# Bayesian Optimization

**Nando de Freitas**

Eric Brochu, Ruben Martinez-Cantin, Matt Hoffman, Ziyu Wang, ...

# More resources

This talk follows the presentation in the following review paper available fr
Oxford website.

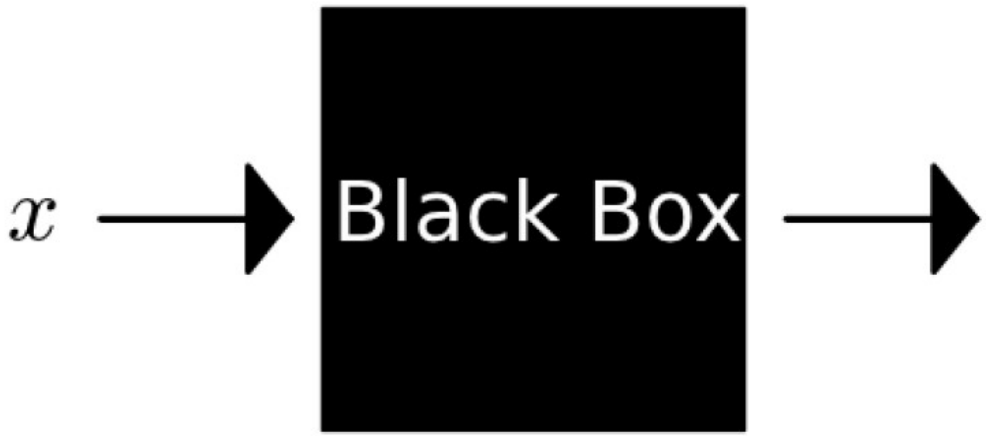
Taking the Human Out of the Loop:
A Review of Bayesian Optimization

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams and Nando de Freitas

# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

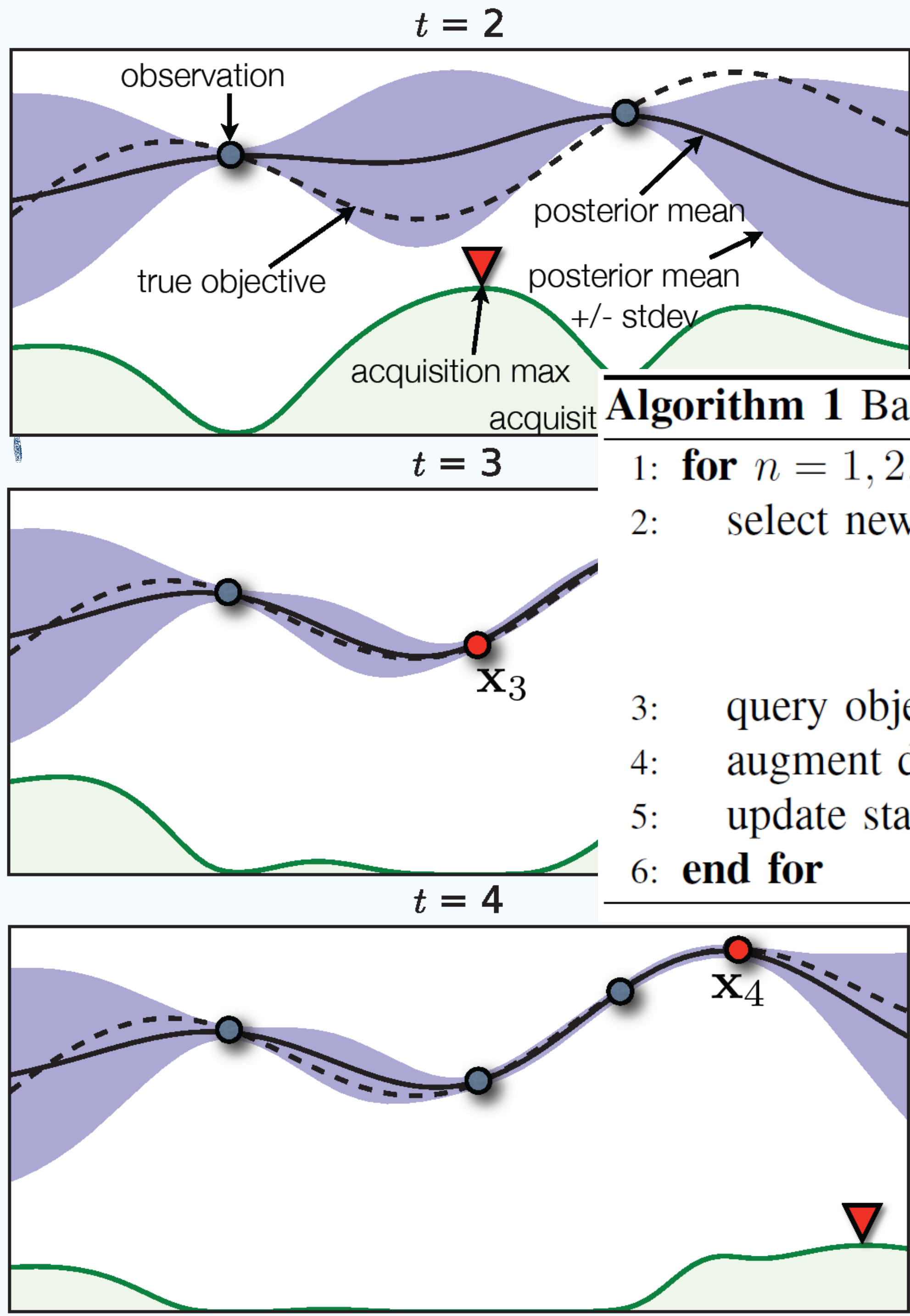
# Black-box optimization / design




$f(x) + \epsilon$

$$\mathbb{E}[y \mid f(\mathbf{x})] = f(\mathbf{x})$$

$$\mathbf{x}^\star = \arg\max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$$




**Algorithm 1** Bayesian optimization

**for** $n = 1, 2, \ldots$ **do**

  select new $\mathbf{x}_{n+1}$ by optimizing acquisition function $\alpha$

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

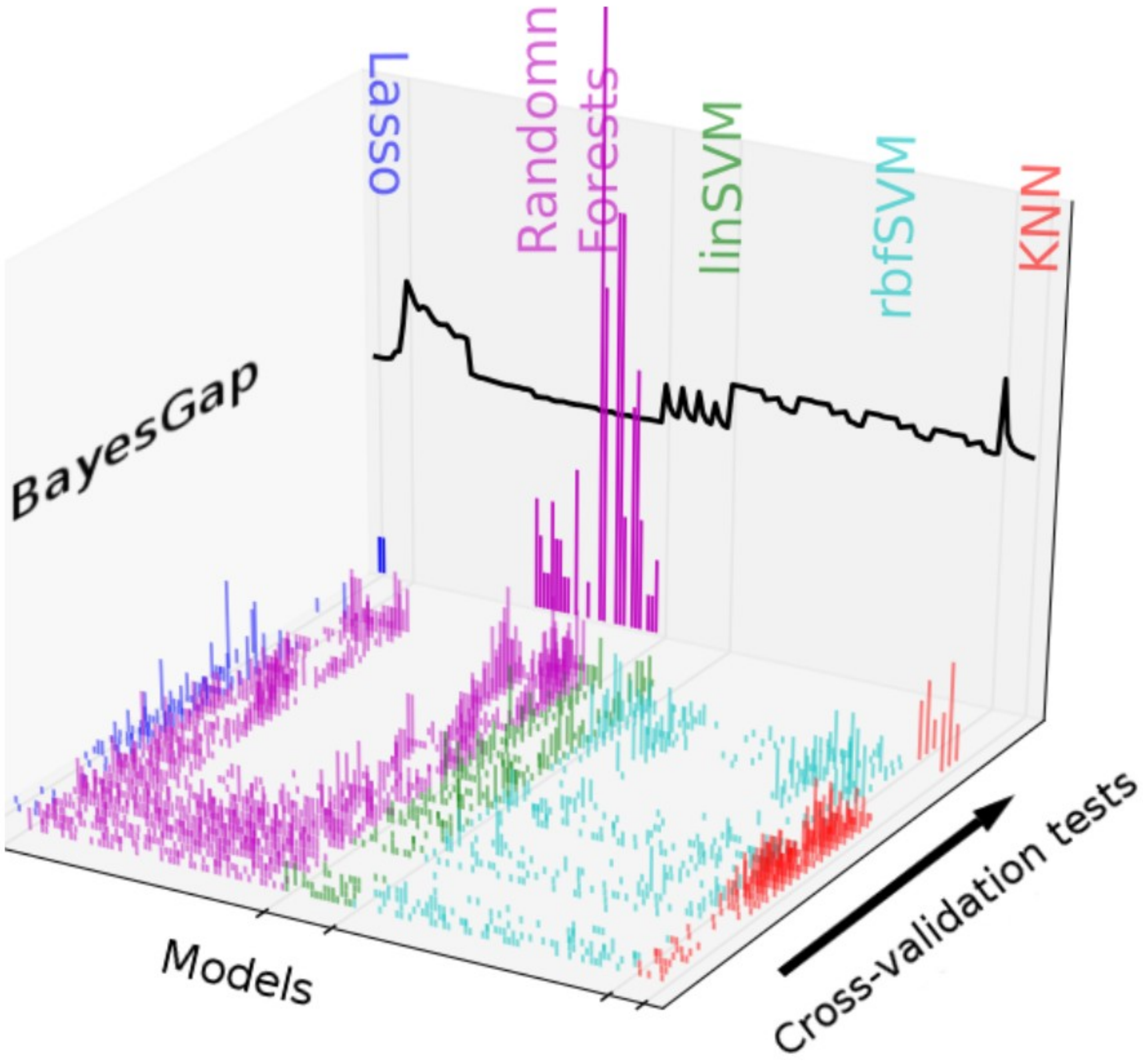
  query objective function to obtain $y_{n+1}$

  augment data $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (\mathbf{x}_{n+1}, y_{n+1})\}$

  update statistical model

**end for**

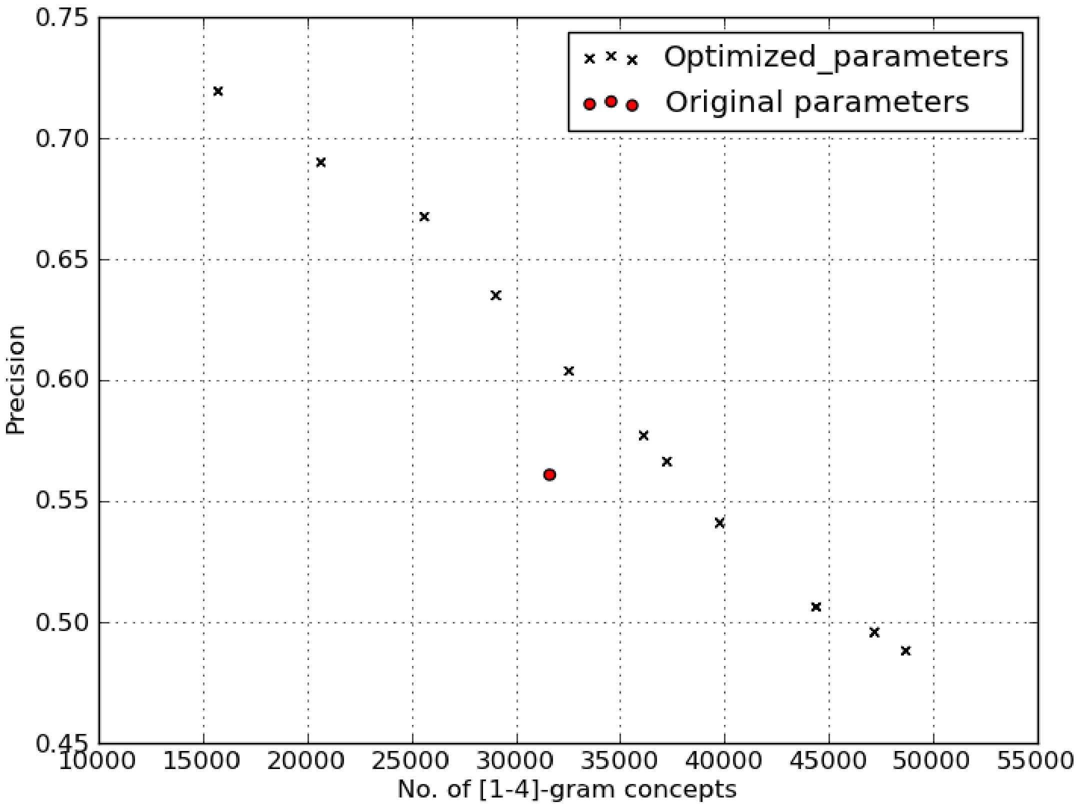
# Automatic machine learning




[Hoffman, Shahriari & dF, 2013]

# Information extraction / concept learning

I love silence of the lambs. It's a scary movie.
The confit de canard was delish!

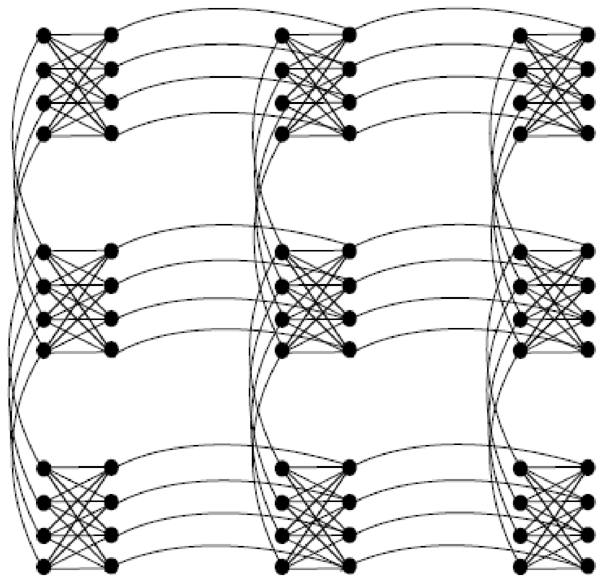


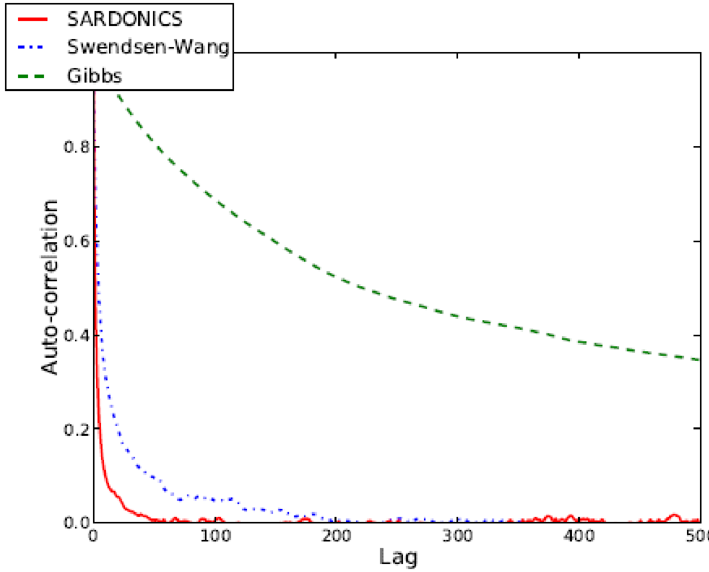

Expert tuning by Hector Garcia-Molina et al: *Towards The Web Of Concepts: Extracting Concepts from Large Datasets*

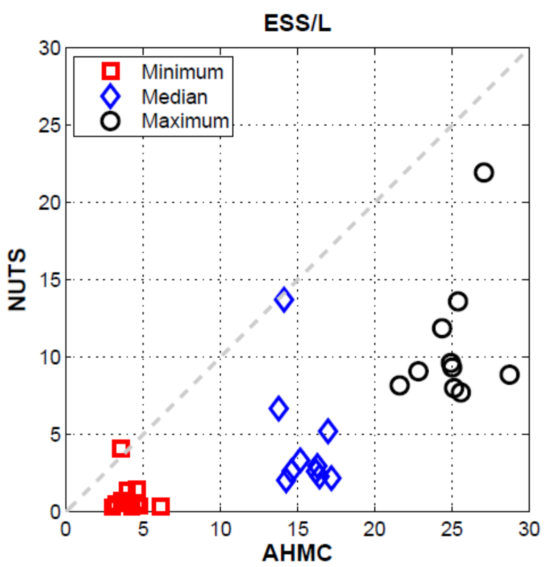
[Ziyu Wang et al, 2014]

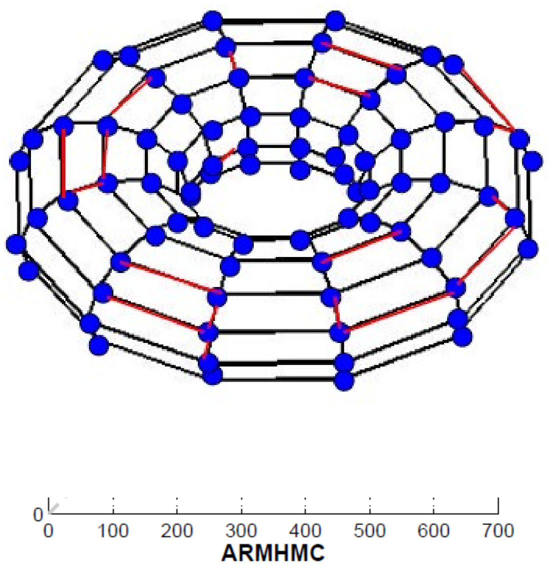
# Automatic (Adaptive) Monte Carlo samplers

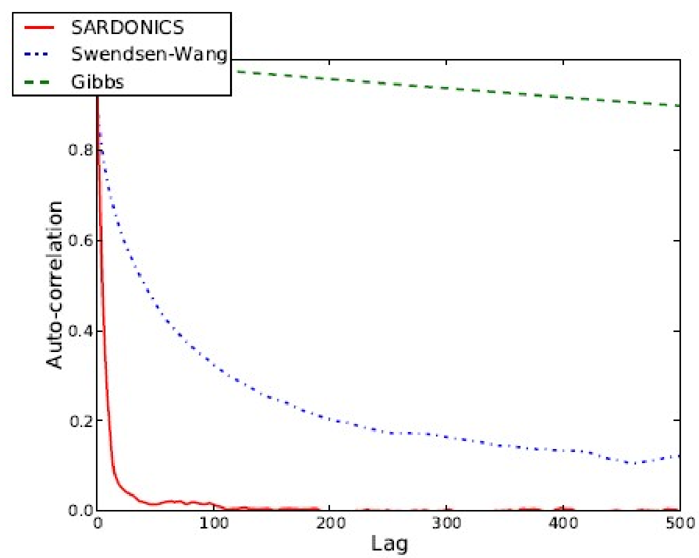
| Method |
|---|
| Rios Insua and Muller's reversible-jump MCMC |
| Mackay's (1992) Gaussia with highest evidence |
| Neal's (1996) HMC |
| Neal's (1996) HMC with |
| Reversible-jump MCM( model by Andrieu et al. |
| Adaptive HMC (Median |
| Adaptive HMC (Mean Err |





