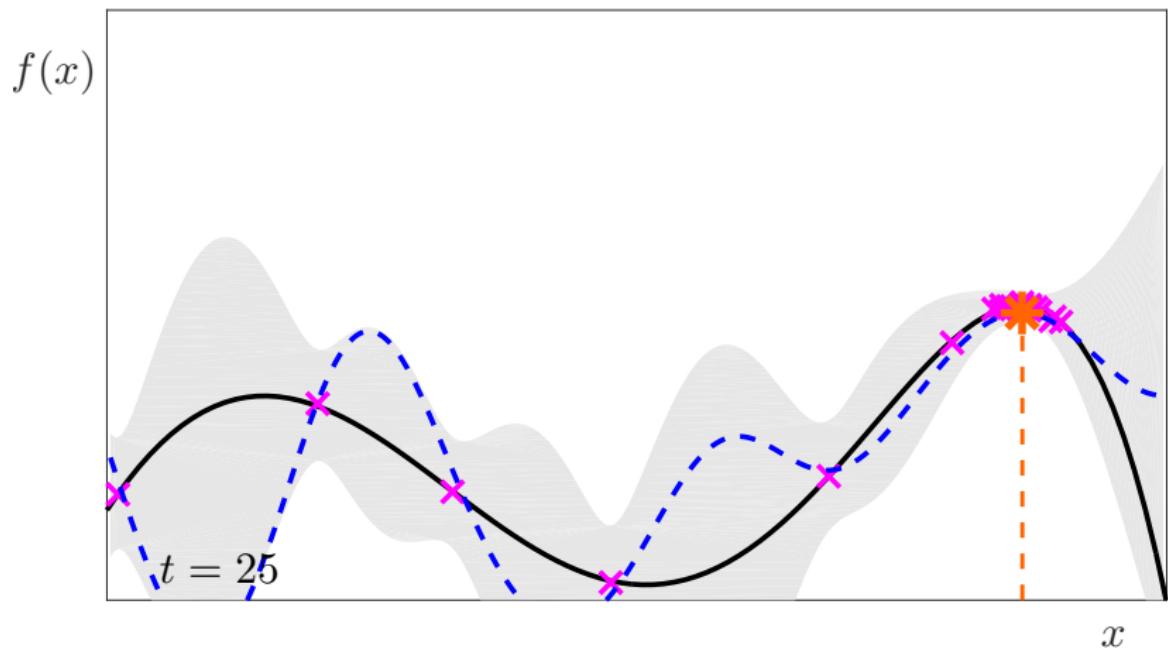
# Thompson Sampling for Black-box Optimisation

(Thompson, 1933)



# Thompson Sampling for Black-box Optimisation

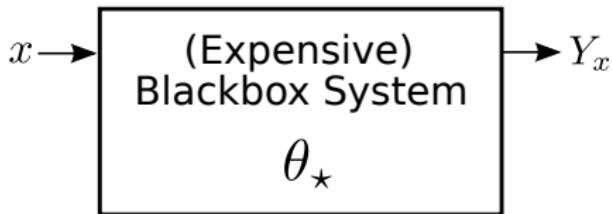
(Thompson, 1933)



# Outline

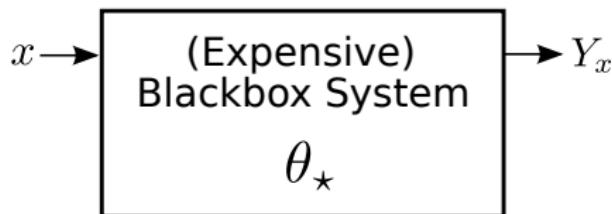
- ▶ Part I: Preliminaries (Black-box Optimisation)
  1. Bayesian Models
  2. Black-box Optimisation via Thompson Sampling
- ▶ Part II: DOE via posterior sampling
- ▶ Part III: Scaling up DOE (back to Black-box Optimisation)
  1. Parallelising experiments
  2. Multi-fidelity experimentation
  3. High dimensional input spaces
  4. Beyond Euclidean/categorical domains

## Design of Experiments



- ▶ Choose experiment  $X \in \mathcal{X}$ , obtain result  $Y_X \sim \mathbb{P}(y|X, \theta_\star)$ .
- ▶  $\theta_\star$  represents everything that is unknown about the system.
- ▶ Repeat in a sequence to collect data  $D_t = \{(X_j, Y_{X_j})\}_{j=1}^t$ .
- ▶ Typically some “goal” in mind.

# Design of Experiments



- ▶ Choose experiment  $X \in \mathcal{X}$ , obtain result  $Y_X \sim \mathbb{P}(y|X, \theta_*)$ .
- ▶  $\theta_*$  represents everything that is unknown about the system.
- ▶ Repeat in a sequence to collect data  $D_t = \{(X_j, Y_{X_j})\}_{j=1}^t$ .
- ▶ Typically some “goal” in mind.

## Desiderata for a General Framework:

- ▶ Flexibility to capture custom/complex relations for  $X \rightarrow Y_X$ .
  - Incorporate domain expertise into models.
- ▶ Ability to achieve any desired goal.

# Formalism for “goal-oriented” DOE

## System:

- ▶ A true parameter  $\theta_* \in \Theta$  that completely specifies the system.
- ▶  $\Theta \leftarrow$  a parameter space.

# Formalism for “goal-oriented” DOE

## System:

- ▶ A true parameter  $\theta_* \in \Theta$  that completely specifies the system.
- ▶  $\Theta \leftarrow$  a parameter space.

## Goal:

- ▶ Collect data  $D_n = \{(x_t, y_{x_t})\}_{t=1}^n$  to achieve a goal specified by a penalty function  $\lambda(\theta, D_n)$ .
- ▶ We wish to achieve small  $\lambda(\theta_*, D_n)$  after  $n$  experiments.

# Formalism for “goal-oriented” DOE

## System:

- ▶ A true parameter  $\theta_\star \in \Theta$  that completely specifies the system.
- ▶  $\Theta \leftarrow$  a parameter space.

## Goal:

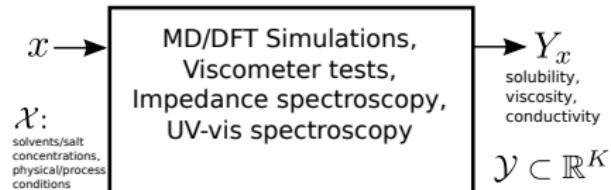
- ▶ Collect data  $D_n = \{(x_t, y_{x_t})\}_{t=1}^n$  to achieve a goal specified by a penalty function  $\lambda(\theta, D_n)$ .
- ▶ We wish to achieve small  $\lambda(\theta_\star, D_n)$  after  $n$  experiments.

## Bayesian Models:

- ▶ A prior for  $\theta_\star$ :  $\mathbb{P}(\theta_\star)$ .
- ▶ A discriminative model for observations  $y|x, \theta$ :  $\mathbb{P}(y|x, \theta)$ .

# Incorporating Domain Expertise via Bayesian Models

## An example in Electrolyte Design

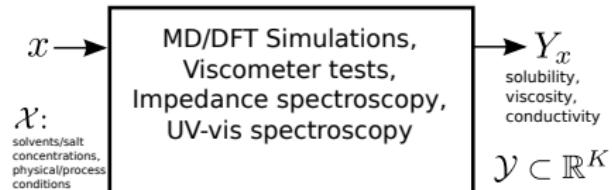


Three control variables:

$Q$ : EC-EMC fraction,  $S$ : molarity of salt LiPF<sub>6</sub>,  $T$ : temperature.

# Incorporating Domain Expertise via Bayesian Models

## An example in Electrolyte Design



Three control variables:

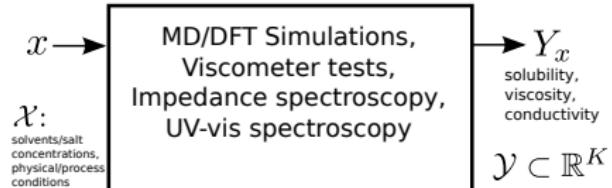
$Q$ : EC-EMC fraction,  $S$ : molarity of salt LiPF<sub>6</sub>,  $T$ : temperature.

$$f_{\text{vis}}(Q, S, T) = \exp(-aT + bS) \cdot g_{\text{vis}}(Q). \quad (\text{Reynolds model})$$

$$f_{\text{dissol}}(Q, S, T) = cT \cdot \frac{1}{1 + \exp(dS)} \cdot g_{\text{dissolv}}(Q).$$

# Incorporating Domain Expertise via Bayesian Models

## An example in Electrolyte Design



Three control variables:

$Q$ : EC-EMC fraction,  $S$ : molarity of salt  $\text{LiPF}_6$ ,  $T$ : temperature.

$$f_{\text{vis}}(Q, S, T) = \exp(-aT + bS) \cdot g_{\text{vis}}(Q). \quad (\text{Reynolds model})$$

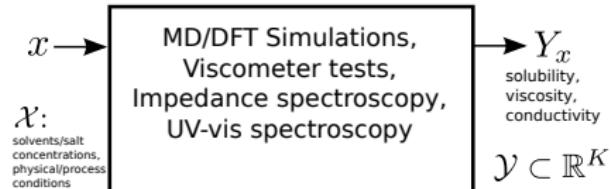
$$f_{\text{dissol}}(Q, S, T) = cT \cdot \frac{1}{1 + \exp(dS)} \cdot g_{\text{dissolv}}(Q).$$

Observations:  $y_{\text{visc}}|x, \theta \sim \mathcal{N}(f_{\text{vis}}(Q, S, T), \eta^2)$ .

$y_{\text{dissol}}|x, \theta \sim \mathcal{N}(f_{\text{dissol}}(Q, S, T), \sigma^2)$ .

# Incorporating Domain Expertise via Bayesian Models

## An example in Electrolyte Design



Three control variables:

$Q$ : EC-EMC fraction,  $S$ : molarity of salt  $\text{LiPF}_6$ ,  $T$ : temperature.

$$f_{\text{vis}}(Q, S, T) = \exp(-aT + bS) \cdot g_{\text{vis}}(Q). \quad (\text{Reynolds model})$$

$$f_{\text{dissol}}(Q, S, T) = cT \cdot \frac{1}{1 + \exp(dS)} \cdot g_{\text{dissolv}}(Q).$$

Observations:  $y_{\text{visc}}|x, \theta \sim \mathcal{N}(f_{\text{vis}}(Q, S, T), \eta^2)$ .

$y_{\text{dissol}}|x, \theta \sim \mathcal{N}(f_{\text{dissol}}(Q, S, T), \sigma^2)$ .

Unknown parameters:  $\theta = (a, b, c, d, g_{\text{vis}}, g_{\text{dissolv}}) \in \Theta$ .

True parameter  $\theta_* = (a_*, b_*, c_*, d_*, g_{\text{vis}*}, g_{\text{dissolv}*})$ .

Use prior  $\mathbb{P}(\theta_*)$  to specify plausible values for  $\theta_*$ .

## Specifying the goal via a Penalty Function

Practitioner specifies goal of the experiment via  $\lambda(\theta, D_n)$ .

## Specifying the goal via a Penalty Function

Practitioner specifies goal of the experiment via  $\lambda(\theta, D_n)$ .

### Example 1: Optimisation

$$\lambda(\theta, D_n) = \max_{x \in \mathcal{X}} f_\theta(x) - \max_{t \leq n} f_\theta(x_t)$$

# Specifying the goal via a Penalty Function

Practitioner specifies goal of the experiment via  $\lambda(\theta, D_n)$ .

## Example 1: Optimisation

$$\lambda(\theta, D_n) = \max_{x \in \mathcal{X}} f_\theta(x) - \max_{t \leq n} f_\theta(x_t)$$

## Example 2: Active Learning

Estimate some parameter  $\tau_* = \tau(\theta_*)$  of the system.

$$\lambda(\theta, D_n) = \|\tau(\theta) - \hat{\tau}(D_n)\|_2^2.$$

$\hat{\tau} \leftarrow$  some prespecified (e.g. maximum likelihood) estimator for  $\tau$  using data.

# Specifying the goal via a Penalty Function

Practitioner specifies goal of the experiment via  $\lambda(\theta, D_n)$ .

## Example 1: Optimisation

$$\lambda(\theta, D_n) = \max_{x \in \mathcal{X}} f_\theta(x) - \max_{t \leq n} f_\theta(x_t)$$

## Example 2: Active Learning

Estimate some parameter  $\tau_* = \tau(\theta_*)$  of the system.

$$\lambda(\theta, D_n) = \|\tau(\theta) - \hat{\tau}(D_n)\|_2^2.$$

$\hat{\tau} \leftarrow$  some prespecified (e.g. maximum likelihood) estimator for  $\tau$  using data.

Will look at more examples shortly.

## MPS: Myopic Posterior Sampling for DOE

Expected look-ahead penalty at  $x$  if  $\theta$  was the true parameter and we have already collected data  $D$ :

$$\lambda^+(\theta, D, x) = \mathbb{E}_{Y_x \sim \mathbb{P}(Y|x, \theta)} [\lambda(\theta, D \cup \{(x, Y_x)\})].$$

## MPS: Myopic Posterior Sampling for DOE

Expected look-ahead penalty at  $x$  if  $\theta$  was the true parameter and we have already collected data  $D$ :

$$\lambda^+(\theta, D, x) = \mathbb{E}_{Y_x \sim \mathbb{P}(Y|x, \theta)} [\lambda(\theta, D \cup \{(x, Y_x)\})].$$

---

### Algorithm: MPS ( $\pi_M^{PS}$ )

---

- Set  $D_0 \leftarrow$  initial data.
  - For  $t = 1, 2, \dots$ , do
    - 1. Sample  $\theta \sim \mathbb{P}(\theta_\star | D_{t-1})$ .
    - 2. Choose  $x_t = \operatorname{argmin}_{x \in \mathcal{X}} \lambda^+(\theta, D_{t-1}, x)$ .
    - 3.  $y_{x_t} \leftarrow$  conduct experiment at  $x_t$ .
    - 4. Set  $D_t \leftarrow D_{t-1} \cup \{(x_t, y_{x_t})\}$ .
-

## MPS: Myopic Posterior Sampling for DOE

Expected look-ahead penalty at  $x$  if  $\theta$  was the true parameter and we have already collected data  $D$ :

$$\lambda^+(\theta, D, x) = \mathbb{E}_{Y_x \sim \mathbb{P}(Y|x, \theta)} [\lambda(\theta, D \cup \{(x, Y_x)\})].$$

---

### Algorithm: MPS ( $\pi_M^{PS}$ )

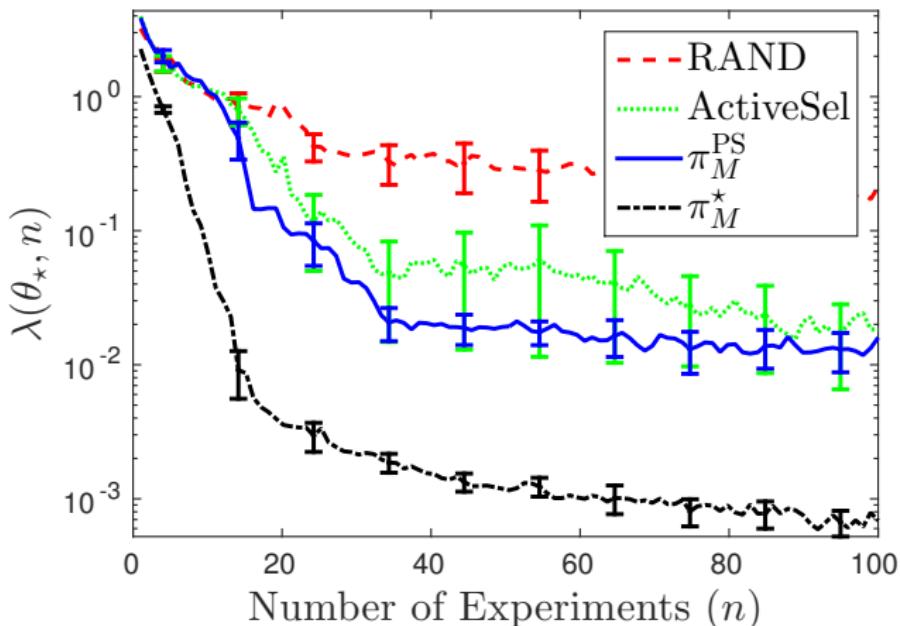
---

- Set  $D_0 \leftarrow$  initial data.
  - For  $t = 1, 2, \dots$ , do
    - 1. Sample  $\theta \sim \mathbb{P}(\theta_* | D_{t-1})$ .
    - 2. Choose  $x_t = \operatorname{argmin}_{x \in \mathcal{X}} \lambda^+(\theta, D_{t-1}, x)$ .
    - 3.  $y_{x_t} \leftarrow$  conduct experiment at  $x_t$ .
    - 4. Set  $D_t \leftarrow D_{t-1} \cup \{(x_t, y_{x_t})\}$ .
- 

**N.B:** When the goal is optimisation, this reduces to exactly Thompson sampling.

# Experiment: Active Learning

Linear RBF

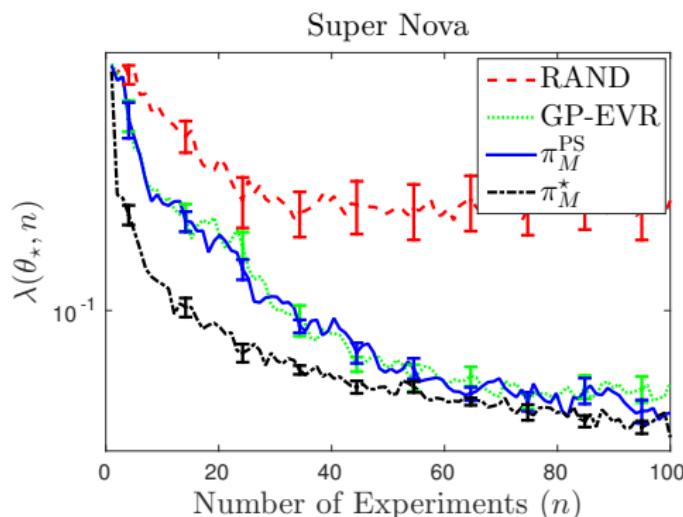


ActiveSel: (Chaudhuri et al. 2015)

# Experiment: Posterior Estimation in Astrophysics

Astrophysicist defines prior on Hubble constant, and dark matter fraction and dark energy fraction. Computer posterior distribution given Type Ia supernova data  $Q$ . Likelihood computed using the Robertson-Walker metric.

$$\lambda(\theta_*, D_n) = \|p(\tau(\theta_*)|Q) - \hat{p}(\tau(\theta_*)|Q)\|_2$$



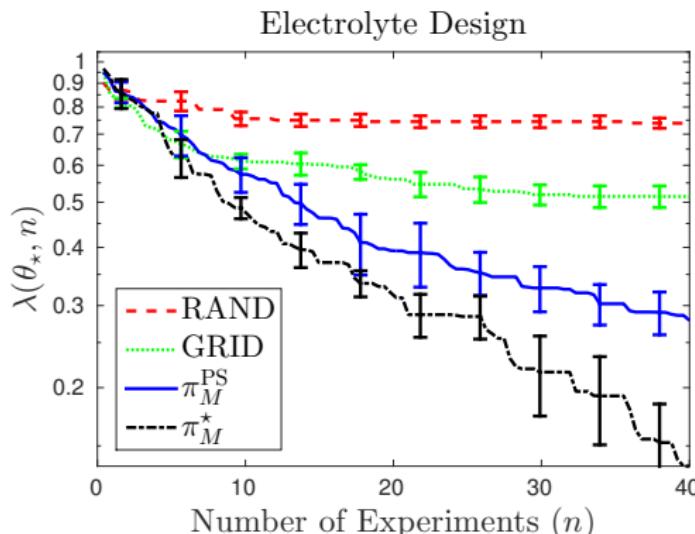
GP-EVR: ([Kandasamy et al. IJCAI 2015](#))

# Experiment: Custom goal in Electrolyte Design

An experiment measures solubility, viscosity and conductivity of an electrolyte design.