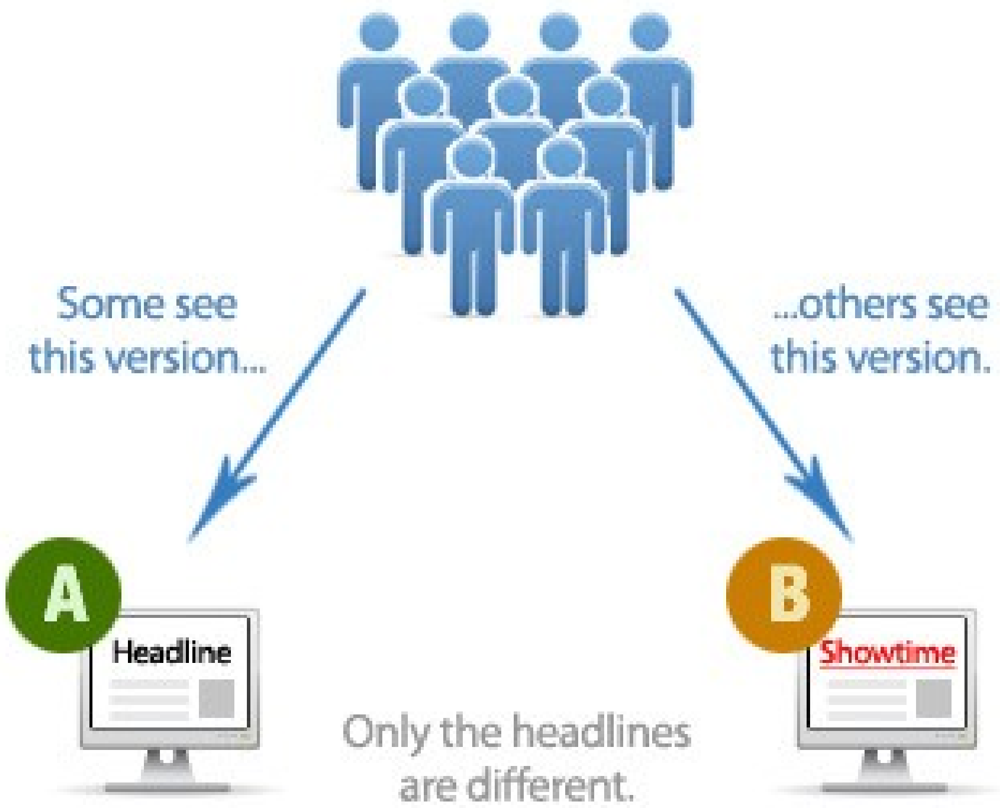




[Wang, Mohamed & dF, 2013]

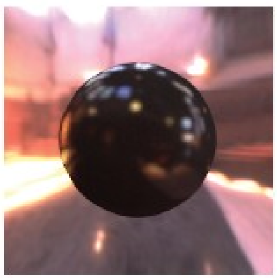
# Analytics, dynamic creative content and A/B testing

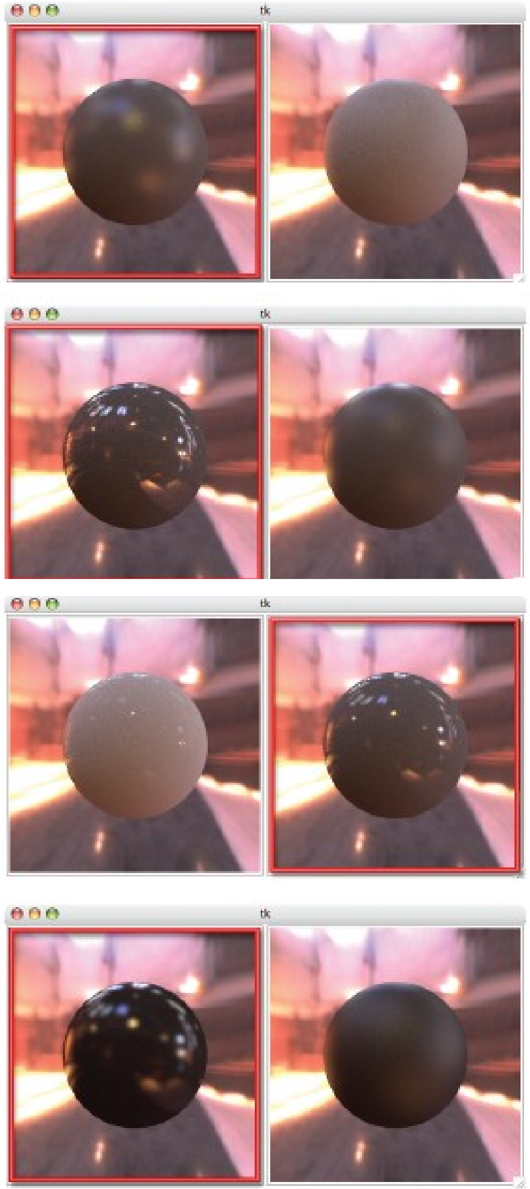



[Steve Scott on Bayesian bandits at Google]

# animation session

# target



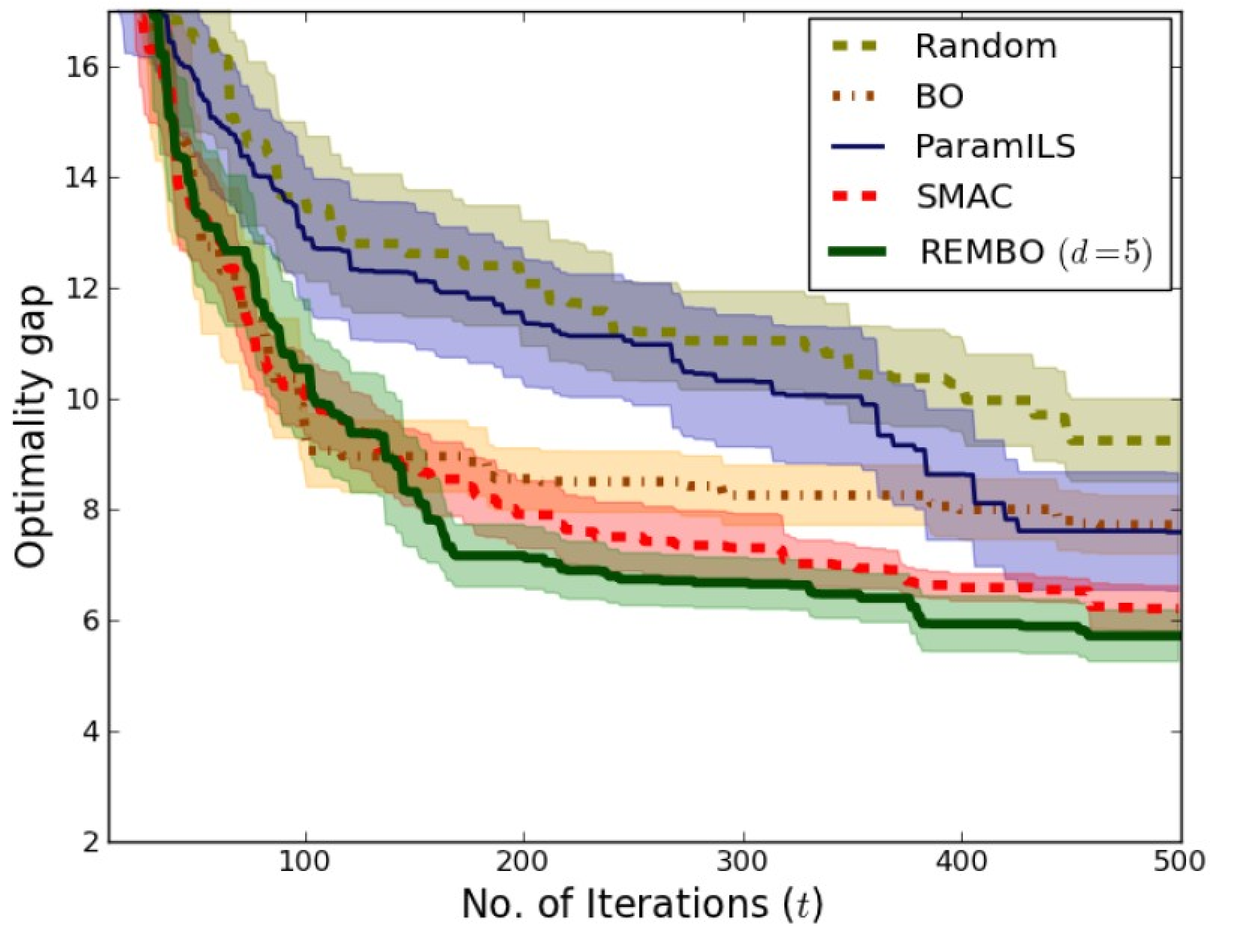


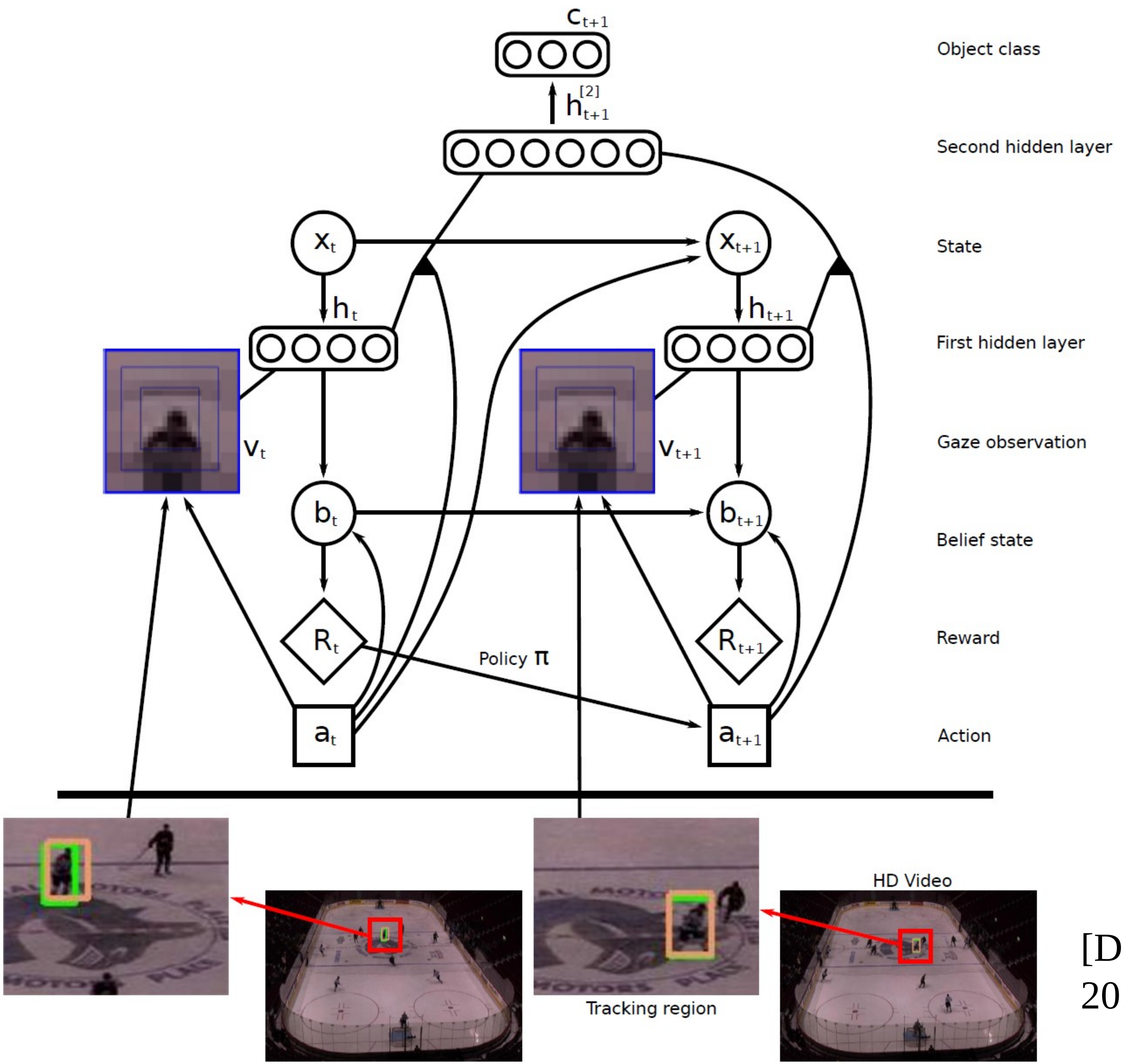
[Brochu, Ghosh & dF, 2007. Brochu, Brochu, dF, 2010]

**Winner of the SRC competition - SIGGRAPH**

# Tuning NP hard problem solvers

- **lpsolve** is a mixed integer programming solver, downloaded over 40,000 times last year.
- 47 discrete parameters (choices)

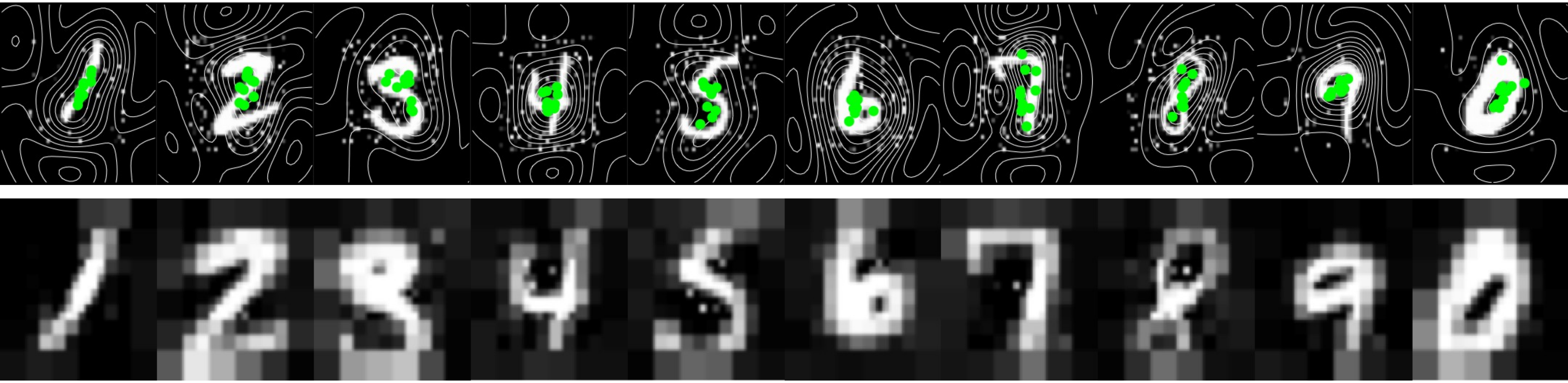




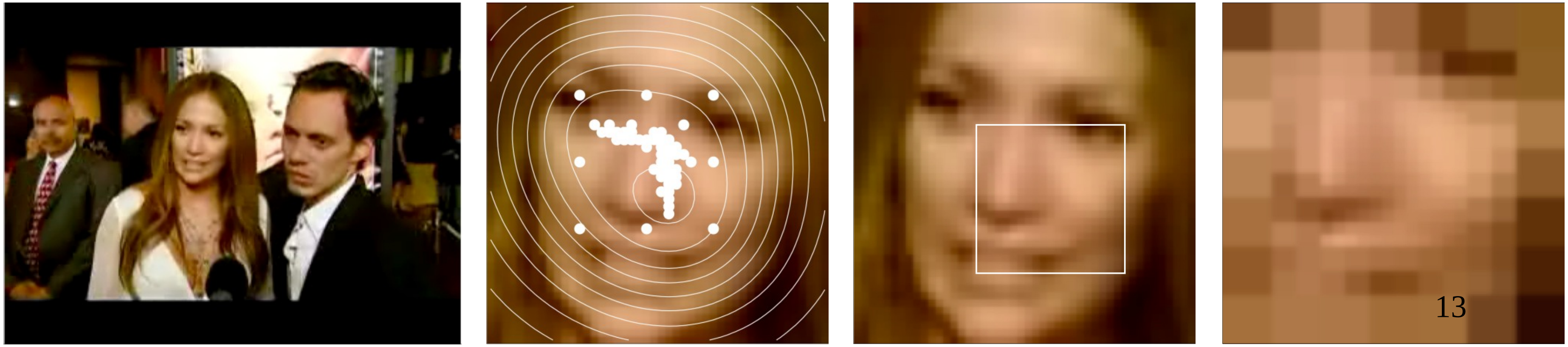

[Denil et al 2012]