**Goal:** Optimise conductivity while learning solubility and viscosity.

$$\lambda(\theta_*, D_n) = \|f_{\text{dissol}} - \hat{f}_{\text{dissol}}(D_n)\|^2 + \|f_{\text{vis}} - \hat{f}_{\text{vis}}(D_n)\|^2 + (\max f_{\text{con}} - \max_{X_t, t \leq n} f_{\text{con}}(X_t)),$$



# Theory

**Theorem (Informal):** Under certain assumptions on the problem, MPS does almost as well as the optimal algorithm that knows  $\theta_*$ .

We use ideas/conditions from

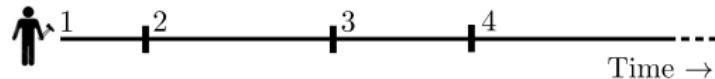
- Adaptive Submodularity
- Re-inforcement Learning
- Bandits

# Outline

- ▶ Part I: Preliminaries (Black-box Optimisation)
  1. Bayesian Models
  2. Black-box Optimisation via Thompson Sampling
- ▶ Part II: DOE via posterior sampling
- ▶ Part III: Scaling up DOE (back to Black-box Optimisation)
  1. Parallelising experiments
  2. Multi-fidelity experimentation
  3. High dimensional input spaces
  4. Beyond Euclidean/categorical domains

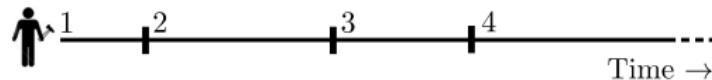
## Part 3.1: Parallel Experiments

Sequential experiments with one worker

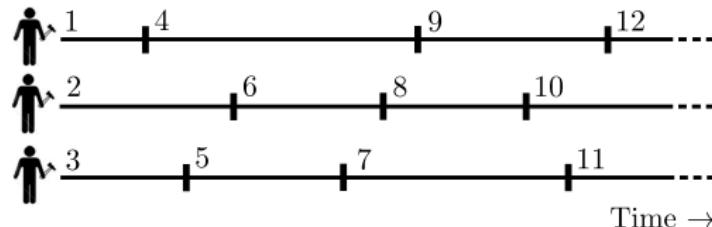


## Part 3.1: Parallel Experiments

Sequential experiments with one worker

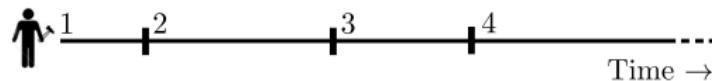


Parallel experiments with  $M$  workers (Asynchronous)

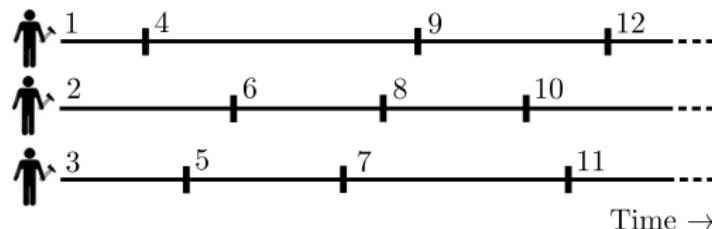


## Part 3.1: Parallel Experiments

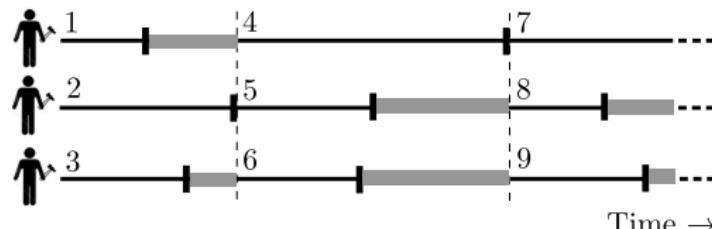
Sequential experiments with one worker



Parallel experiments with  $M$  workers (Asynchronous)



Parallel experiments with  $M$  workers (Synchronous)



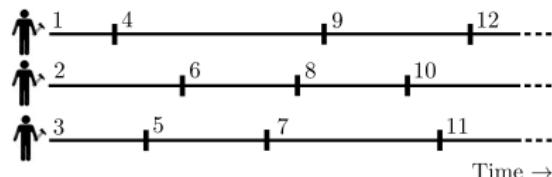
# Parallelised DOE via Posterior Sampling

## Asynchronous:

---

At any given time,

1.  $(x', y') \leftarrow$  Wait for a worker to finish.
  2. Update posterior for  $\theta_*$ .
  3. Draw a sample  
 $\theta \sim \mathbb{P}(\theta_* | D_t)$ .
  4. Re-deploy worker at  
 $\operatorname{argmin} \lambda^+(\theta, D_t, x)$ .
- 

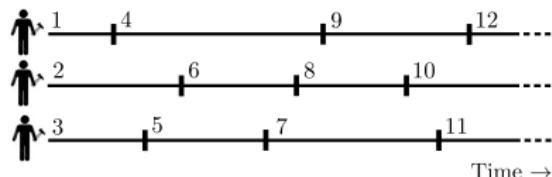


# Parallelised DOE via Posterior Sampling

## Asynchronous:

At any given time,

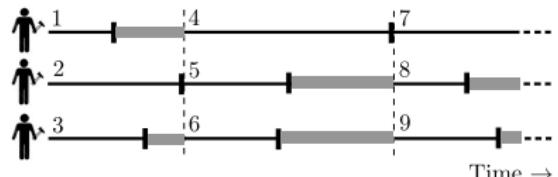
1.  $(x', y') \leftarrow$  Wait for a worker to finish.
2. Update posterior for  $\theta_*$ .
3. Draw a sample  $\theta \sim \mathbb{P}(\theta_* | D_t)$ .
4. Re-deploy worker at  $\operatorname{argmin} \lambda^+(\theta, D_t, x)$ .



## Synchronous:

At any given time,

1.  $\{(x'_m, y'_m)\}_{m=1}^M \leftarrow$  Wait for all workers to finish.
2. Update posterior for  $\theta_*$ .
3. Draw  $M$  samples  $\theta_m \sim \mathbb{P}(\theta_* | D_t), \forall m$ .
4. Re-deploy worker  $m$  at  $\operatorname{argmin} \lambda^+(\theta_m, D_t, x)$ .



## Theory: parallel DOE via posterior sampling

**Conjecture:** For synchronous & asynchronous parallel DOE via posterior sampling

$$\mathbb{E}[\lambda(\theta_*, D_n)] \lesssim \frac{M \log(M)}{n} + \text{sequential result}$$

**Theorem:** For parallelised Thompson sampling (Black-box Optimisation) (Kandasamy et al. AISTATS 2018)

$$\mathbb{E}[f(x_*) - \max_{t \leq n} f(x_t)] \lesssim \frac{M \log(M)}{n} + \frac{C}{\sqrt{n}}$$

Can also quantify difference between synchronous and asynchronous settings.

(Kandasamy et al. AISTATS 2018)

- ▶ If evaluation times are the same, synchronous is slightly better.
- ▶ When there is high variability in evaluation times, asynchronous is much better than synchronous.

# Outline

- ▶ Part I: Preliminaries (Black-box Optimisation)
  1. Bayesian Models
  2. Black-box Optimisation via Thompson Sampling
- ▶ Part II: DOE via posterior sampling
- ▶ Part III: Scaling up DOE (back to Black-box Optimisation)
  1. Parallelising experiments
  2. Multi-fidelity experimentation
  3. High dimensional input spaces
  4. Beyond Euclidean/categorical domains

## Part 3.2: Multi-fidelity Optimisation

**Motivating question:**

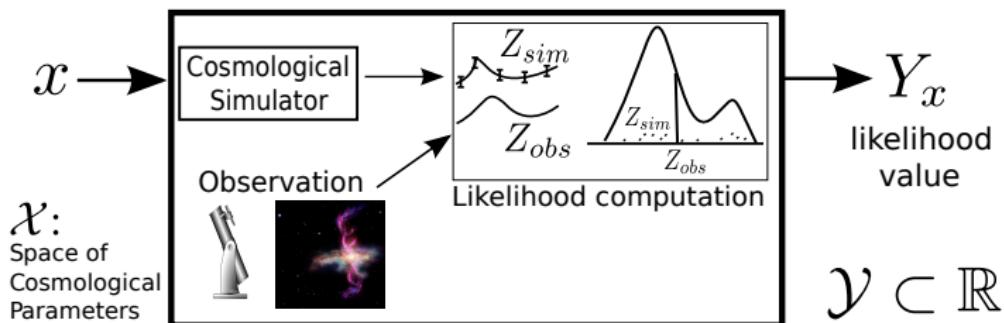
What if we have cheap approximations to  $f$ ?