# GP Policy for tracking

**Digits Experiment:**



**Face Experiment:**

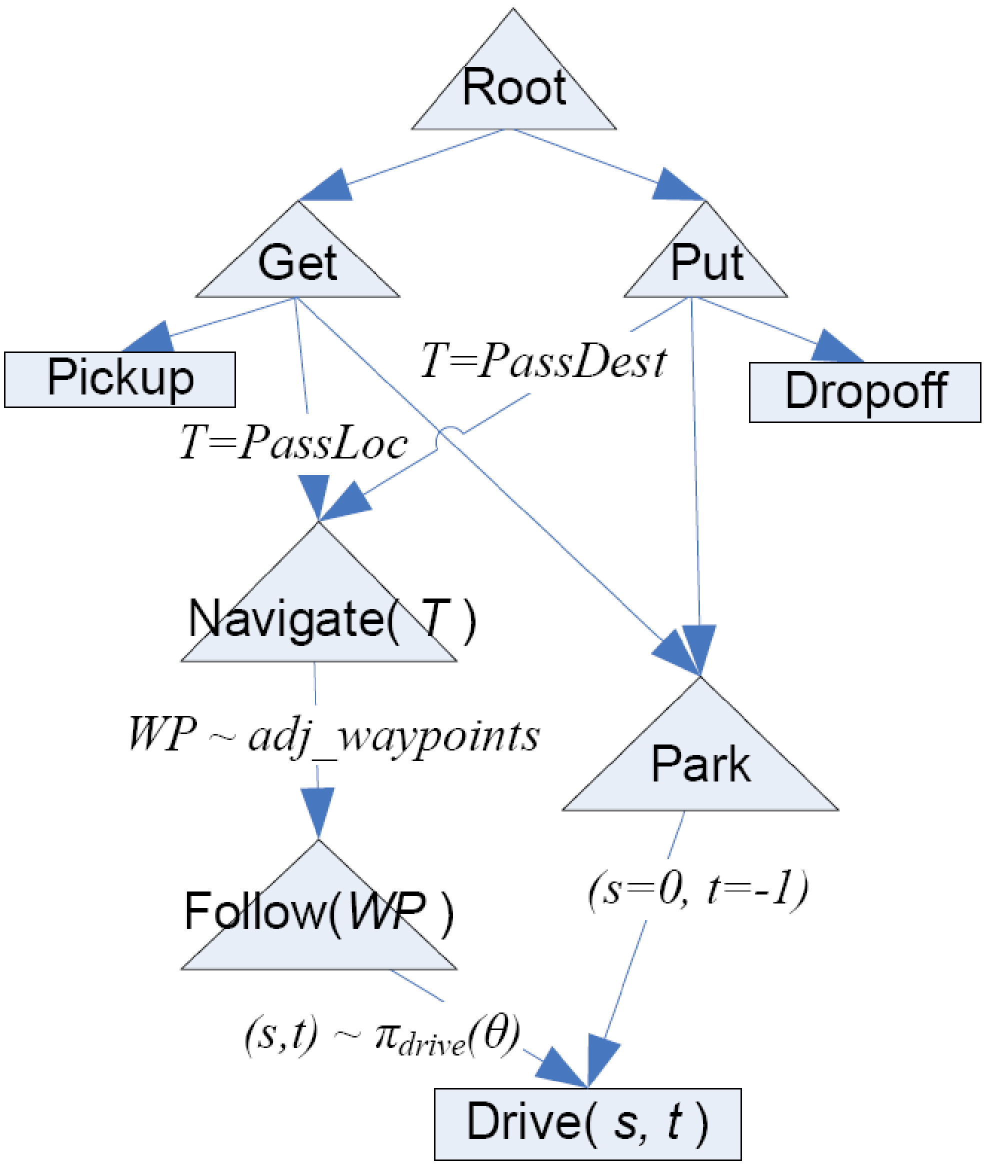


# Hierarchical reinforcement learning

**High-level** model-based learning for deciding when to navigate, park, pickup and dropoff passengers.

**Mid-level** active path learning for navigating a topological map.

**Low-level** active policy optimizer to learn control of continuous non-linear vehicle dynamics.

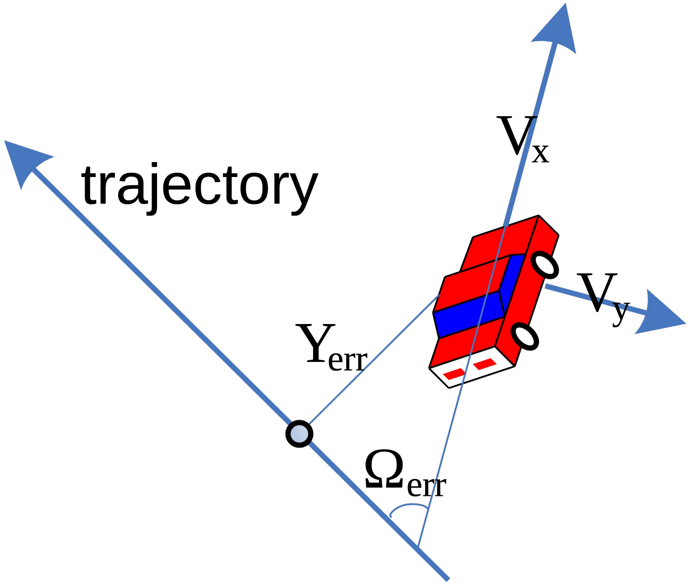


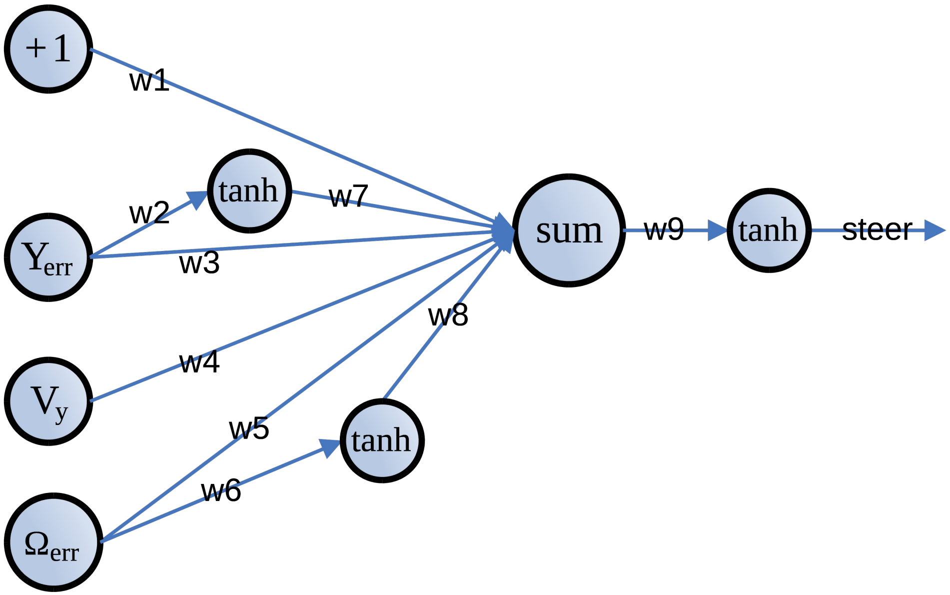
# Active Path Finding in Middle Level

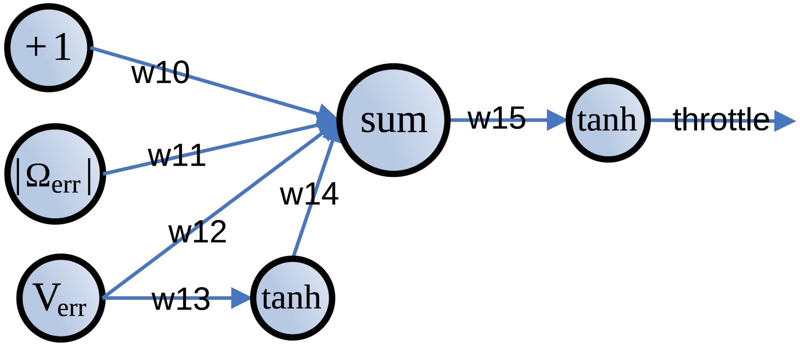
*Navigate* policy generates sequence of waypoints on a topological map to navigate from a location to a destination.

# Low-Level: Trajectory following

TORCS: 3D game engine that implements complex vehicle dynamics complete with manual and automatic transmission, engine, clutch, tire, and suspension models.



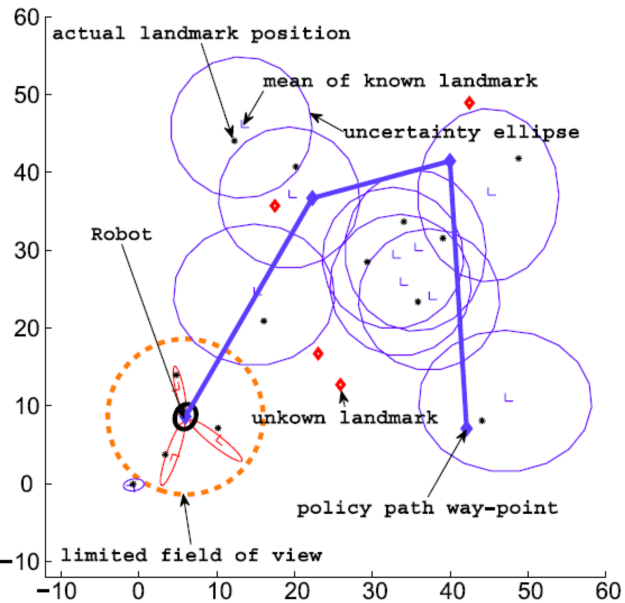



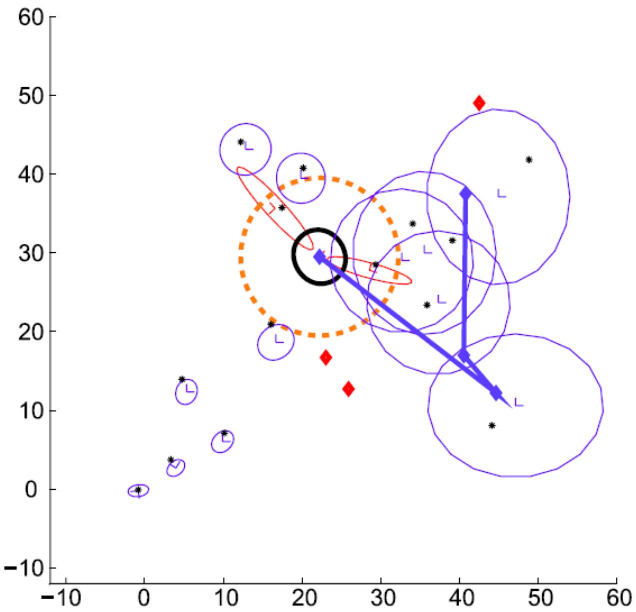



Bayesian optimization was used to find the neural net low-level policy and the value function for the upper levels

# Many other applications

- Robotics, control, reinforcement learning, …
- SAT solvers, scheduling, planning
- Configuration of ad-centers, compilers, hardware, software…
- Programming by optimization
- …



# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Bayesian parametric optimization

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathbf{w})\, p(\mathbf{w})}{p(\mathcal{D})}$$

# Beta-Bernoulli Bayesian optimization

Consider the following K-armed bandit setting.

$$a \in 1, \ldots, K$$

$$y_i \in \{0, 1\}$$

$$\mathcal{D}_n = \{(a_i, y_i)\}_{i=1}^{n}$$

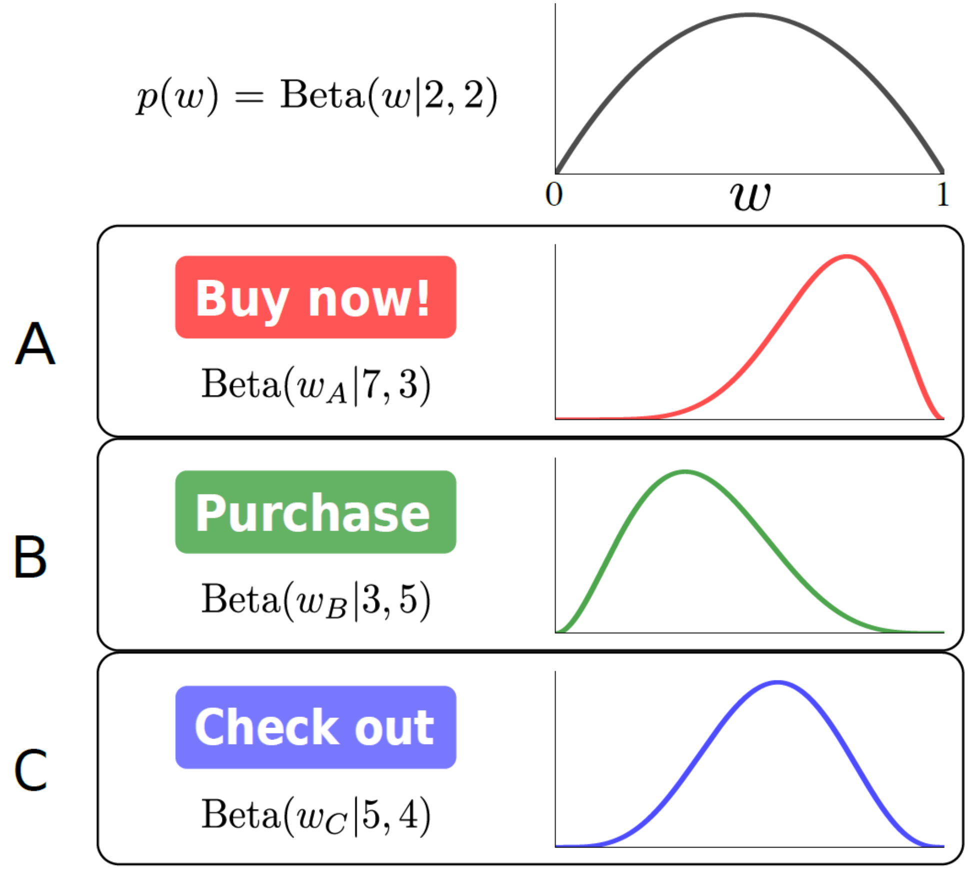
$$\mathbf{w} \in (0, 1)^K$$

$$f_{\mathbf{w}}(a) := w_a$$

# Beta-Bernoulli Bayesian optimization

Specify a Beta prior on each arm:

$$p(\mathbf{w} \mid \alpha, \beta) = \prod_{a=1}^{K} \mathrm{Beta}(w_a \mid \alpha, \beta)$$



# Beta-Bernoulli Bayesian

$$p(\mathbf{w} \mid \alpha, \beta) = \prod_{a=1}^{K} \mathrm{Beta}(w_a \mid \alpha, \beta)$$

$$n_{a,0} = \sum_{i=1}^{n} \mathbb{I}(y_i = 0, a_i = a)$$

$$n_{a,1} = \sum_{i=1}^{n} \mathbb{I}(y_i = 1, a_i = a)$$

$$p(\mathbf{w} \mid \mathcal{D}) = \prod_{a=1}^{K} \mathrm{Beta}(w_a \mid \alpha + n_{a,1}, \beta + n_{a,0})$$

# Thompson sampling

- At each iteration, draw a sample from the poster
- Pick the action of highest expected re $\tilde{w}_a$ 1

$$a_{n+1} = \arg\max_a f_{\tilde{\mathbf{w}}}(a) \quad \text{where } \tilde{\mathbf{w}} \sim p(\mathbf{w} \mid \mathcal{D}_n)$$

# Thompson sampling

**Algorithm 2** Thompson sampling for Beta-Bernoulli bandit

**Require:** $\alpha, \beta$: hyperparameters of the beta prior

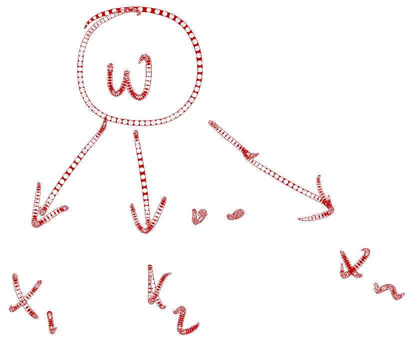
Initialize $n_{a,0} = n_{a,1} = i = 0$ for all $a$
**repeat**
  **for** $a = 1, \ldots, K$ **do**
    $\tilde{w}_a \sim \text{Beta}(\alpha + n_{a,1}, \beta + n_{a,0})$
  **end for**
  $a_i = \arg\max_a \tilde{w}_a$
  Observe $y_i$ by pulling arm $a_i$
  **if** $y_i = 0$ **then**
    $n_{a_i,0} = n_{a_i,0} + 1$
  **else**
    $n_{a_i,1} = n_{a_i,1} + 1$
  **end if**
  $i = i + 1$
**until** stopping criterion reached

# Linear bandits

Introduce correlations among the arms

feature vector $\mathbf{x}_a \in \mathbb{R}^d$

$f_{\mathbf{w}}(a) = \mathbf{x}_a^T \mathbf{w}$



Use standard conjugate Normal Inverse Gamma p

$$\mathrm{NIG}(\mathbf{w}, \sigma^2 \mid \mathbf{w}_0, \mathbf{V}_0, \alpha_0, \beta_0) =$$

$$|2\pi\sigma^2 \mathbf{V}_0|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{w} - \mathbf{w}_0)^T \mathbf{V}_0^{-1}(\mathbf{w} - \mathbf{w}_0)\right\}$$

$$\times \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)(\sigma^2)^{\alpha_0+1}} \exp\left\{-\frac{\beta_0}{\sigma^2}\right\} .$$

# Linear bandits

The posterior mean and covariance are analytical

$$\mathbf{w}_n = \mathbf{V}_n(\mathbf{V}_0^{-1}\mathbf{w}_0 + \mathbf{X}^T\mathbf{y})$$

$$\mathbf{V}_n = (\mathbf{V}_0^{-1} + \mathbf{X}^T\mathbf{X})^{-1}$$

$$\alpha_n = \alpha_0 + n/2$$

$$\beta_n = \beta_0 + \frac{1}{2}\left(\mathbf{w}_0^T\mathbf{V}_0^{-1}\mathbf{w}_0 + \mathbf{y}^T\mathbf{y} - \mathbf{w}_n^T\mathbf{V}_n^{-1}\mathbf{w}_n\right)$$

Once again, it is straightforward to do Thompson

$$a_{n+1} = \arg\max_a \mathbf{x}_a^T\tilde{\mathbf{w}} \;\; \text{where } \tilde{\mathbf{w}} \sim p(\mathbf{w} \mid \mathcal{D}_n)$$

# Being linear in features

It is trivial to extend the linear model using feature

$J$ basis functions $\phi_j : \mathcal{X} \mapsto \mathbb{R}$

$$f(\mathbf{x}) = \Phi(\mathbf{x})^T \mathbf{w}$$

Features can be RBFs, sinusoids

$$\phi_j(\mathbf{x}) = \exp\left\{-\frac{1}{2}(\mathbf{x} - \mathbf{z}_j)^T \mathbf{\Lambda}(\mathbf{x} - \mathbf{z}_j)\right\}$$

$$\phi_j(\mathbf{x}) = \exp\left\{-i\mathbf{x}^T \boldsymbol{\omega}_j\right\}$$

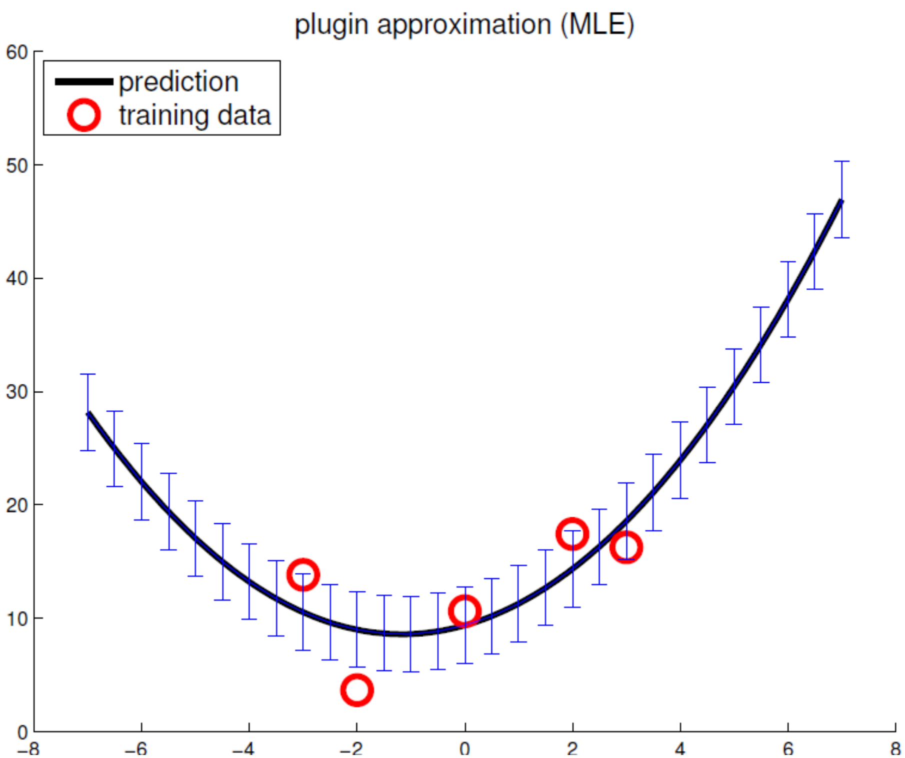
Or even popular deep networks

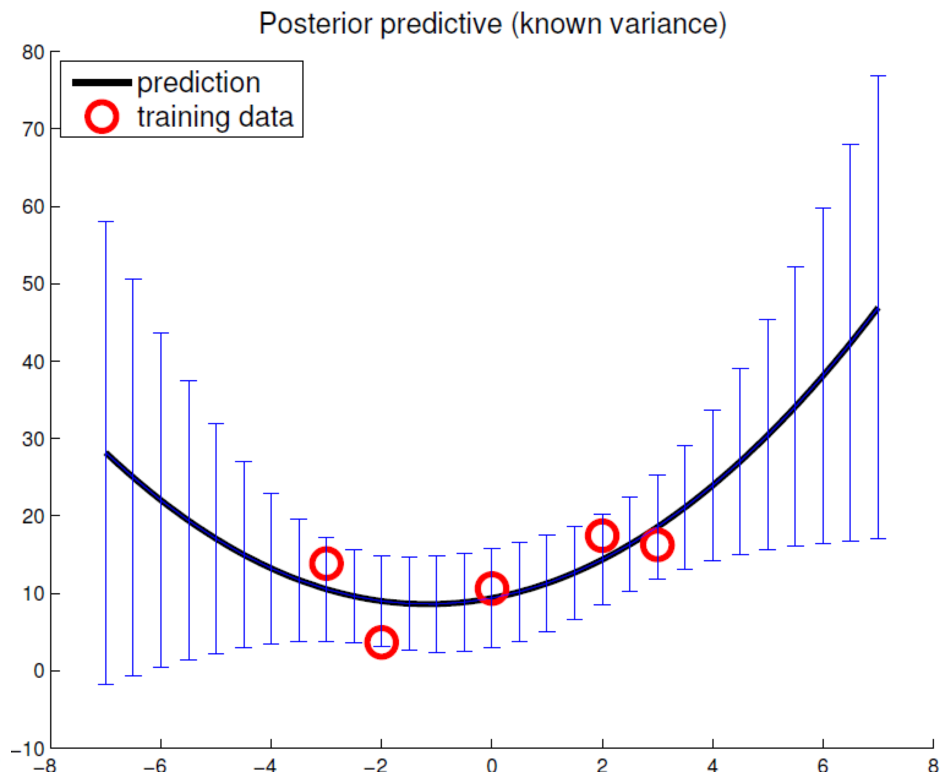
$$\Phi(\mathbf{x}) = \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1(\mathbf{x})$$

# One reason for Bayes

The predictive distribution at a test point $\mathbf{x}$ is:

$$
\begin{aligned}
p(y|\mathbf{x}, \mathcal{D}, \sigma^2) &= \int \mathcal{N}(y|\mathbf{x}\mathbf{w}, \sigma^2)\mathcal{N}(\mathbf{w}|\mathbf{w}_n, \mathbf{V}_n)d\mathbf{w} \\
&= \mathcal{N}(y|\mathbf{x}\mathbf{w}_n, \sigma^2 + \mathbf{x}\mathbf{V}_n\mathbf{x}^T)
\end{aligned}
$$



# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# From linear models to Gaussian processes

$$
\begin{aligned}
p(\mathbf{y} \mid \mathbf{X}, \sigma^2) &= \int p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}, \sigma^2)\, p(\mathbf{w} \mid 0, \mathbf{V}_0)\, \mathrm{d}\mathbf{w} \\
&= \int \mathcal{N}(\mathbf{y} \mid \mathbf{X}\mathbf{w}, \sigma^2 \mathbf{I})\, \mathcal{N}(\mathbf{w} \mid 0, \mathbf{V}_0)\, \mathrm{d}\mathbf{w} \\
&= \mathcal{N}(\mathbf{y} \mid 0, \mathbf{X}\mathbf{V}_0\mathbf{X}^T + \sigma^2 \mathbf{I})\,.
\end{aligned}
$$

$$
p(\mathbf{y} \mid \mathbf{X}, \sigma^2) = \mathcal{N}(\mathbf{y} \mid 0, \mathbf{\Phi}\mathbf{V}_0\mathbf{\Phi}^T + \sigma^2 \mathbf{I})
$$

$$
\begin{aligned}
\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) &= \mathbf{\Phi}(\mathbf{x}_i)\mathbf{V}_0\mathbf{\Phi}(\mathbf{x}_j)^T \\
&= \langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) \rangle_{\mathbf{V}_0}
\end{aligned}
$$

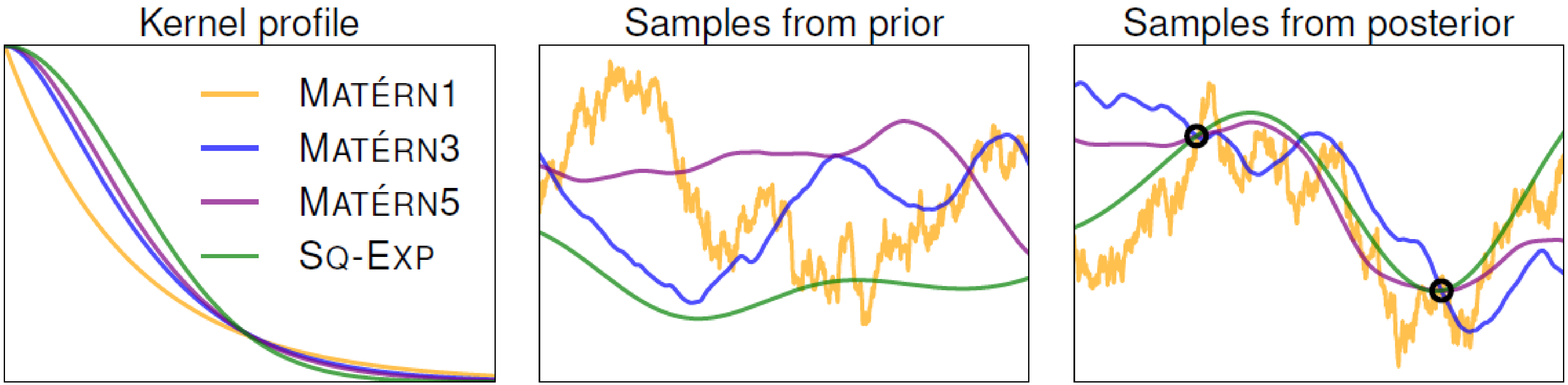
# Gaussian processes

$$p(\mathbf{y}_\star \mid \mathbf{X}_\star, \mathbf{X}, \mathbf{y}, \sigma^2) = \frac{p(\mathbf{y}_\star, \mathbf{y} \mid \mathbf{X}_\star, \mathbf{X}, \sigma^2)}{p(\mathbf{y} \mid \mathbf{X}, \sigma^2)}$$

$$\mathbf{f} \mid \mathbf{X} \sim \mathcal{N}(\mathbf{m}, \mathbf{K})$$
$$\mathbf{y} \mid \mathbf{f}, \sigma^2 \sim \mathcal{N}(\mathbf{f}, \sigma^2 \mathbf{I})$$

$$\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$$
$$\mu_n(\mathbf{x}) = \mu_0(\mathbf{x}) + \mathbf{k}(\mathbf{x})^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} (\mathbf{y} - \mathbf{m})$$
$$\sigma_n^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}(\mathbf{x}),$$

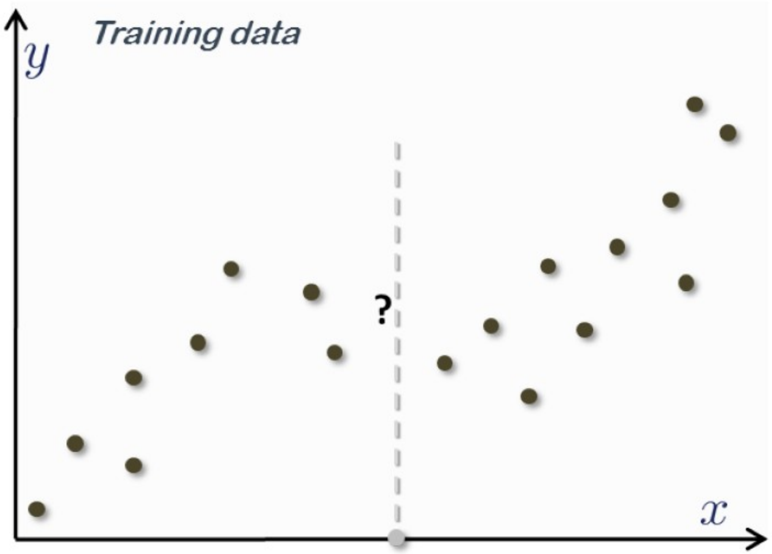
# Gaussian processes

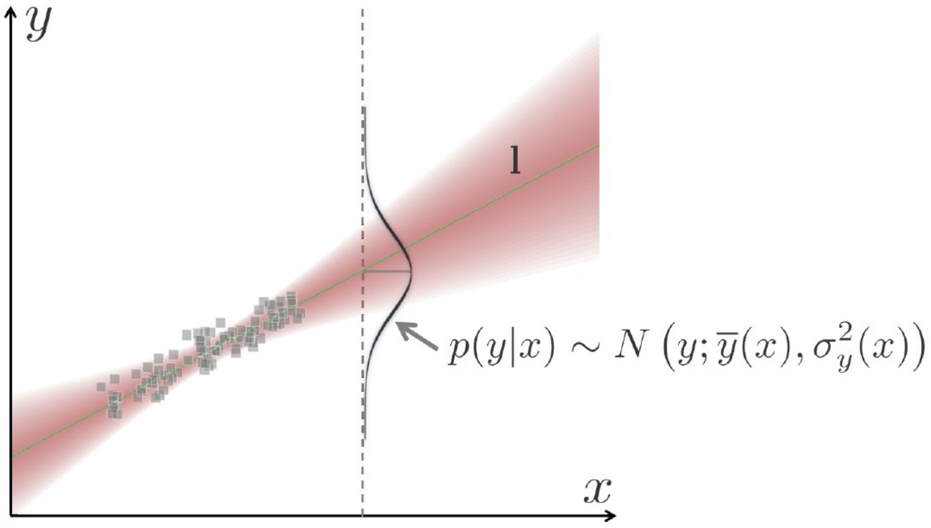


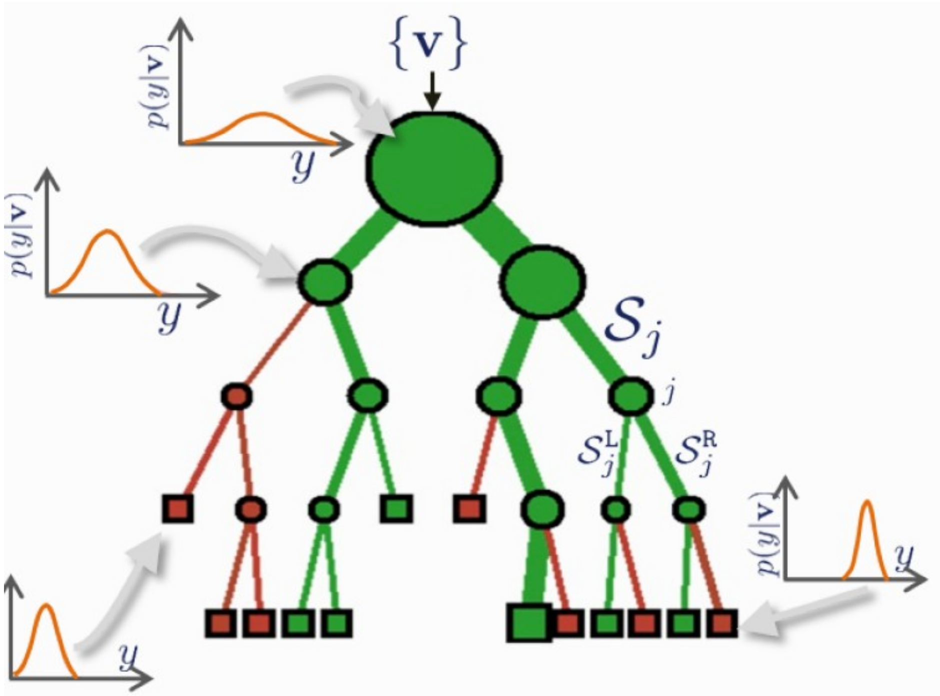
# GP log marginal likelihood And GP hyper-parameters

$$\log p(\mathbf{y}|\mathbf{x}_{1:n}, \theta) = -\frac{1}{2}(\mathbf{y} - \mathbf{m}_\theta)^T (\mathbf{K}^\theta + \sigma^2 \mathbf{I})^{-1} (\mathbf{y} - \mathbf{m}_\theta) - \frac{1}{2} \log |\mathbf{K}^\theta + \sigma^2 \mathbf{I}| - \frac{n}{2} \log(2\pi)$$

# Trees for regression

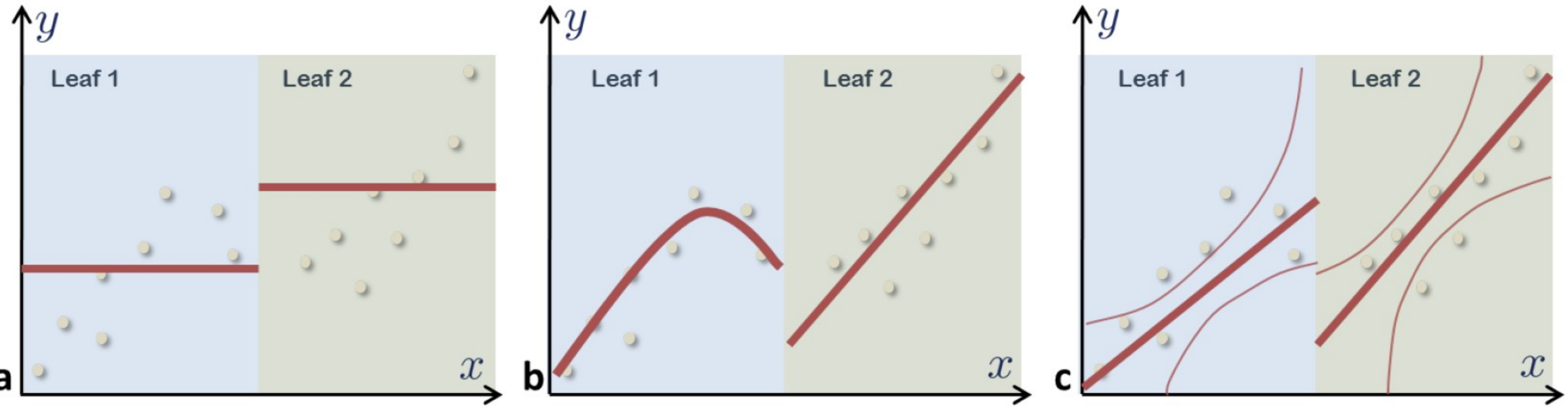


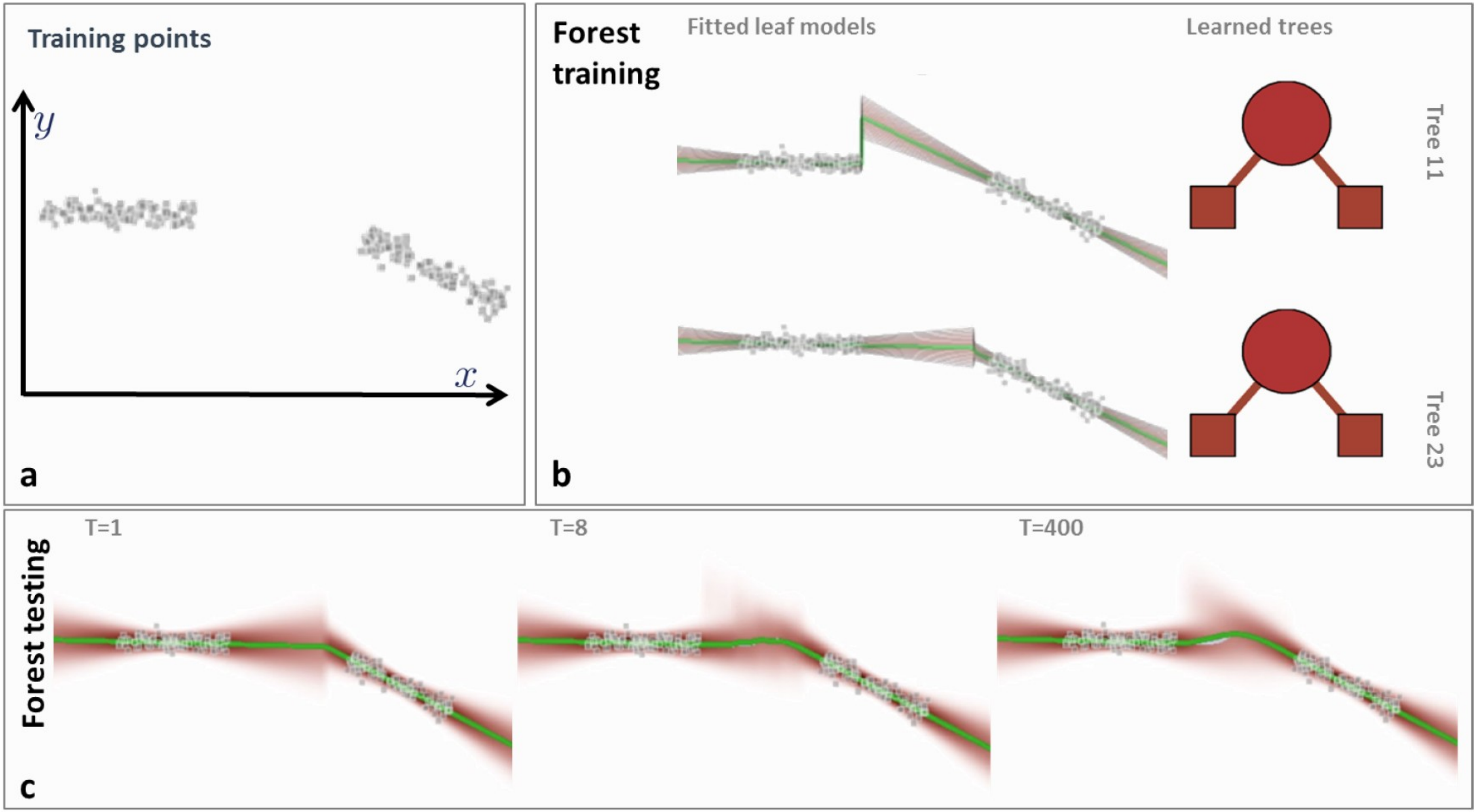

[Criminisi et al, 2011]