## Part 3.2: Multi-fidelity Optimisation

### Motivating question:

What if we have cheap approximations to  $f$ ?

1. Computational astrophysics and other scientific experiments: simulations and numerical computations with less granularity.

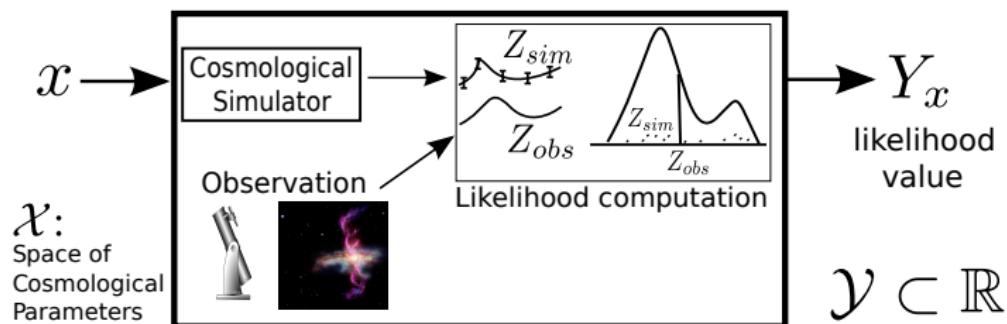


## Part 3.2: Multi-fidelity Optimisation

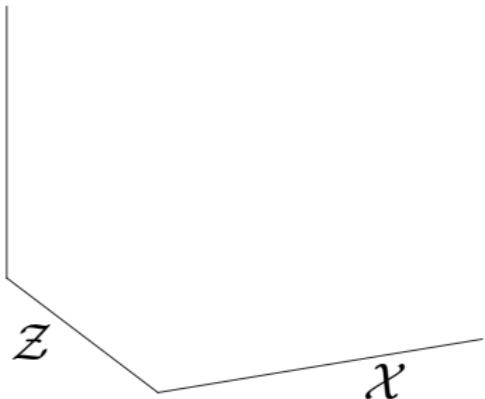
### Motivating question:

What if we have cheap approximations to  $f$ ?

1. Computational astrophysics and other scientific experiments: simulations and numerical computations with less granularity.



2. Hyper-parameter tuning: Train & validate with a subset of the data.
3. In many applications: computer simulation vs real world experiment.



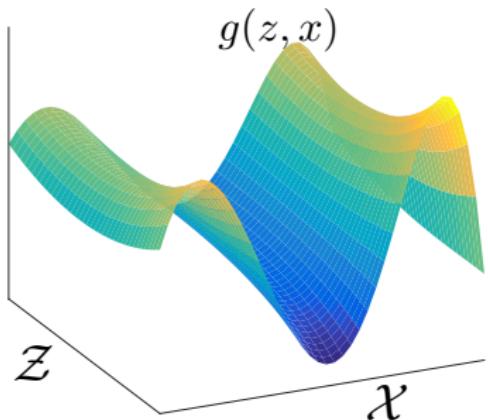
A fidelity space  $\mathcal{Z}$  and domain  $\mathcal{X}$

$\mathcal{Z} \leftarrow$  all granularity values

$\mathcal{X} \leftarrow$  space of cosmological parameters

## Multi-fidelity Optimisation

(Kandasamy et al. ICML 2017)



A fidelity space  $\mathcal{Z}$  and domain  $\mathcal{X}$

$\mathcal{Z} \leftarrow$  all granularity values

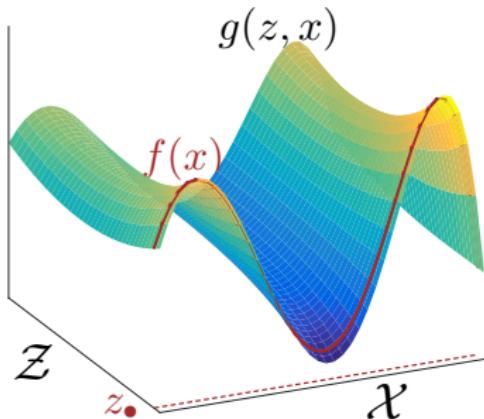
$\mathcal{X} \leftarrow$  space of cosmological parameters

$$g : \mathcal{Z} \times \mathcal{X} \rightarrow \mathbb{R}.$$

$g(z, x) \leftarrow$  likelihood when performing integrations on a grid of size  $z$  at cosmological parameters  $x$ .

## Multi-fidelity Optimisation

(Kandasamy et al. ICML 2017)



A fidelity space  $\mathcal{Z}$  and domain  $\mathcal{X}$

$\mathcal{Z} \leftarrow$  all granularity values

$\mathcal{X} \leftarrow$  space of cosmological parameters

$$g : \mathcal{Z} \times \mathcal{X} \rightarrow \mathbb{R}.$$

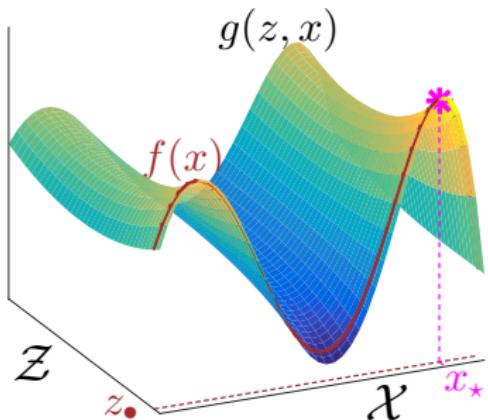
$g(z, x) \leftarrow$  likelihood when performing integrations on a grid of size  $z$  at cosmological parameters  $x$ .

Denote  $f(x) = g(z_•, x)$  where  $z_• \in \mathcal{Z}$ .

$z_• =$  highest grid size.

# Multi-fidelity Optimisation

(Kandasamy et al. ICML 2017)



A fidelity space  $\mathcal{Z}$  and domain  $\mathcal{X}$

$\mathcal{Z} \leftarrow$  all granularity values

$\mathcal{X} \leftarrow$  space of cosmological parameters

$$g : \mathcal{Z} \times \mathcal{X} \rightarrow \mathbb{R}.$$

$g(z, x) \leftarrow$  likelihood when performing integrations on a grid of size  $z$  at cosmological parameters  $x$ .

Denote  $f(x) = g(z_•, x)$  where  $z_• \in \mathcal{Z}$ .

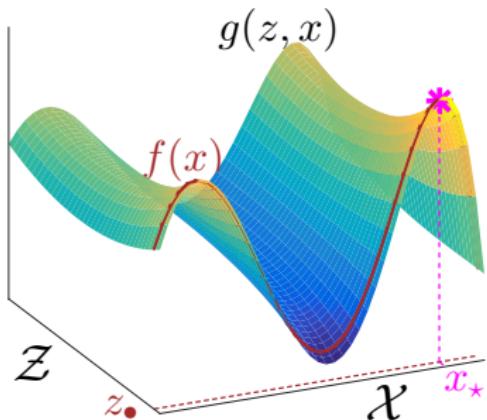
$z_• =$  highest grid size.

**End Goal:** Find  $x_* = \operatorname{argmax}_x f(x)$ .

Therefore,  $\lambda(f, D_t) = f(x_*) - \max_{t: z_t = z_•} f(x_t)$ .

# Multi-fidelity Optimisation

(Kandasamy et al. ICML 2017)



A fidelity space  $\mathcal{Z}$  and domain  $\mathcal{X}$

$\mathcal{Z} \leftarrow$  all granularity values

$\mathcal{X} \leftarrow$  space of cosmological parameters

$$g : \mathcal{Z} \times \mathcal{X} \rightarrow \mathbb{R}.$$

$g(z, x) \leftarrow$  likelihood when performing integrations on a grid of size  $z$  at cosmological parameters  $x$ .

Denote  $f(x) = g(z_•, x)$  where  $z_• \in \mathcal{Z}$ .

$z_• =$  highest grid size.

**End Goal:** Find  $x_* = \operatorname{argmax}_x f(x)$ .

Therefore,  $\lambda(f, D_t) = f(x_*) - \max_{t: z_t = z_•} f(x_t)$ .

A cost function,  $\gamma : \mathcal{Z} \rightarrow \mathbb{R}_+$ .

$\gamma(z) = \mathcal{O}(z^3)$  (say).

# Algorithms

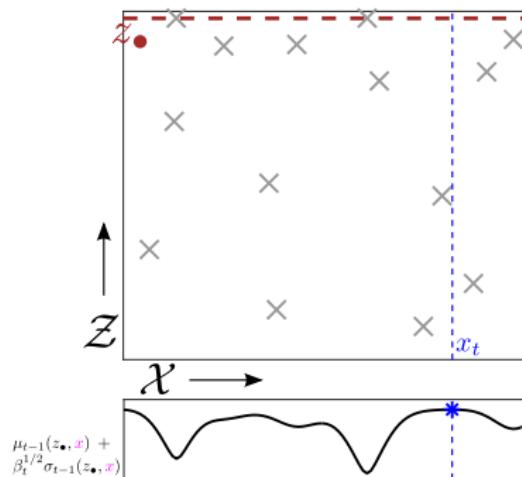
- ▶ Finite number of approximations: MF-GP-UCB  
(Kandasamy et al. NIPS 2016b)
- ▶ Continuous approximations: BOCA  
(Kandasamy et al. ICML 2017)

## Key intuition in both algorithms

- ▶ By default, will evaluate at the low (cheap) fidelities.
- ▶ Proceed to higher (expensive) fidelities when there is a good information to cost trade-off.