# Regression trees




**[Criminisi et al, 2011]**

# Regression forests




[Criminisi et al, 2011]

# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Bayesian experimental design

Uses the principle of maximum expected utility:

$$\alpha(\mathbf{X}) := \mathbb{E}_{\mathbf{w}}\mathbb{E}_{\mathbf{y}\,|\,\mathbf{X},\mathbf{w}}\left[U(\mathbf{X},\mathbf{y},\mathbf{w})\right]$$

Example 1: active learning

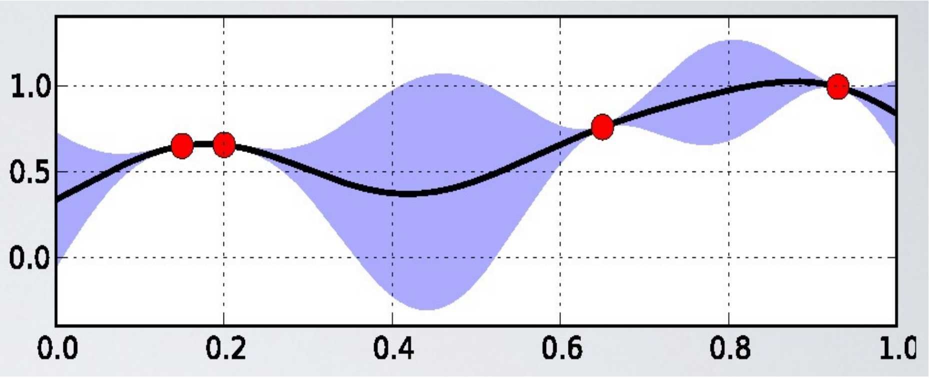
$$\alpha(\mathbf{X}) = \mathbb{E}_{\mathbf{w}}\mathbb{E}_{\mathbf{y}\,|\,\mathbf{X},\mathbf{w}}\left[\int p(\mathbf{w}'\,|\,\mathbf{X},\mathbf{y})\log p(\mathbf{w}'\,|\,\mathbf{X},\mathbf{y})\mathrm{d}\mathbf{w}'\right]$$

Example 2: sequential decision making with discou

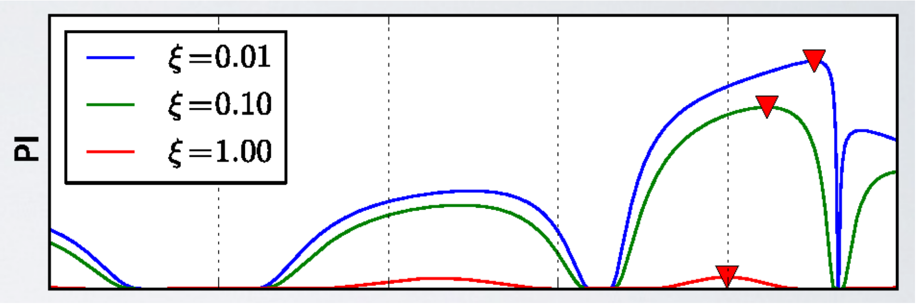
$$\alpha(\mathbf{X}) = \mathbb{E}_{\mathbf{w}}\mathbb{E}_{\mathbf{y}\,|\,\mathbf{X},\mathbf{w}}\left[\sum_{i=1}^{n}\gamma^{i-1}y_i\right]$$

# Some acquisition functions

$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v \,|\, \mathbf{x}, \theta}[U(\mathbf{x}, v, \theta)]$$

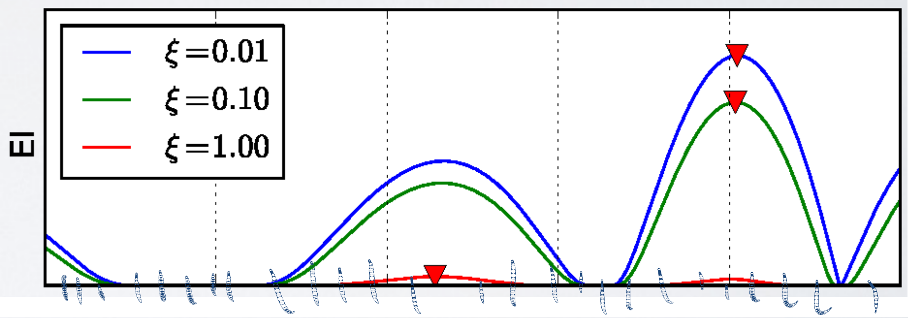



$$\alpha_{\mathrm{PI}}(\mathbf{x}; \mathcal{D}_n) := \mathbb{P}[v > \tau] = \Phi\left(\frac{\mu_n(\mathbf{x}) - \tau}{\sigma_n(\mathbf{x})}\right)$$

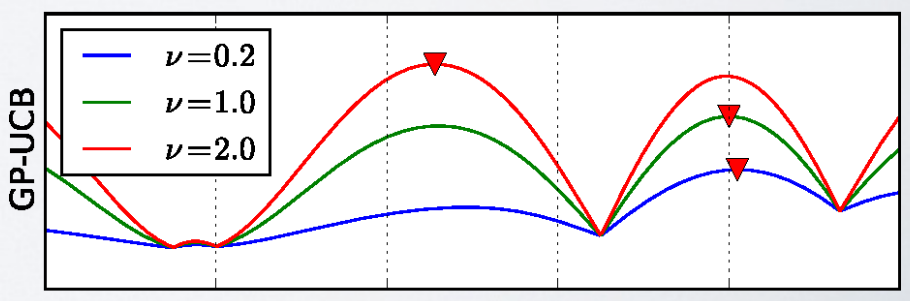



$$I(\mathbf{x}, v, \theta) := (v - \tau)\,\mathbb{I}(v > \tau)$$

$$\begin{aligned}\alpha_{\mathrm{EI}}(\mathbf{x}; \mathcal{D}_n) &:= \mathbb{E}\left[I(\mathbf{x}, v, \theta)\right] \\ &= (\mu_n(\mathbf{x}) - \tau)\Phi\left(\frac{\mu_n(\mathbf{x}) - \tau}{\sigma_n(\mathbf{x})}\right) \\ &\quad + \sigma_n(\mathbf{x})\phi\left(\frac{\mu_n(\mathbf{x}) - \tau}{\sigma_n(\mathbf{x})}\right),\end{aligned}$$




$$\alpha_{\mathrm{UCB}}(\mathbf{x}; \mathcal{D}_n) := \mu_n(\mathbf{x}) + \beta_n \sigma_n(\mathbf{x})$$



# Thompson sampling with GPs

Posterior over location of the minimu $p_\star(\mathbf{x} \mid \mathcal{D}_n)$

$$\mathbb{P}(\,\mathrm{d}\mathbf{x}_\star|\mathcal{D}) = \mathbb{P}\left(\arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \in \,\mathrm{d}\mathbf{x}_\star \middle| \mathcal{D}\right)$$

Thompson: $\mathbf{x}_{n+1} \sim p_\star(\mathbf{x} \mid \mathcal{D}_n)$

Mechanism: $\alpha_{\mathrm{TS}}(\mathbf{x};\mathcal{D}_n) := f^{(n)}(\mathbf{x})$

where $f^{(n)} \overset{s.s.}{\sim} \mathrm{GP}(\mu_0, k \mid \mathcal{D}_n)$

# Randomized Thompson sampling

Bochner's lemma tell us that:

$$k(\mathbf{x}, \mathbf{x}') = \nu\, \mathbb{E}_{\boldsymbol{\omega}}[e^{-i\boldsymbol{\omega}^T(\mathbf{x}-\mathbf{x}')}]$$

With sampling and this lemma, we can construct a
feature base $\boldsymbol{\omega}^{(i)} \sim s(\boldsymbol{\omega})/\nu$ ation that is amenable to computin

$$k(\mathbf{x}, \mathbf{x}') \approx \frac{\nu}{m} \sum_{i=1}^{m} e^{-i\boldsymbol{\omega}^{(i)T}\mathbf{x}} e^{i\boldsymbol{\omega}^{(i)T}\mathbf{x}'}$$

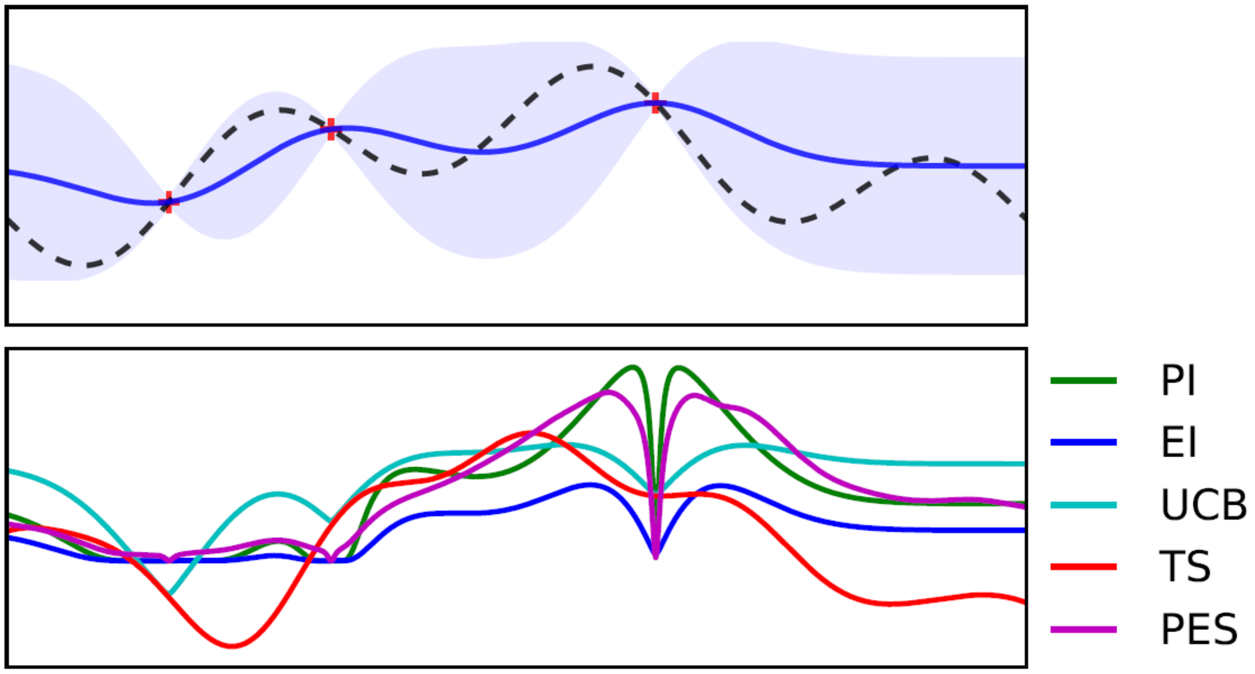
Then do "linear" Bayesian optimization with sinuso

# Entropy search and variants

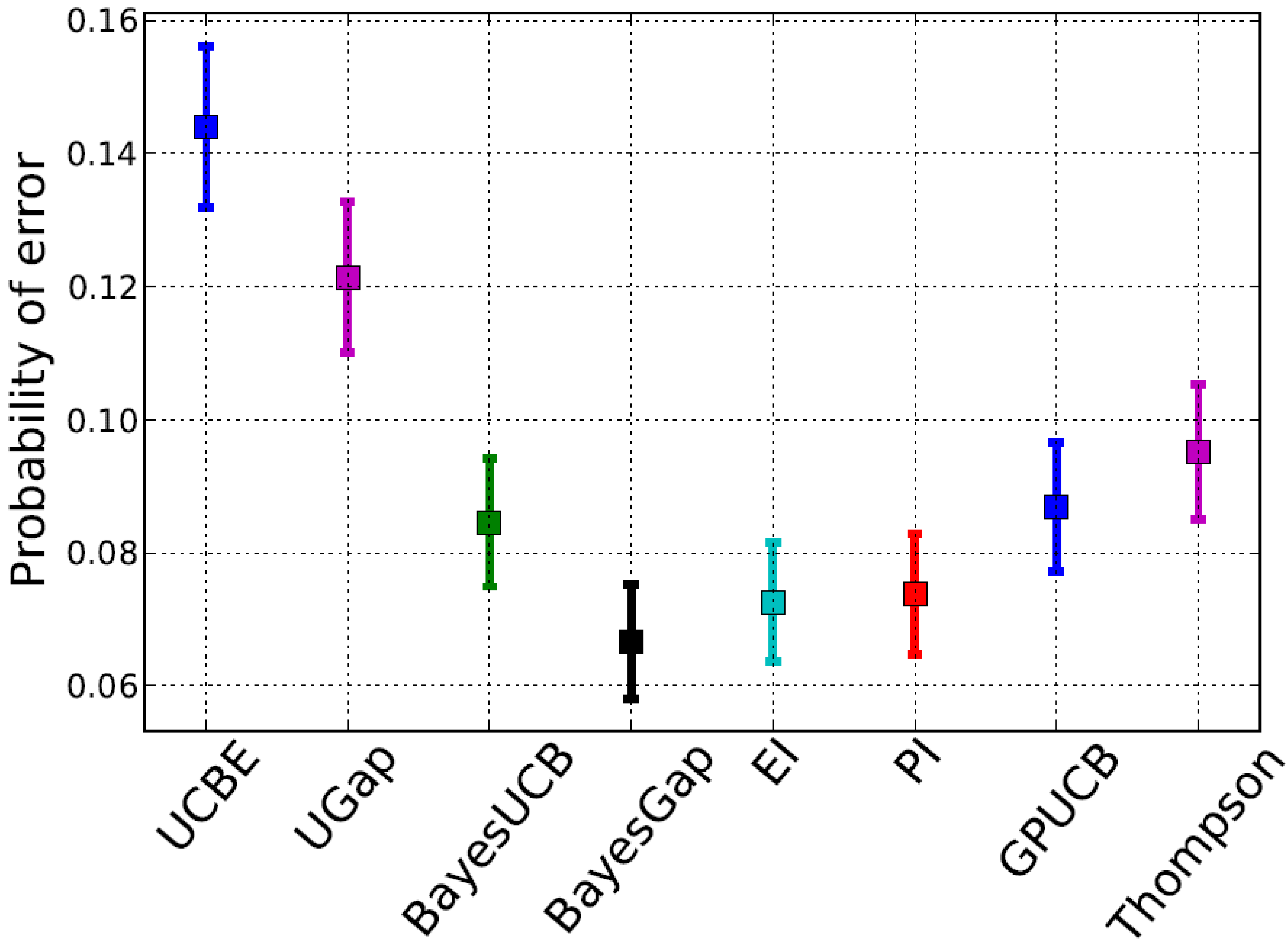
Posterior over location of the minimu $p_\star(\mathbf{x} \mid \mathcal{D}_n)$

Utility: minimize uncertainty of location of minimur

$$U(\mathbf{x}, y, \theta) = H(\mathbf{x}^\star \mid \mathcal{D}_n) - H(\mathbf{x}^\star \mid \mathcal{D}_n \cup \{(\mathbf{x}, y)\})$$

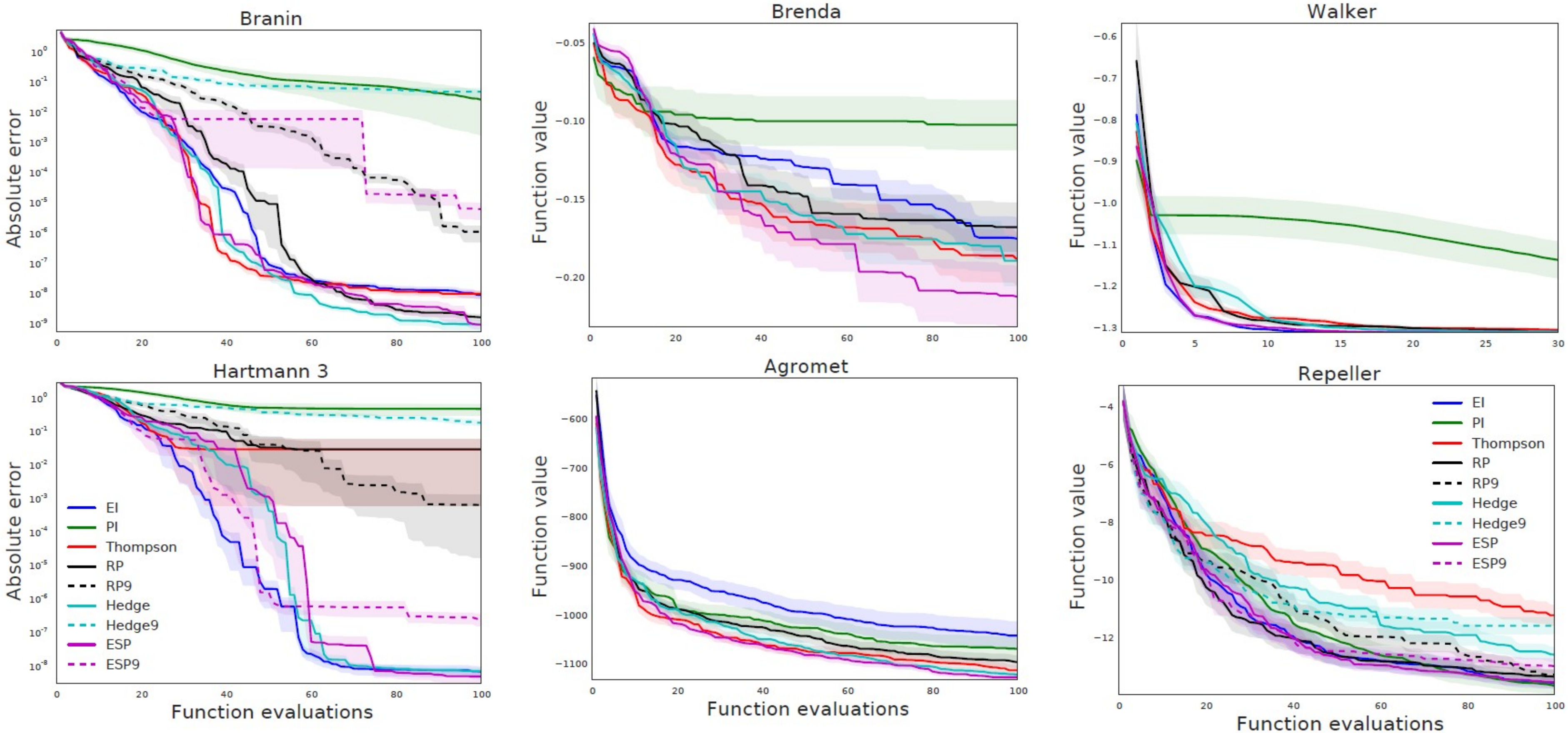


# The choice of utility in practice




[Hoffman, Shahriari & dF, 2013]

# The choice of utility in practice



# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Hyper-parameters and robustness

Learning the hyper-parameters of the GP is important.