# Algorithm: BOCA

(Kandasamy et al. ICML 2017)



Model  $g \sim \mathcal{GP}(0, \kappa)$  and compute posterior  $\mathcal{GP}$ :

mean  $\mu_{t-1} : \mathcal{Z} \times \mathcal{X} \rightarrow \mathbb{R}$

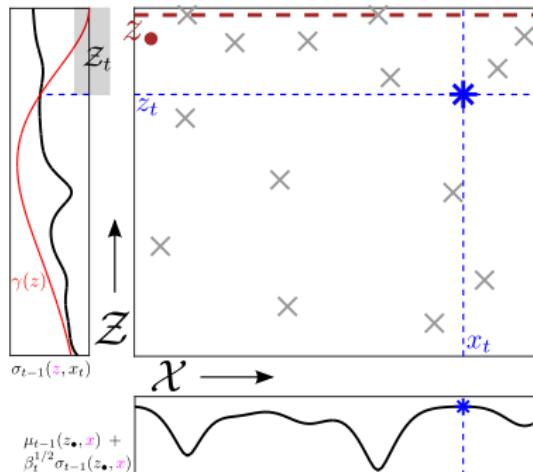
std-dev  $\sigma_{t-1} : \mathcal{Z} \times \mathcal{X} \rightarrow \mathbb{R}_+$

(1)  $x_t \leftarrow$  maximise upper confidence bound for  $f(x) = g(z_{\bullet}, x)$ .

$$x_t = \operatorname{argmax}_{x \in \mathcal{X}} \mu_{t-1}(z_{\bullet}, x) + \beta_t^{1/2} \sigma_{t-1}(z_{\bullet}, x)$$

# Algorithm: BOCA

(Kandasamy et al. ICML 2017)



Model  $g \sim \mathcal{GP}(0, \kappa)$  and compute posterior  $\mathcal{GP}$ :

$$\text{mean} \quad \mu_{t-1} : \mathcal{Z} \times \mathcal{X} \rightarrow \mathbb{R}$$

$$\text{std-dev} \quad \sigma_{t-1} : \mathcal{Z} \times \mathcal{X} \rightarrow \mathbb{R}_+$$

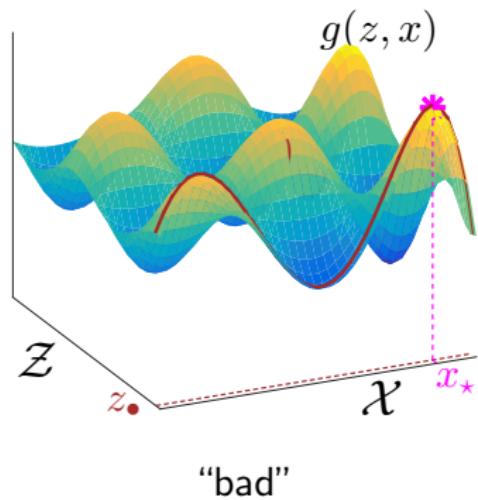
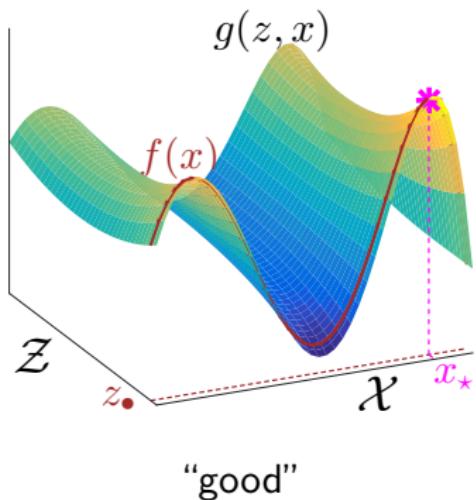
**(1)**  $x_t \leftarrow \text{maximise upper confidence bound for } f(x) = g(z_\bullet, x)$ .

$$x_t = \operatorname{argmax}_{x \in \mathcal{X}} \mu_{t-1}(z_\bullet, \textcolor{magenta}{x}) + \beta_t^{1/2} \sigma_{t-1}(z_\bullet, \textcolor{magenta}{x})$$

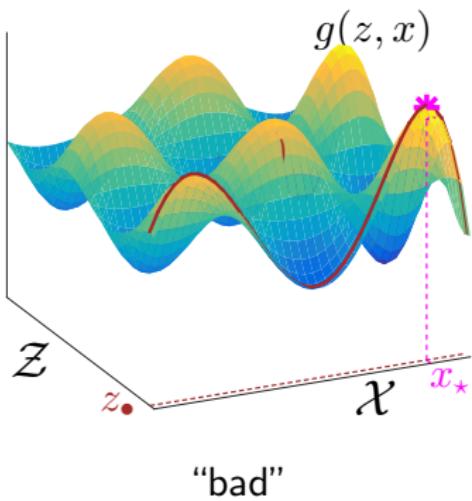
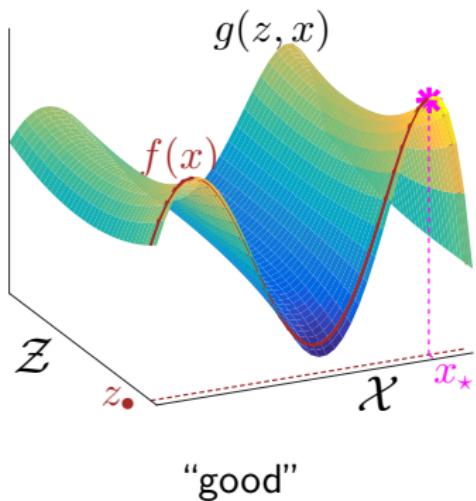
**(2)**  $\mathcal{Z}_t \approx \{z_\bullet\} \cup \left\{ \textcolor{magenta}{z} : \sigma_{t-1}(\textcolor{magenta}{z}, x_t) \geq \gamma(\textcolor{magenta}{z}) = \left( \frac{\gamma(z)}{\gamma(z_\bullet)} \right)^q \xi(z) \right\}$

**(3)**  $z_t = \operatorname{argmin}_{z \in \mathcal{Z}_t} \gamma(\textcolor{magenta}{z}) \quad (\text{cheapest } z \text{ in } \mathcal{Z}_t)$

# Theoretical Results for BOCA



# Theoretical Results for BOCA



**Theorem:** (Informal)

(Kandasamy et al. ICML 2017)

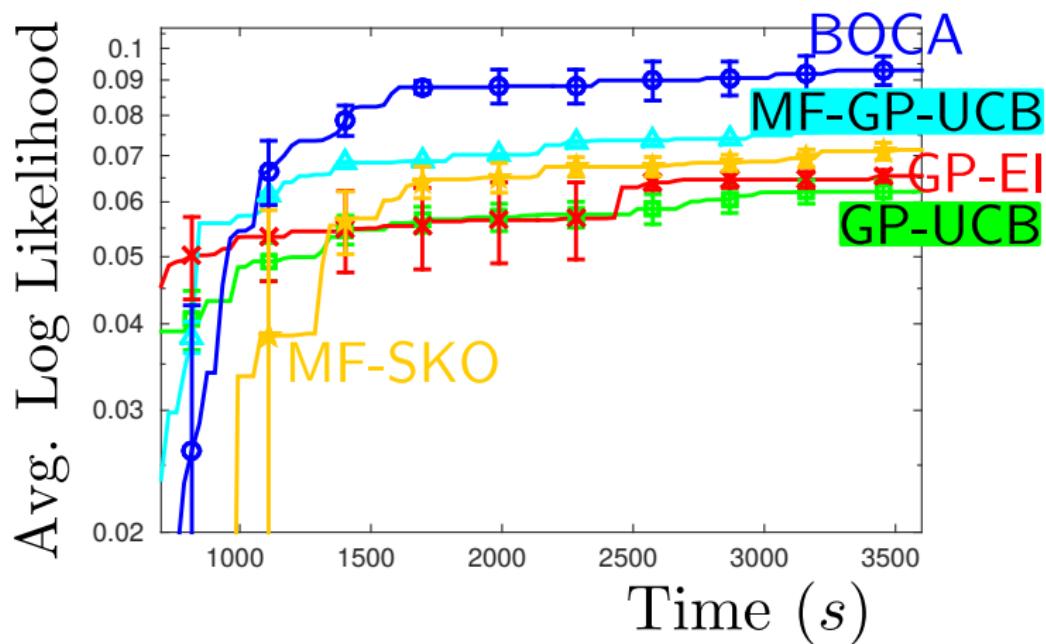
BOCA does better, i.e. achieves better Simple regret, than GP-UCB. The improvements are better in the "good" setting when compared to the "bad" setting.

## Experiment: Cosmological inference on Type-1a supernovae data

Estimate Hubble constant, dark matter fraction & dark energy fraction by maximising likelihood on  $N_\bullet = 192$  data.

Requires numerical integration on a grid of size  $G_\bullet = 10^6$ .

Approximate with  $N \in [50, 192]$  or  $G \in [10^2, 10^6]$  (2D fidelity space).

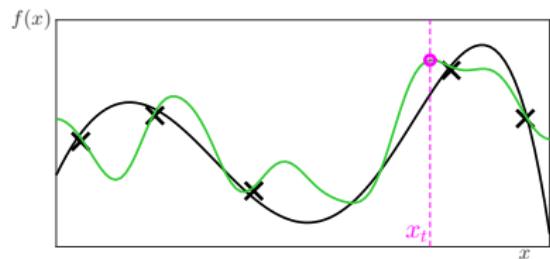
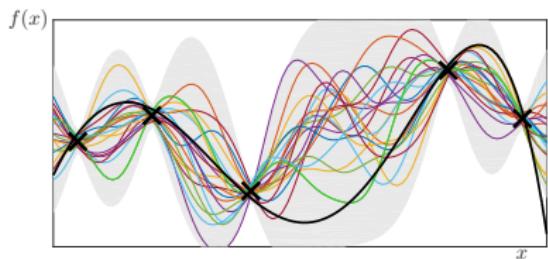


# Outline

- ▶ Part I: Preliminaries (Black-box Optimisation)
  1. Bayesian Models
  2. Black-box Optimisation via Thompson Sampling
- ▶ Part II: DOE via posterior sampling
- ▶ Part III: Scaling up DOE (back to Black-box Optimisation)
  1. Parallelising experiments
  2. Multi-fidelity experimentation
  3. High dimensional input spaces
  4. Beyond Euclidean/categorical domains

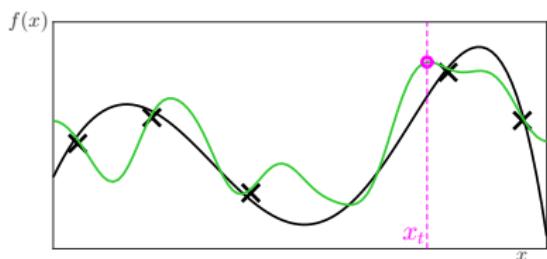
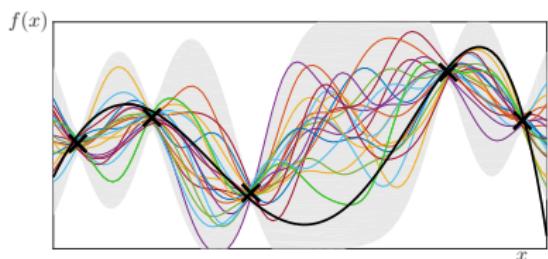
## Part 3.3: Optimisation in High Dimensional Input Spaces

At each time step



## Part 3.3: Optimisation in High Dimensional Input Spaces

At each time step



- 1. Statistical Difficulty:** estimating a high dimensional GP.
- 2. Computational Difficulty:** maximising a high dimensional acquisition (e.g. sample or UCB)  $\varphi_t$ .

# Additive Models for High Dimensional BO

(Kandasamy et al. ICML 2015)

**Structural assumption:**

$$f(x) = f^{(1)}(x^{(1)}) + f^{(2)}(x^{(2)}) + \dots + f^{(M)}(x^{(M)}).$$

$$x^{(j)} \in \mathcal{X}^{(j)} = [0, 1]^p, \quad p \ll d, \quad x^{(i)} \cap x^{(j)} = \emptyset.$$

# Additive Models for High Dimensional BO

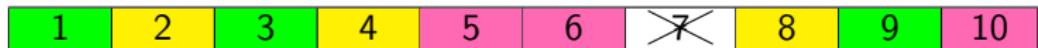
(Kandasamy et al. ICML 2015)

**Structural assumption:**

$$f(x) = f^{(1)}(x^{(1)}) + f^{(2)}(x^{(2)}) + \dots + f^{(M)}(x^{(M)}).$$

$$x^{(j)} \in \mathcal{X}^{(j)} = [0, 1]^p, \quad p \ll d, \quad x^{(i)} \cap x^{(j)} = \emptyset.$$

E.g.  $f(x_{\{1, \dots, 10\}}) = f^{(1)}(x_{\{1, 3, 9\}}) + f^{(2)}(x_{\{2, 4, 8\}}) + f^{(3)}(x_{\{5, 6, 10\}})$ .



Call  $\{\mathcal{X}^{(j)}\}_{j=1}^M = \{(1, 3, 9), (2, 4, 8), (5, 6, 10)\}$  the “decomposition”.

# Additive Models for High Dimensional BO

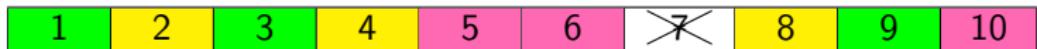
(Kandasamy et al. ICML 2015)

## Structural assumption:

$$f(x) = f^{(1)}(x^{(1)}) + f^{(2)}(x^{(2)}) + \dots + f^{(M)}(x^{(M)}).$$

$$x^{(j)} \in \mathcal{X}^{(j)} = [0, 1]^p, \quad p \ll d, \quad x^{(i)} \cap x^{(j)} = \emptyset.$$

E.g.  $f(x_{\{1, \dots, 10\}}) = f^{(1)}(x_{\{1, 3, 9\}}) + f^{(2)}(x_{\{2, 4, 8\}}) + f^{(3)}(x_{\{5, 6, 10\}})$ .



Call  $\{\mathcal{X}^{(j)}\}_{j=1}^M = \{(1, 3, 9), (2, 4, 8), (5, 6, 10)\}$  the “decomposition”.

## Advantages:

- ▶ Statistical: Better bias-variance trade-offs in high dimensions.
- ▶ Computational: Easy to maximise acquisition and choose  $x_t$ .

Upper Confidence Bound:

$$\varphi_t(x) = \sum_{j=1}^M \underbrace{\mu_{t-1}^{(j)}(x^{(j)}) + \beta_t^{1/2} \sigma_{t-1}^{(j)}(x^{(j)})}_{\tilde{\varphi}_t^{(j)}(x^{(j)})}.$$

Maximise each  $\tilde{\varphi}_t^{(j)}$  separately.

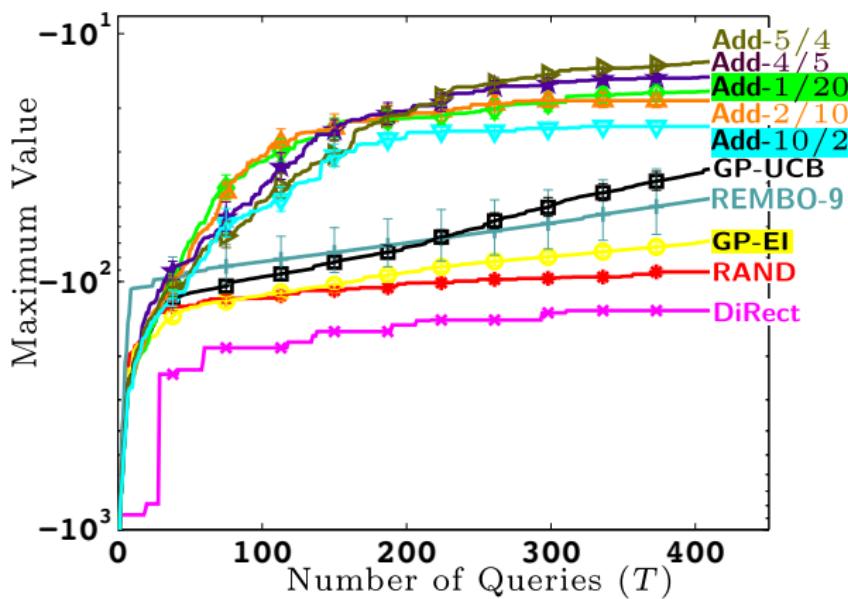
Requires only  $\mathcal{O}(\text{poly}(d)\epsilon^{-p})$  effort (vs  $\mathcal{O}(\epsilon^{-d})$  for GP-UCB).

## Additive models can still be useful in non-additive settings

- ▶ Additive models common in high dimensional regression.  
E.g.: Backfitting, MARS, COSSO, RODEO, SpAM etc.  
(Friedman '91, Lin et al. '06, Lafferty et al '05, Ravikumar et al. '09)
  - ▶ Additive models are *statistically* simpler  $\implies$  worse bias, but much better variance in low sample regime.
  - ▶ In bandit applications queries are *expensive*. So we usually cannot afford many queries.
- 
- ▶ **Observation:**  
Add-GP-UCB does well even when  $f$  is not additive.
    - ▶ Better bias/ variance trade-off in estimating the GP.
    - ▶ Easy to maximise upper confidence bound.

# Experiment: Cosmological Inference

Estimating 20 cosmological parameters from data on luminous red galaxies.

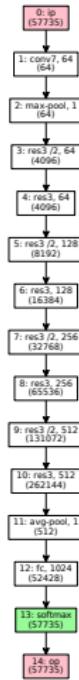


In the paper we go up to  $> 100$  dimensions.