**The tuning method must be automatic!**

One of the best ways to manage the GP hyper-parameters is to integrate them out, *e.g.* with slice sampling as in **Spearmint**. But this is still dangerous!

# Hyper-parameters and robustness

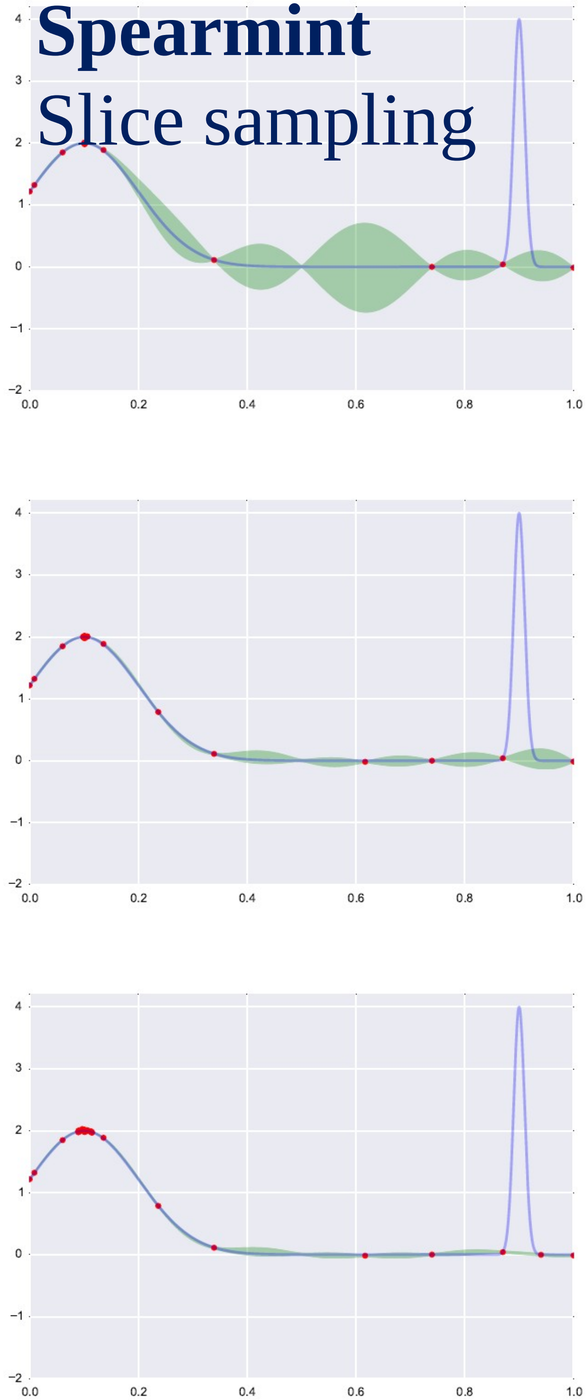
$$\alpha_{\boldsymbol{\theta}}^{\mathrm{EI}}(\mathbf{x}|\mathcal{D}_t) = \mathbb{E}[\max\{0, f(\mathbf{x}) - \mu_{\boldsymbol{\theta}}^{+}\}|\mathcal{D}_t] = \nu\sigma_t(\mathbf{x};\boldsymbol{\theta})[\frac{u}{\nu}\Phi(\frac{u}{\nu}) + \phi(\frac{u}{\nu})]$$

**Theorem 1.** *Let* $C_2 := \prod_{i=1}^{d} \frac{\theta_i^U}{\theta_i^L}$. *Suppose* $\boldsymbol{\theta}^L \leq \boldsymbol{\theta}_t \leq \boldsymbol{\theta}^U$ *for all* $t \geq 1$ *and* $f(\cdot) \in \mathcal{H}_{\boldsymbol{\theta}^U}(\mathcal{X})$. *If* $(\nu_t^{\boldsymbol{\theta}})^2 = \boldsymbol{\Theta}\left(\gamma_{t-1}^{\boldsymbol{\theta}} + \log^{1/2}(2t^2\pi^2/3\delta)\sqrt{\gamma_{t-1}^{\boldsymbol{\theta}}} + \log(t^2\pi^2/3\delta)\right)$ *for all* $t \geq 1$. *Then with probability at least* $1-\delta$, *the cumulative regret obeys the following rate:*

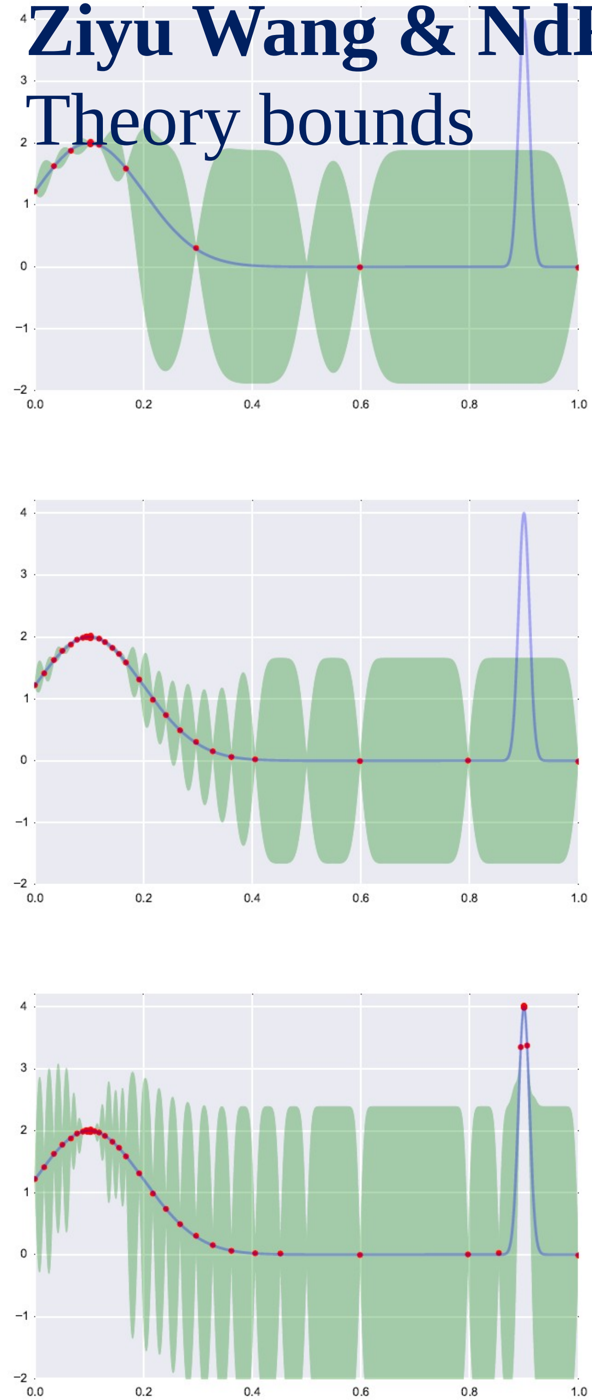
$$R_T = \mathcal{O}\left(\beta_T\sqrt{\gamma_T^{\boldsymbol{\theta}^L} T}\right), \tag{17}$$

*where* $\beta_T = 2\log\left(\frac{T}{\sigma^2}\right)\gamma_{T-1}^{\boldsymbol{\theta}^L} + \sqrt{8}\log\left(\frac{T}{\sigma^2}\right)\log^{1/2}(4T^2\pi^2/6\delta)\left(\sqrt{C_2}\|f\|_{\mathcal{H}_{\boldsymbol{\theta}^U}(\mathcal{X})} + \sqrt{\gamma_{T-1}^{\boldsymbol{\theta}^L}}\right) + C_2\|f\|_{\mathcal{H}_{\boldsymbol{\theta}^U}(\mathcal{X})}^2$.

**Spearmint**
Slice sampling



**Ziyu Wang & NdF 2014**
Theory bounds



$t = 20$

$t = 40$

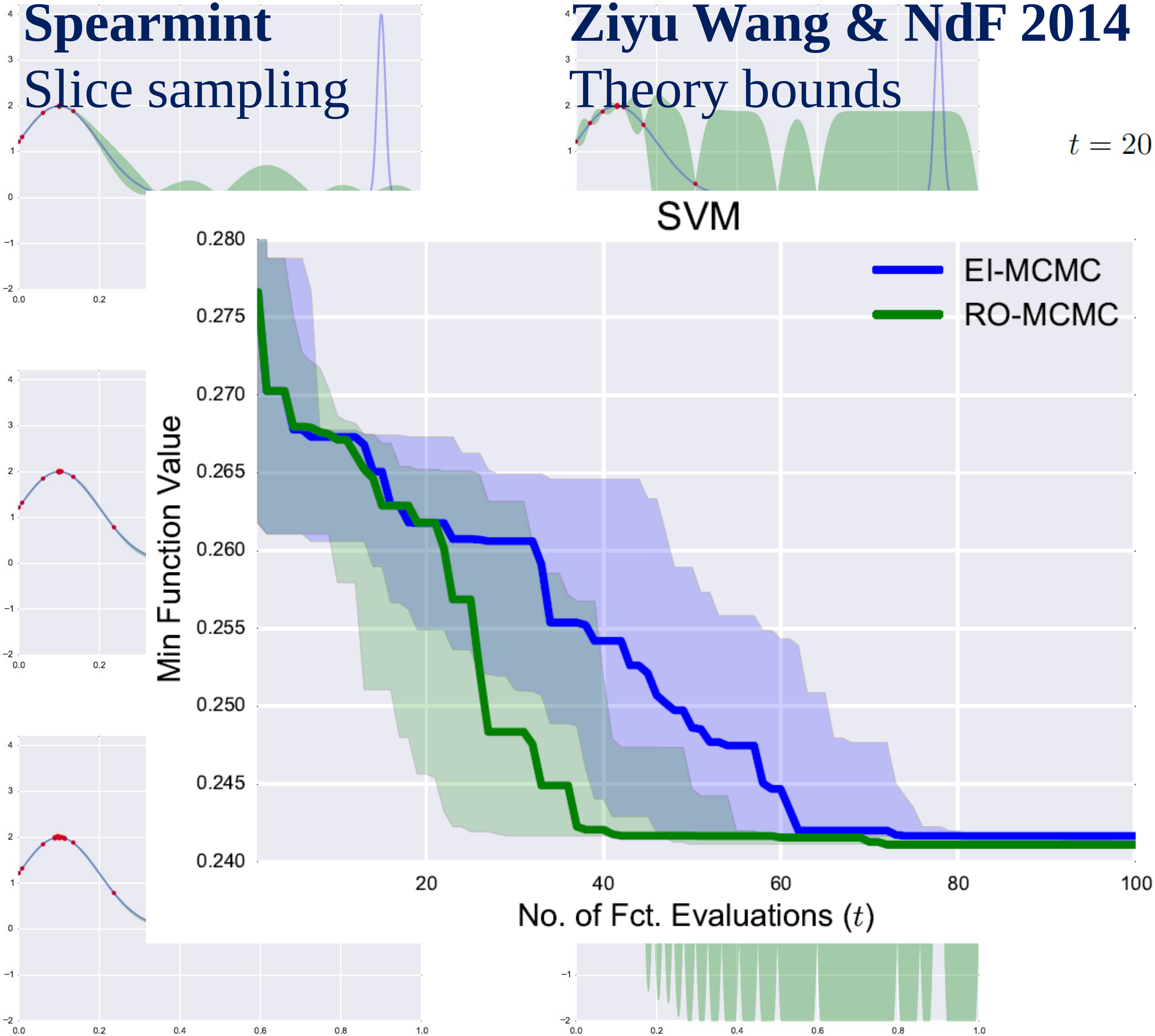
$t = 60$

**Spearmint**

Slice sampling

**Ziyu Wang & NdF 2014**

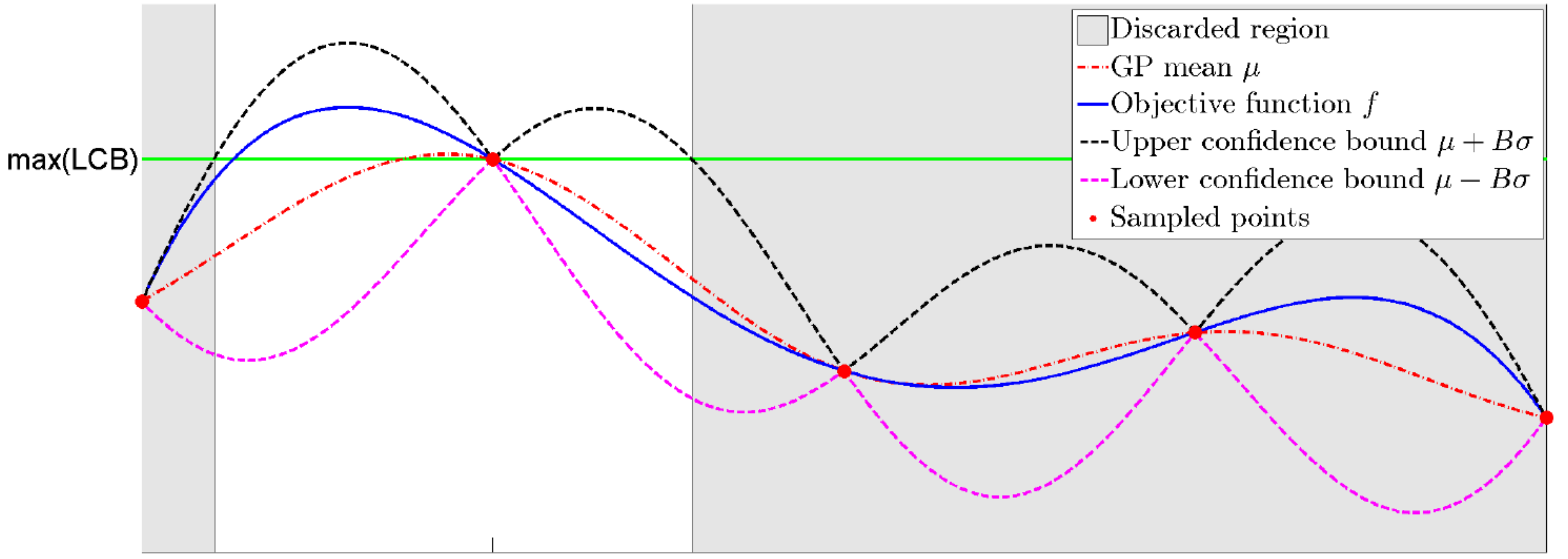
Theory bounds



# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Analysis of Bayes Opt [dF, Zoghi & Smola, 2012]

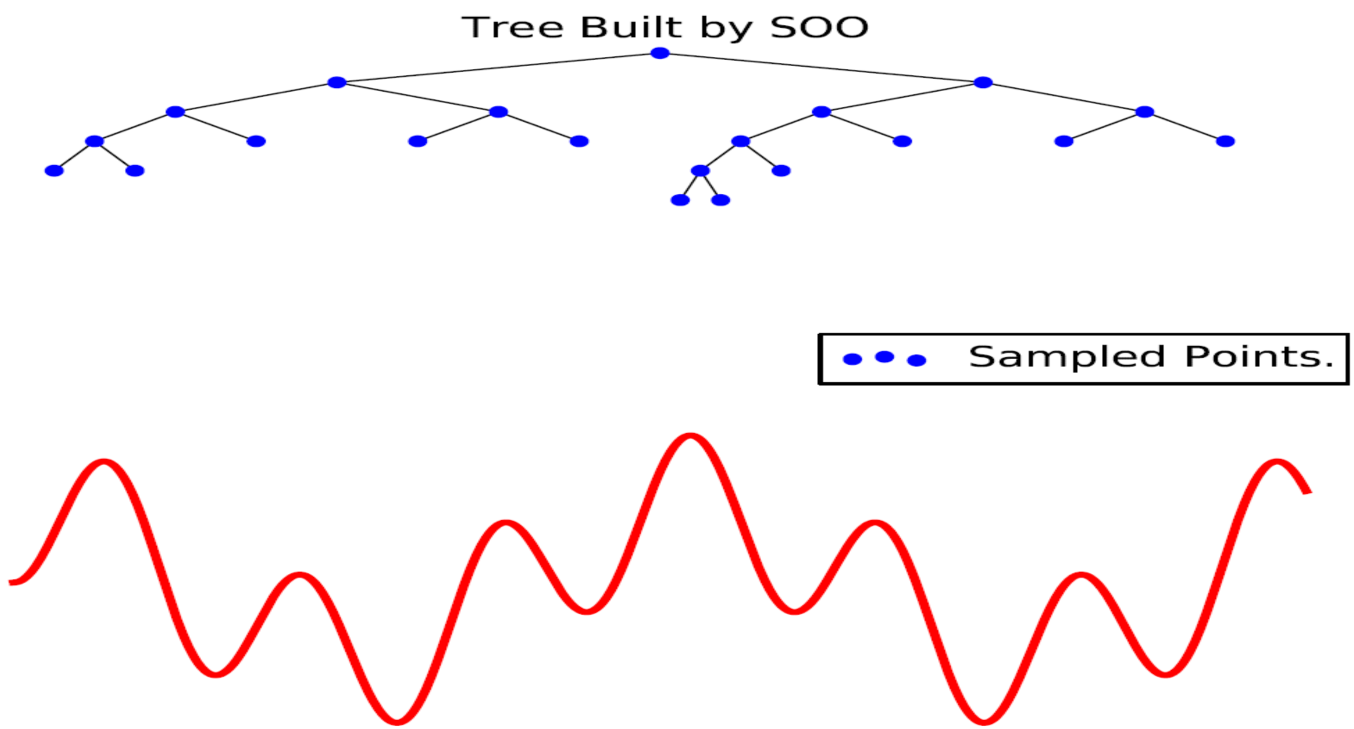



**Proposition 1 (Variance Bound) :** $$\sup_{\mathcal{D}} \sigma_T \leq \frac{Q\delta^2}{4}$$

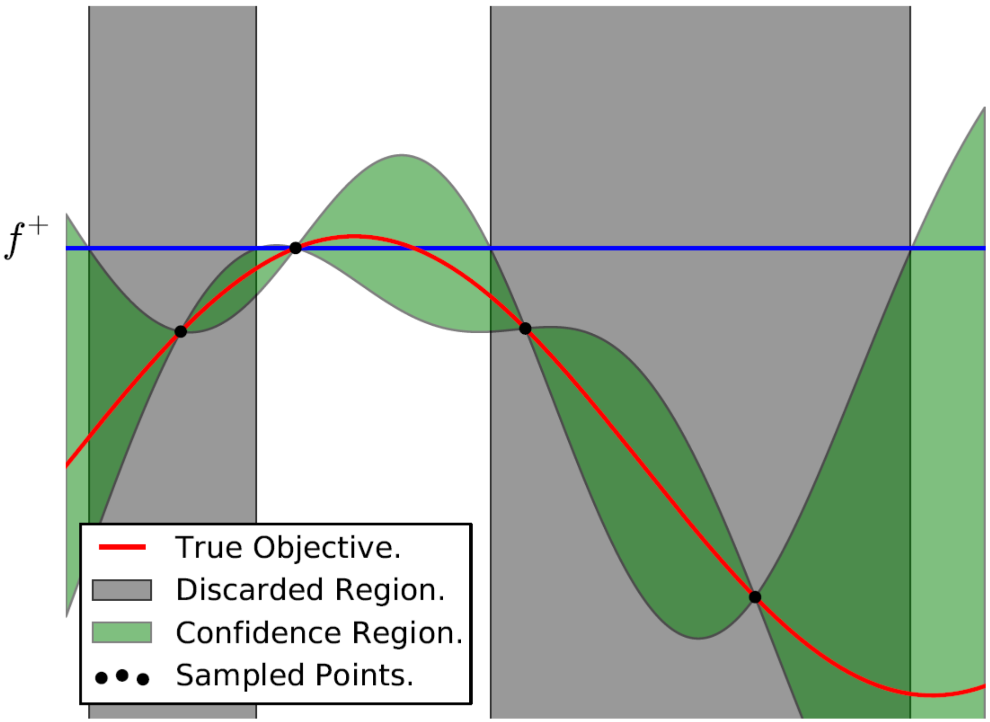
**Theorem 2 (Vanishing regret) :** $$r(x_t) < Ae^{-\frac{\tau t}{(\ln t)^{d/4}}}$$

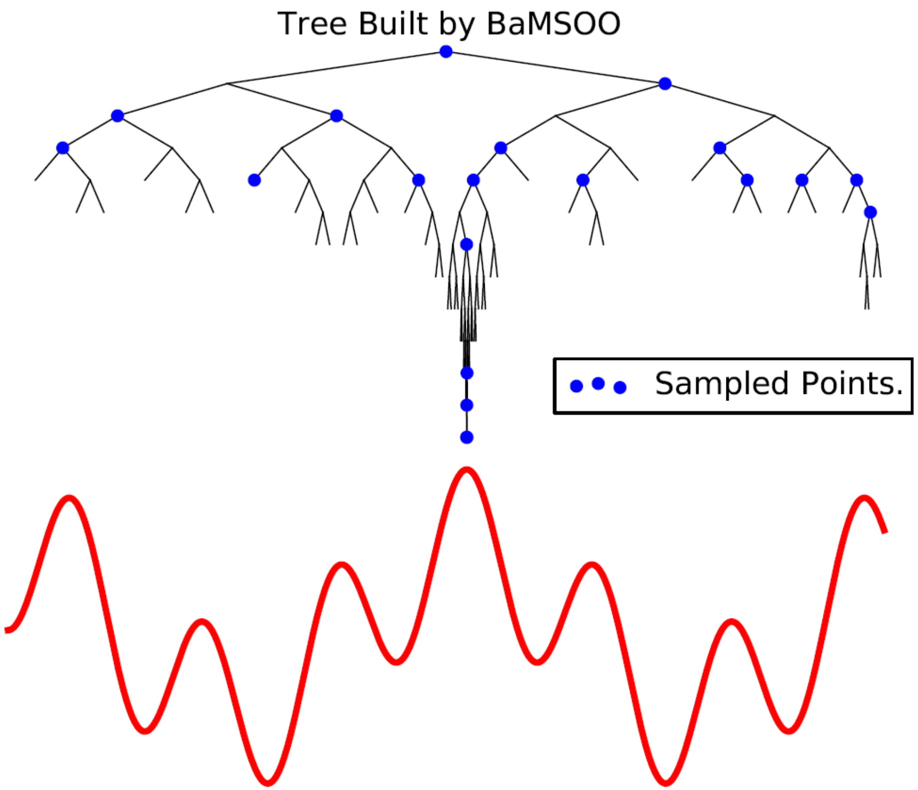
$$r(x_t) = f(x_M) - f(x_t)$$

The maximum depth function $t \mapsto h_{\max}(t)$ is a parameter of the algorithm.
**Initialization:** $\mathcal{T}_1 = \{(0,0)\}$ (root node). Set $t = 1$.
**while** True **do**
  Set $v_{\max} = -\infty$.
  **for** $h = 0$ to $\min(\text{depth}(\mathcal{T}_t), h_{\max}(t))$ **do**
    Among all leaves $(h,j) \in \mathcal{L}_t$ of depth $h$, select $(h,i) \in \arg\max_{(h,j)\in\mathcal{L}_t} f(x_{h,j})$
    **if** $f(x_{h,i}) \geq v_{\max}$ **then**
      Expand this node: add to $\mathcal{T}_t$ the $K$ children $(h+1, i_k)_{1\leq k\leq K}$
      Set $v_{\max} = f(x_{h,i})$, Set $t = t+1$
      **if** $t = n$ **then Return** $x(n) = \arg\max_{(h,i)\in\mathcal{T}_n} x_{h,i}$
    **end if**
  **end for**
**end while.**



# Multi-scale optimistic optimization




[Remi Munos - SOO, UCT]

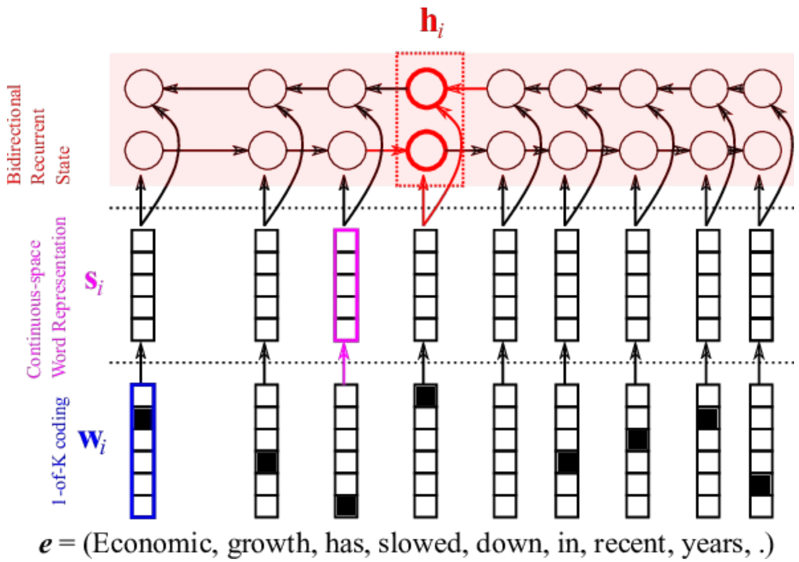
[Ziyu Wang et al, 2014]

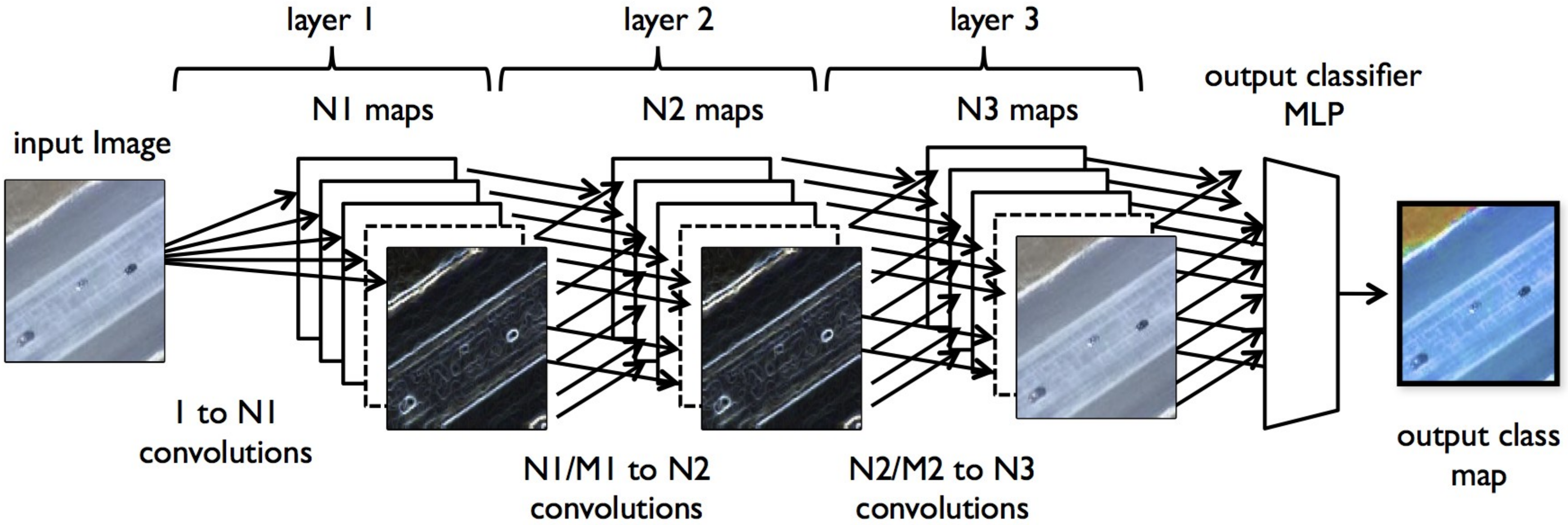
# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Conditional parameters

- A big problem in deep learning (see Torch 7)

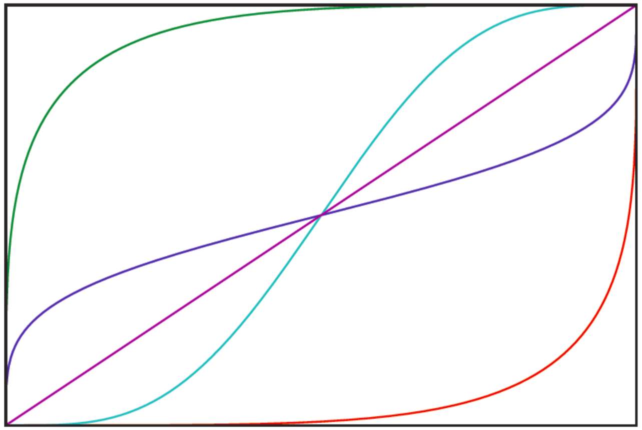


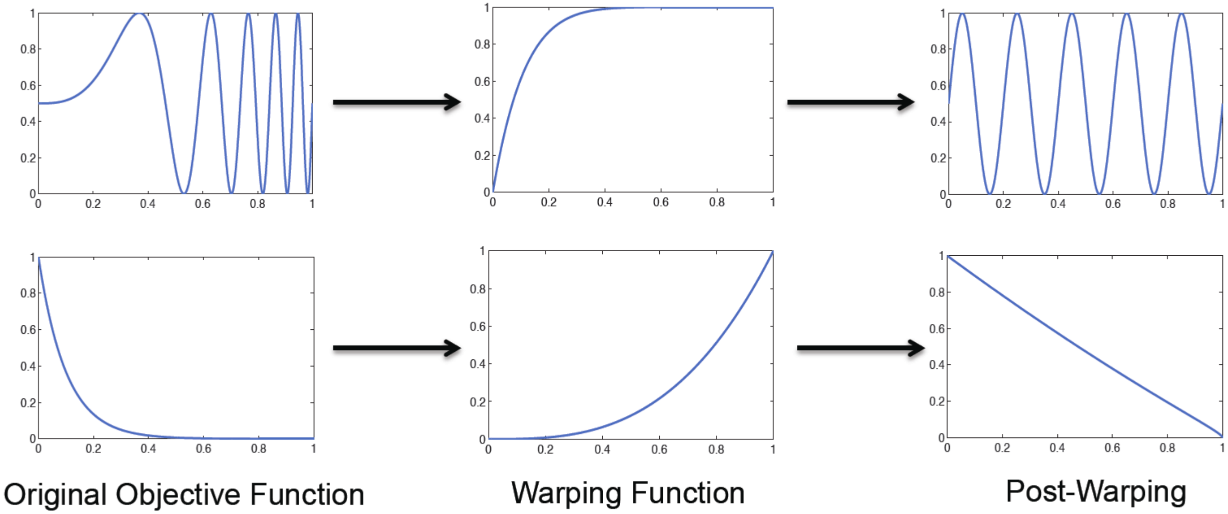



# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Input warping

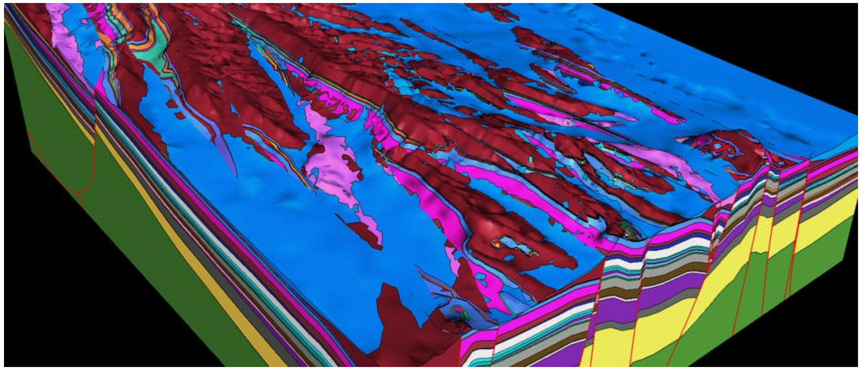




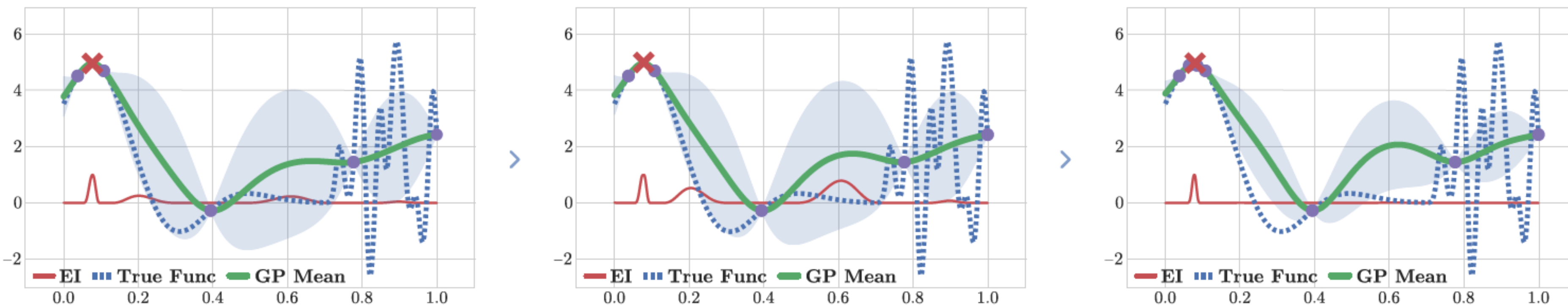

$$w_d(\mathbf{x}) = \frac{\mathbf{x}_d^{\alpha-1}(1-\mathbf{x}_d)^{\beta-1}}{B(\alpha,\beta)}$$

$$w_d(\mathbf{x}) = 1 - (1-\mathbf{x}_d^{\alpha})^{\beta}$$

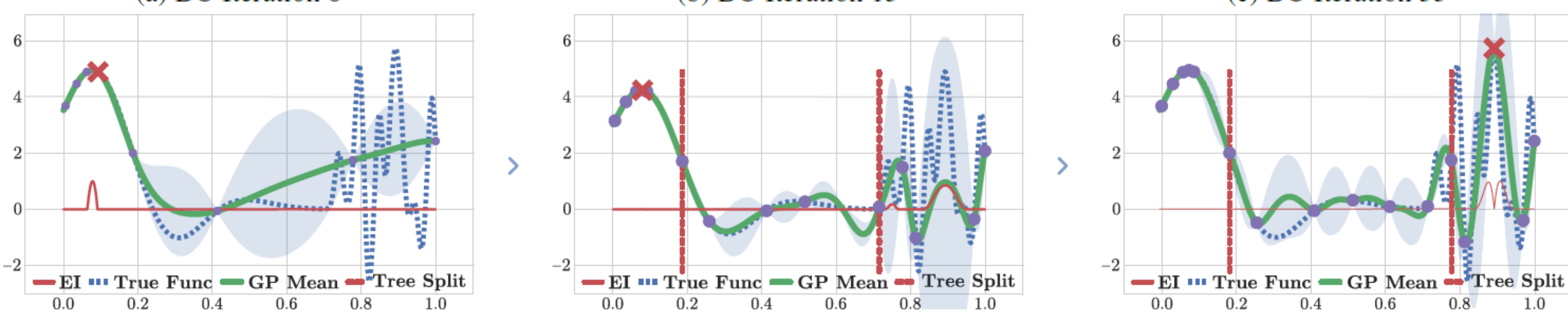
[Jasper Snoek, Kevin Swersky, Rich Zemel, Ryan Adams, 2014]