# Outline

- ▶ Part I: Preliminaries (Black-box Optimisation)
  1. Bayesian Models
  2. Black-box Optimisation via Thompson Sampling
- ▶ Part II: DOE via posterior sampling
- ▶ Part III: Scaling up DOE (back to Black-box Optimisation)
  1. Parallelising experiments
  2. Multi-fidelity experimentation
  3. High dimensional input spaces
  4. Beyond Euclidean/categorical domains

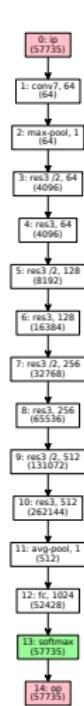
## Part 3.4: Tuning Neural Network Architectures



Feedforward  
network

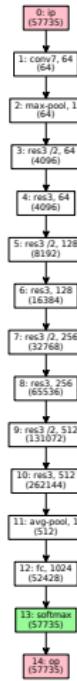
## Part 3.4: Tuning Neural Network Architectures

Feedforward  
network

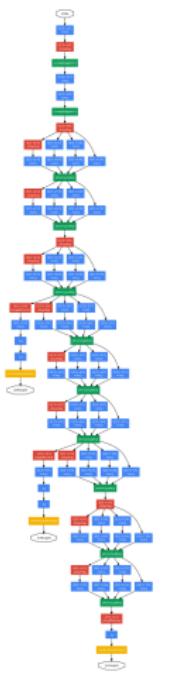


GoogLeNet  
(Szegedy et  
al. 2015)

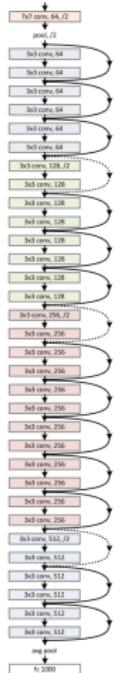
## Part 3.4: Tuning Neural Network Architectures



# Feedforward network



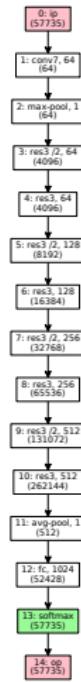
## GoogLeNet (Szegedy et al. 2015)



# ResNet (He et al. 2016)

## Part 3.4: Tuning Neural Network Architectures

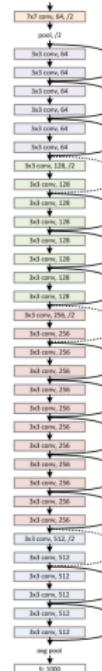
Feedforward network



GoogLeNet  
(Szegedy et al. 2015)



ResNet  
(He et al. 2016)

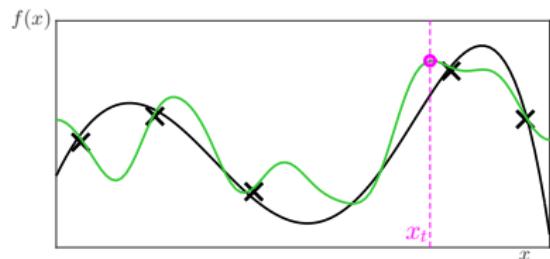
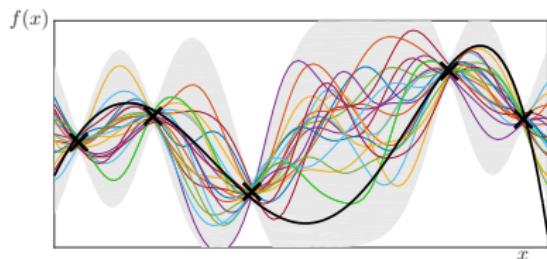


DenseNet  
(Huang et al. 2017)



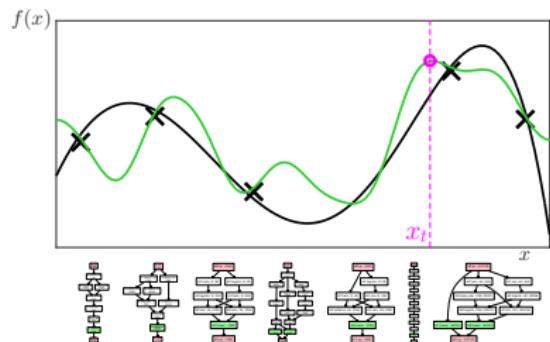
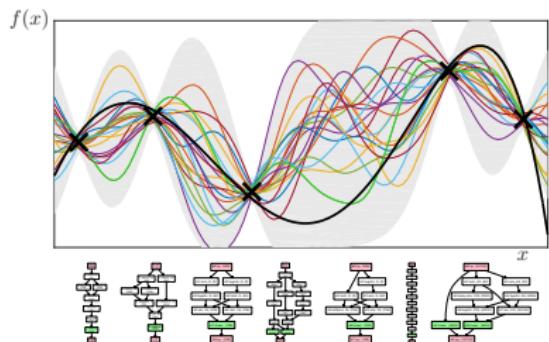
## Part 3.4: Tuning Neural Network Architectures

At each time step



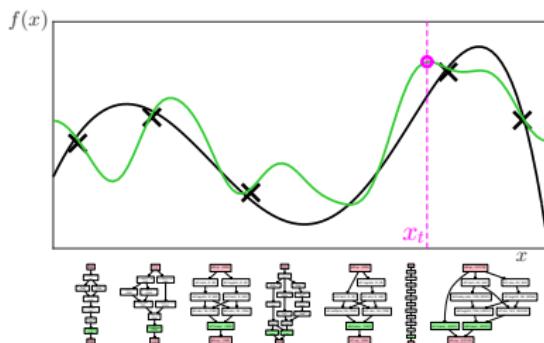
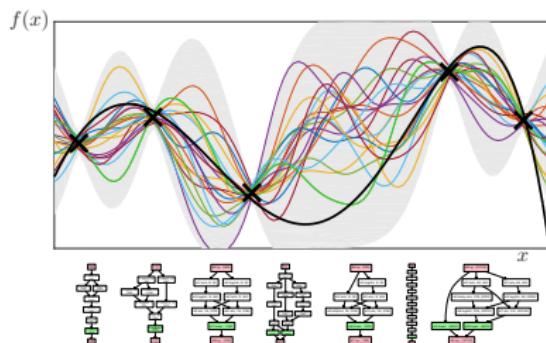
## Part 3.4: Tuning Neural Network Architectures

At each time step



## Part 3.4: Tuning Neural Network Architectures

At each time step



### Main challenges

- ▶ Define a distance between neural network architectures.
- ▶ Optimise  $\varphi_t$  on the space of neural networks.

# OTMANN: A distance between Neural Architectures

(Kandasamy et al. Arxiv 2018)

**Key idea:** To compute distance between architectures  $G_1$ ,  $G_2$ ,  
match computation in layers in  $G_1$  to  $G_2$ .

$$Z \in \mathbb{R}^{n_1 \times n_2}.$$

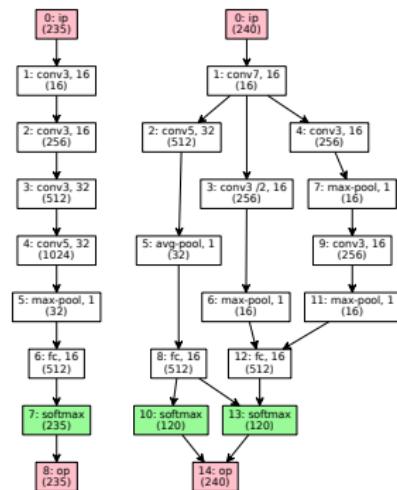
$Z_{ij} \leftarrow$  amount matched between layer  
 $i \in G_1$  and  $j \in G_2$ .

Minimise  $\phi_{lmm}(Z) + \phi_{str}(Z) + \phi_{nas}(Z)$

$\phi_{lmm}(Z)$ : label mismatch penalty

$\phi_{str}(Z)$  : structural penalty

$\phi_{nas}(Z)$  : non-assignment penalty



# Optimising the sample from the posterior

Via an evolutionary algorithm.

Operation	Description
dec_single	Pick a layer at random and decrease the number of units by $1/8$ .
dec_en_masse	Pick several layers at random in topological order and decrease the number of units by $1/8$ for all of them.
inc_single	Pick a layer at random and increase the number of units by $1/8$ .
inc_en_masse	Pick several layers at random in topological order and increase the number of units by $1/8$ for all of them.
dup_path	Pick a random path $u_1, u_2, \dots, u_{k-1}, u_k$ , duplicate layers $u_2, \dots, u_{k-1}$ and connect them to $u_1$ and $u_k$ .
remove_layer	Pick a layer at random and remove it. Connect the layer's parents to its children if necessary.
skip	Randomly pick layers $u, v$ where $u$ is topologically before $v$ . Add $(u, v)$ to $\mathcal{E}$ .
swap_label	Randomly pick a layer and change its label.
wedge_layer	Randomly remove an edge $(u, v)$ from $\mathcal{E}$ . Create a new layer $w$ and add $(u, w), (w, v)$ to $\mathcal{E}$ .

# Optimising the sample from the posterior

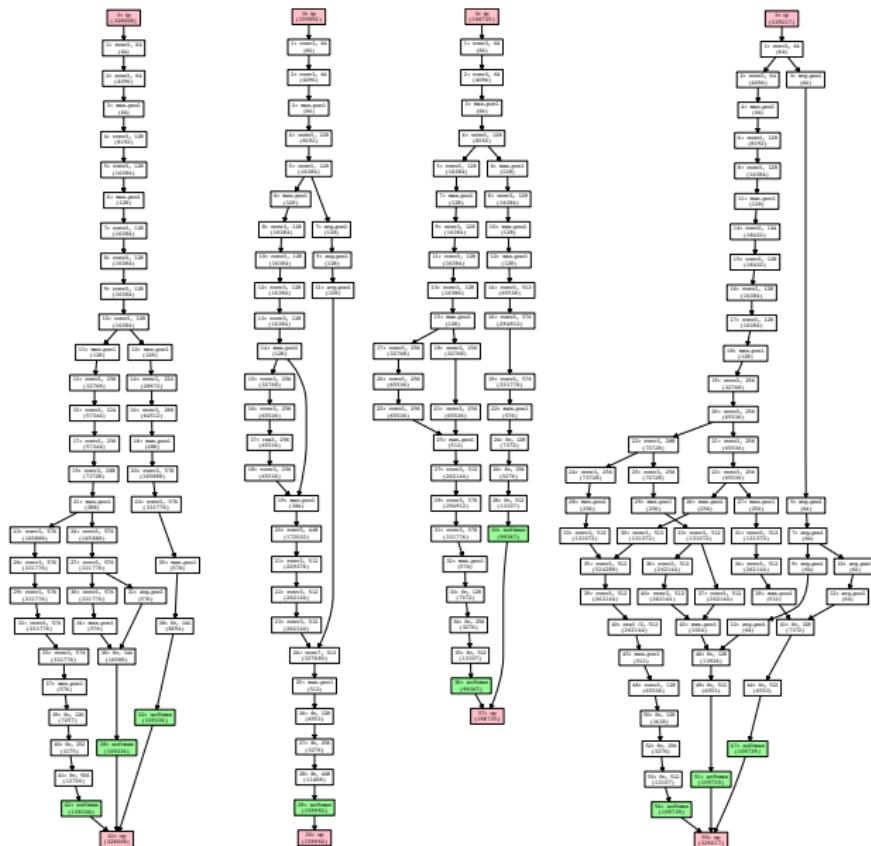
Via an evolutionary algorithm.

Operation	Description
dec_single	Pick a layer at random and decrease the number of units by $1/8$ .
dec_en_masse	Pick several layers at random in topological order and decrease the number of units by $1/8$ for all of them.
inc_single	Pick a layer at random and increase the number of units by $1/8$ .
inc_en_masse	Pick several layers at random in topological order and increase the number of units by $1/8$ for all of them.
dup_path	Pick a random path $u_1, u_2, \dots, u_{k-1}, u_k$ , duplicate layers $u_2, \dots, u_{k-1}$ and connect them to $u_1$ and $u_k$ .
remove_layer	Pick a layer at random and remove it. Connect the layer's parents to its children if necessary.
skip	Randomly pick layers $u, v$ where $u$ is topologically before $v$ . Add $(u, v)$ to $\mathcal{E}$ .
swap_label	Randomly pick a layer and change its label.
wedge_layer	Randomly remove an edge $(u, v)$ from $\mathcal{E}$ . Create a new layer $w$ and add $(u, w), (w, v)$ to $\mathcal{E}$ .

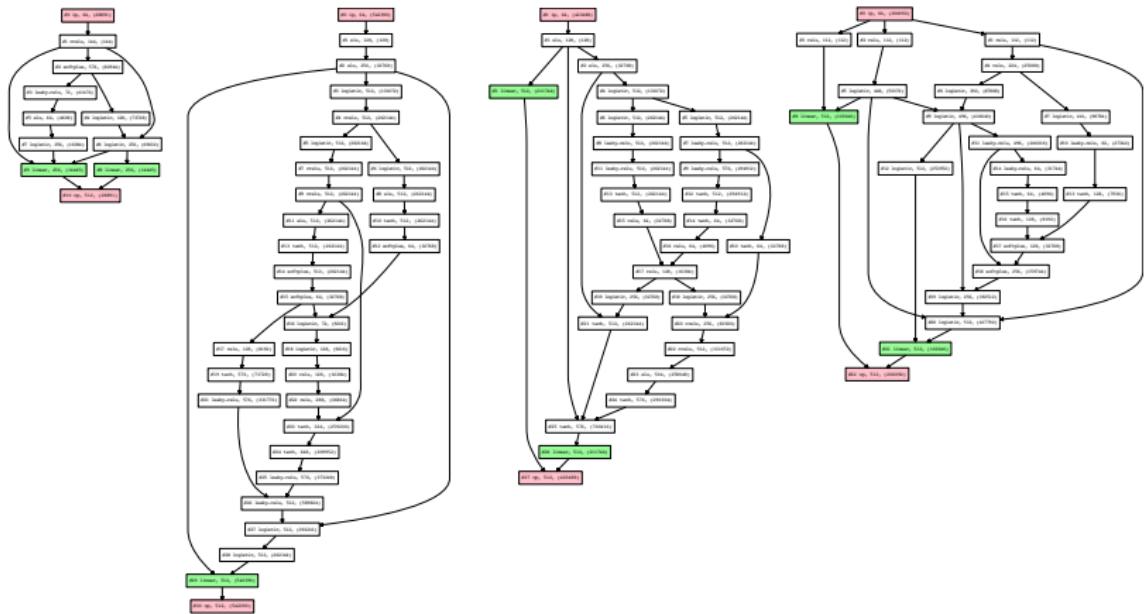
Resulting procedure: NASBOT

Neural Architecture Search with Bayesian Optimisation and  
Optimal Transport  
*(Kandasamy et al. Arxiv 2018)*

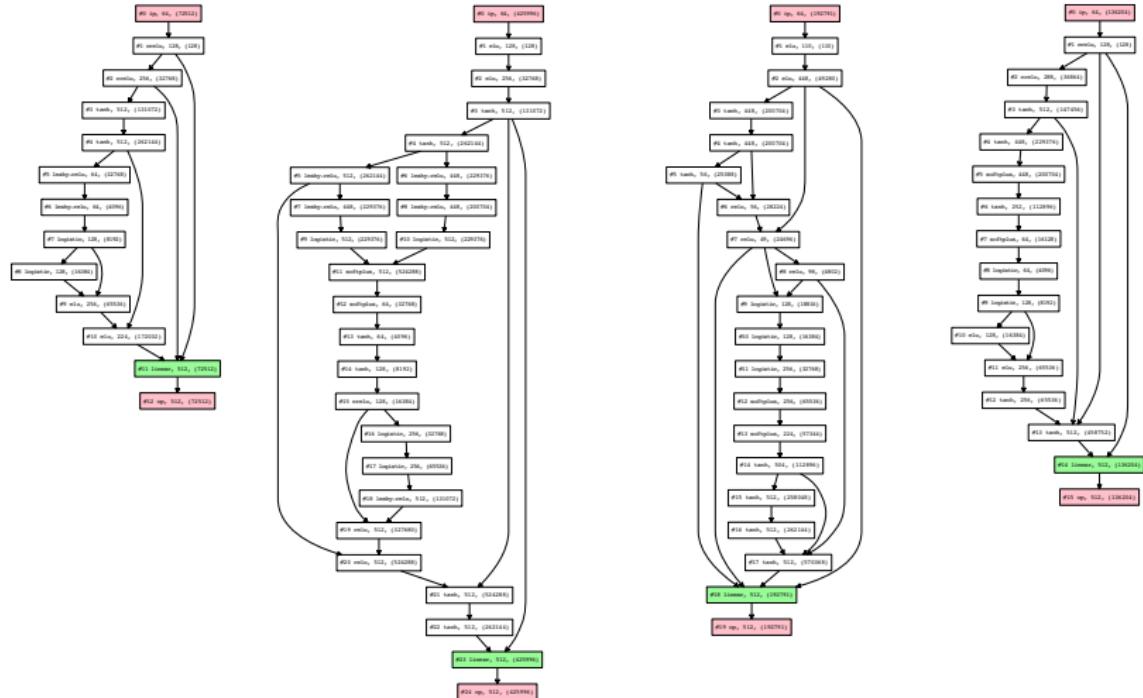
## Architectures found on Cifar10



# Architectures found on Indoor Location

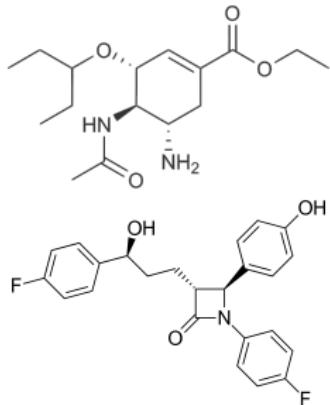


# Architectures found on Slice Localisation

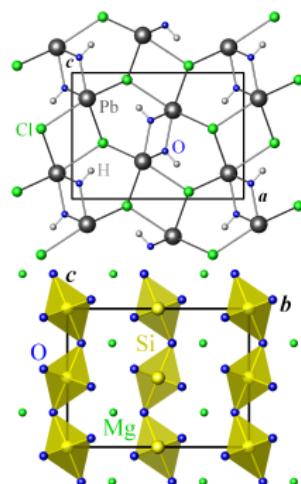


# DOE on other (graphical) structures

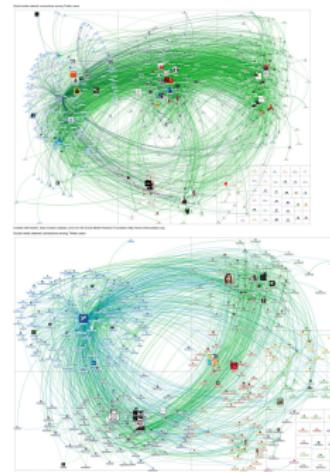
## Drug Discovery with Small molecules



## Crystal Structures



## Social networks & viral marketing



## Summary

- ▶ A framework for “goal oriented” DOE.
  - General: can achieve any desired goal.
  - Flexible: can incorporate domain expertise.
- ▶ Myopic Posterior Sampling (MPS): An algorithm for DOE inspired by Thompson sampling.
- ▶ Can be trivially parallelised.

## Summary

- ▶ A framework for “goal oriented” DOE.
  - General: can achieve any desired goal.
  - Flexible: can incorporate domain expertise.
- ▶ Myopic Posterior Sampling (MPS): An algorithm for DOE inspired by Thompson sampling.
- ▶ Can be trivially parallelised.

## Scaling up DOE

- ▶ Multi-fidelity experimentation: Use cheap approximations to a an expensive experiment to speed things up.
- ▶ High dimensional DOE: Additive models have favourable statistical and computational properties.
- ▶ DOE in “complex” domains.



Akshay



Barnabás



Biswajit



Chun-liang



Eric



Gautam



Jeff



Junier



Karun



Rajat



Reed



Sanjay



Shalom



Shuli



Willie

CMU, University of Massachusetts Amherst, Rice University,  
University of Texas Austin, Microsoft Research, ExperiML

Thank You