# Treed BayesOpt

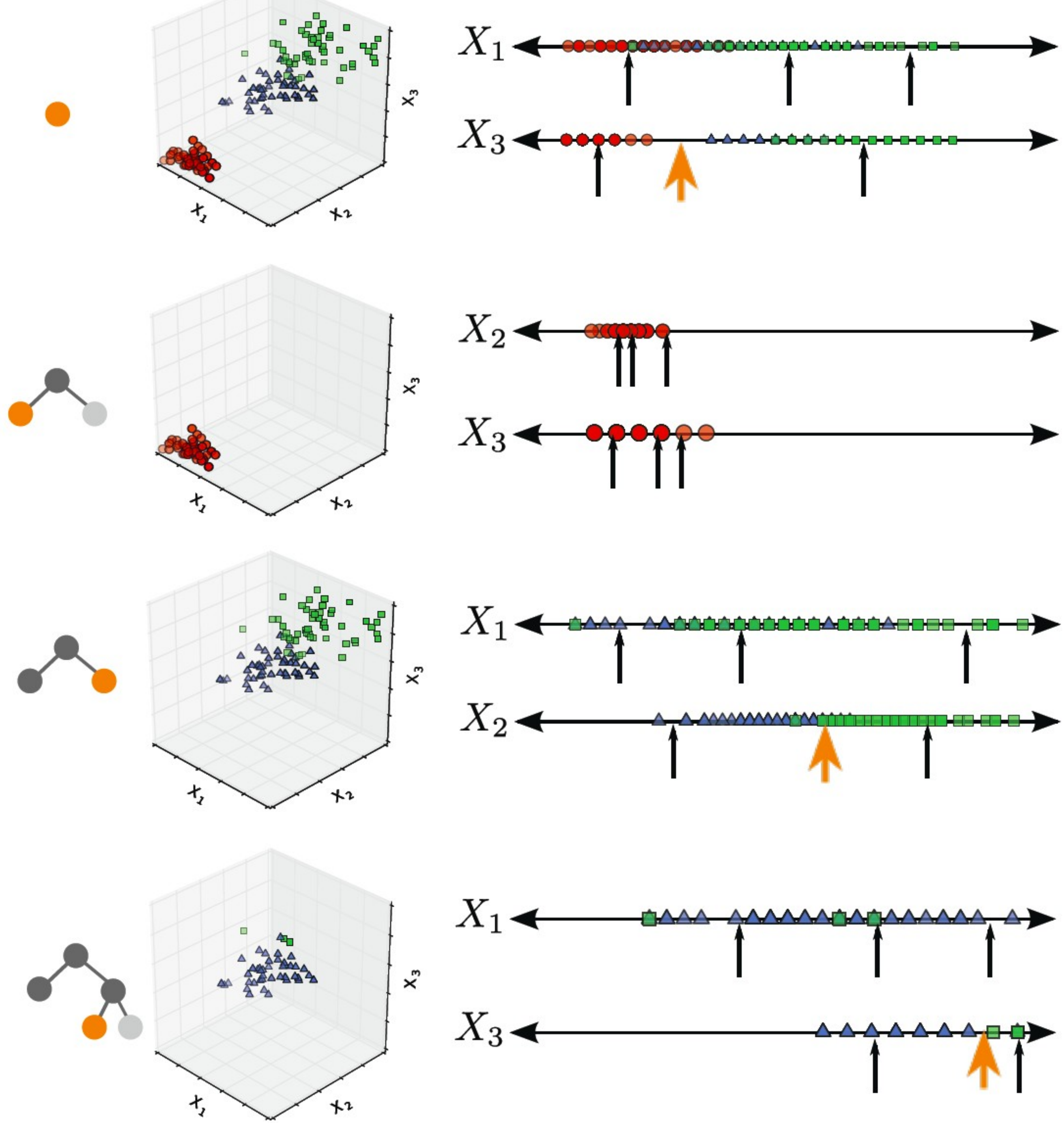



(a) BO Iteration 8 (b) BO Iteration 15 (c) BO Iteration 35



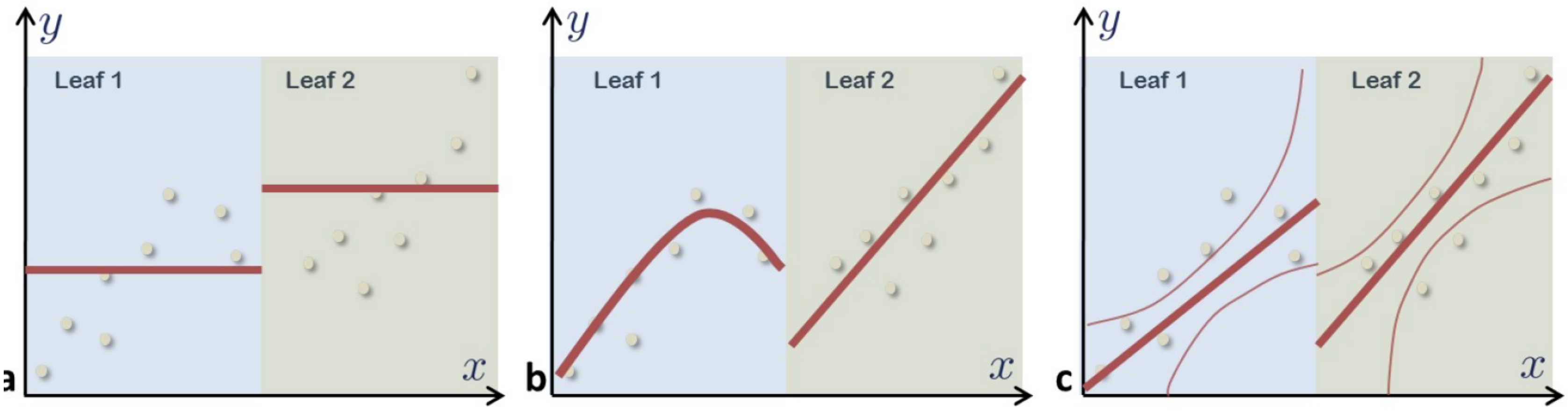

(d) HTBO Iteration 8 (e) HTBO Iteration 15 (f) HTBO Iteration 16

[Assael, Wang and NdF – Rob Gramacy has may papers using trees and GPs ]



X1
X3
X2
X3
X1
X2
X1
X3

# Treed BayesOpt

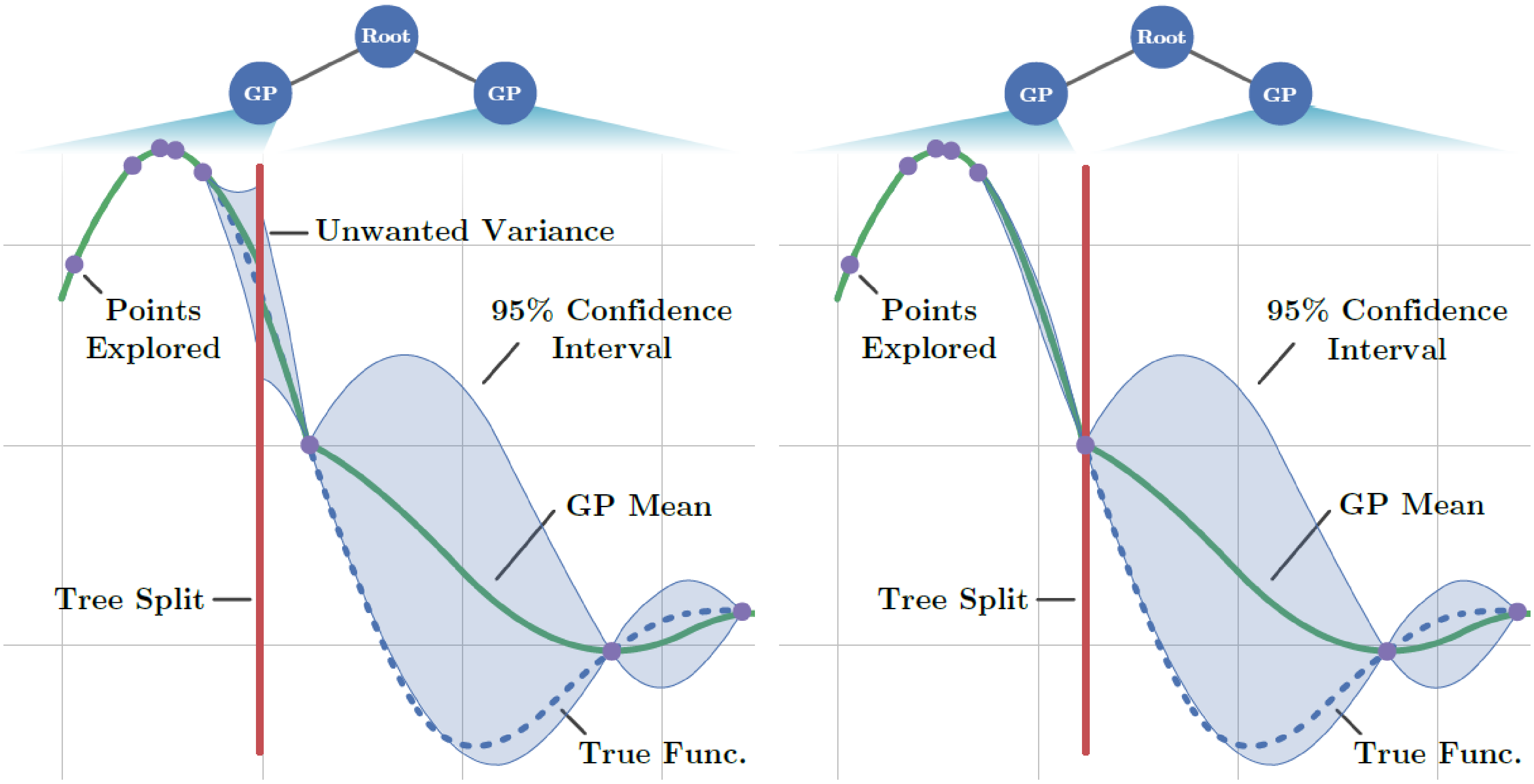



Each node has a function of the form $h(\mathbf{x}) > \tau$

$$\mathrm{U}(A) = \frac{1}{|A|} \sum_{y_i \in A} (\bar{y}_A - y_i)^2$$

$$\begin{aligned} \mathrm{I}(A, A'_{h,\tau}, A''_{h,\tau}) = \mathrm{U}(A) &- \frac{|A'_{h,\tau}|}{|A|} \mathrm{U}(A'_{h,\tau}) \\ &- \frac{|A''_{h,\tau}|}{|A|} \mathrm{U}(A''_{h,\tau}) \end{aligned}$$

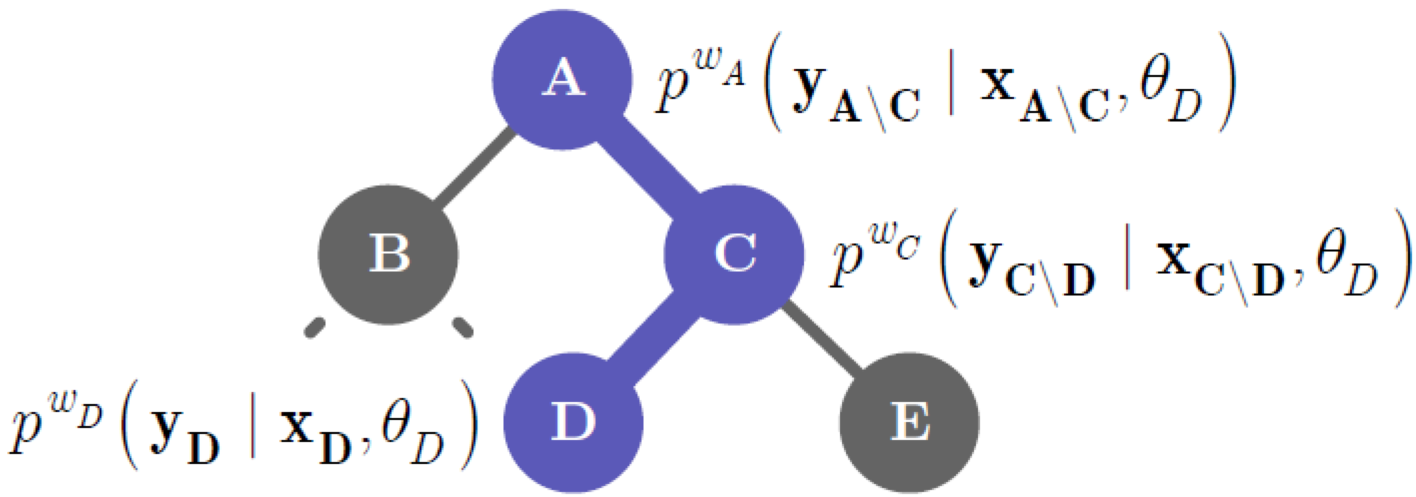
# Treed BayesOpt




(a) CART Splitting (b) Proposed approach

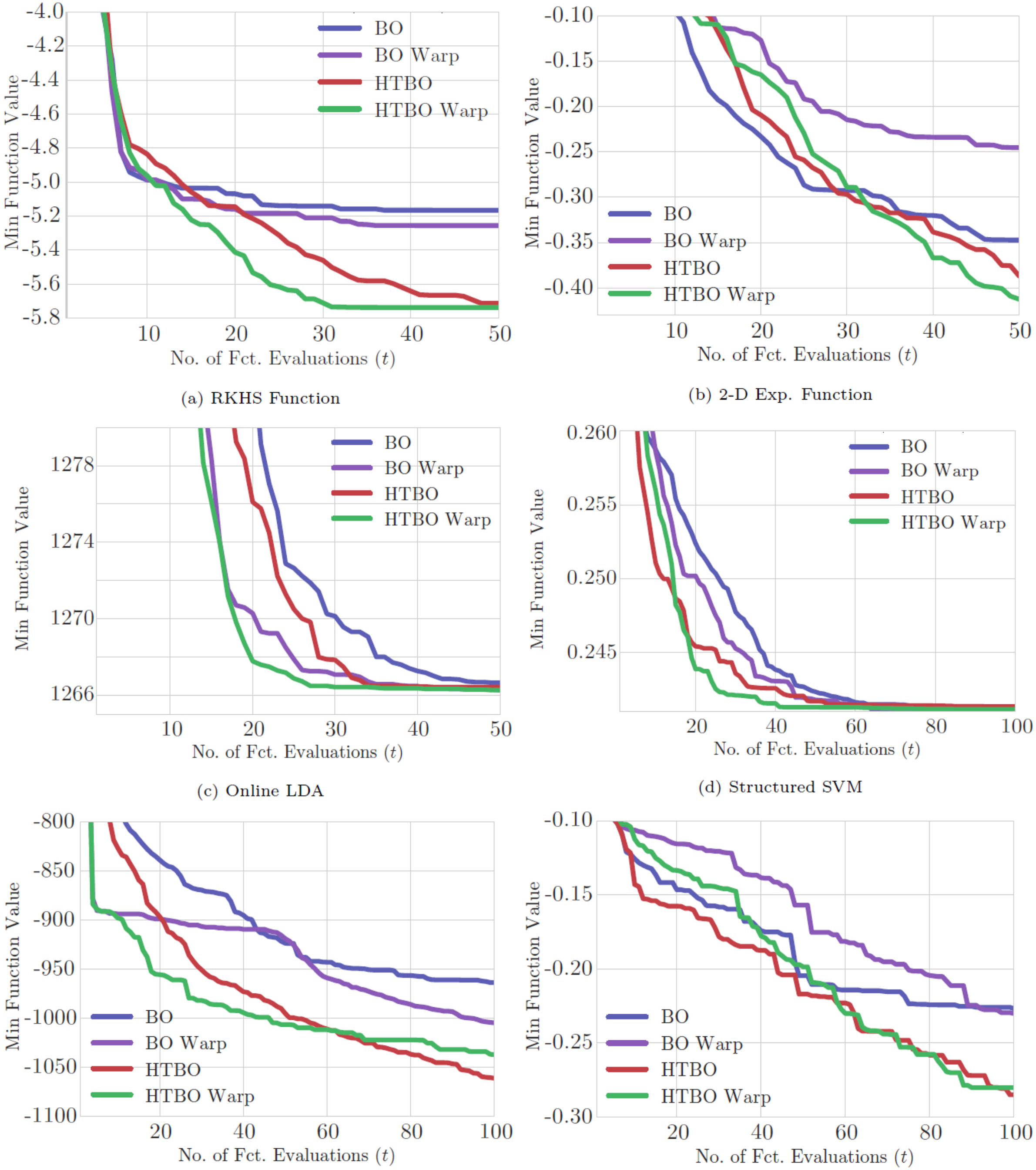
# Treed BayesOpt

- Aggregate to deal with paucity of data in leaves to estimate the hyper-parameters




$$p(\theta|\mathbf{x}_{1:t}, \mathbf{y}) \propto p(\theta) p^{w_0^j}(\mathbf{y}_{(j)}|\mathbf{x}_{(j)}, \theta) \times \prod_{i=1}^{|\rho^j|} p^{w_i^j}\left(\mathbf{y}_{(\rho_i^j \setminus \rho_{i-1}^j)} \middle| \mathbf{x}_{(\rho_i^j \setminus \rho_{i-1}^j)}, \theta\right)$$

# Warping + trees




(a) RKHS Function

(b) 2-D Exp. Function

(c) Online LDA

(d) Structured SVM

(e) Agromet

(f) Brenda mines

# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Parallelization

- Talk to David Ginsbourger. Essentially, augment the observations for finished runs with predicted observations for unfinished runs. E.g.,

$$
\begin{aligned}
\alpha(x;\mathcal{D}_n,\mathcal{D}_p) &= \int_{\mathbb{R}^J} \alpha(x;\mathcal{D}_n \cup \tilde{\mathcal{D}}_p) P(\tilde{y}_{1:J};\mathcal{D}_n) dy_{p1:J}, \\
&\approx \frac{1}{S}\sum_{s=1}^{S} \alpha(x;\mathcal{D}_n \cup \tilde{\mathcal{D}}_p^{(s)}),
\end{aligned}
$$

$$
\tilde{\mathcal{D}}_p^{(s)} \sim P(\tilde{y}_{1:J};\mathcal{D}_n),
$$

# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Constraints and cost sensitivity

- There are many approaches (Gramacy, Snoek, ...). Could use a GP with binary observations h(.) that indicate the probability of the constraint being satisfied:

$$\alpha_{\mathrm{wEI}}(\mathbf{x}) = \alpha_{\mathrm{EI}}(\mathbf{x}, \mathcal{D}_n) h(\mathbf{x}, \mathcal{D}_n)$$

- Often trivial scaling is used to deal with time.

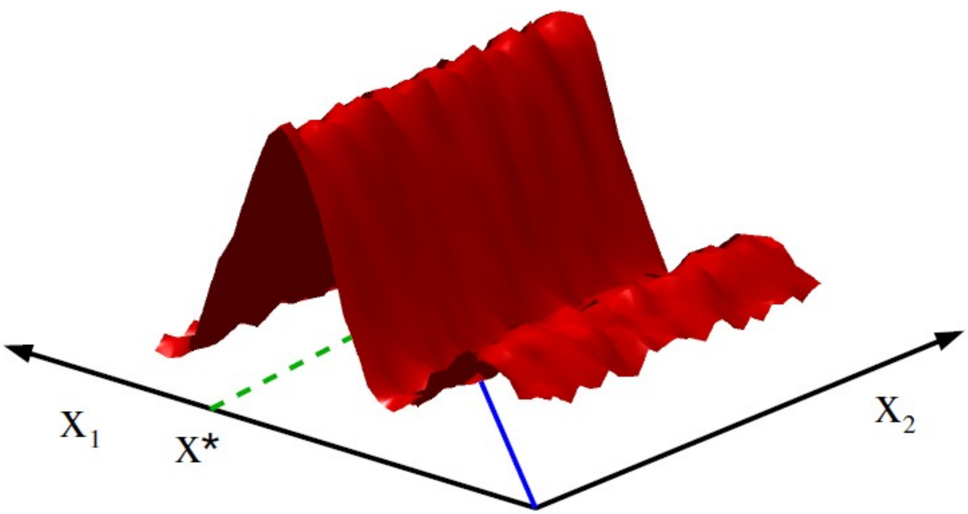
$$\mathrm{EI}(\mathbf{x}, \mathcal{D}_n)/c(\mathbf{x})$$

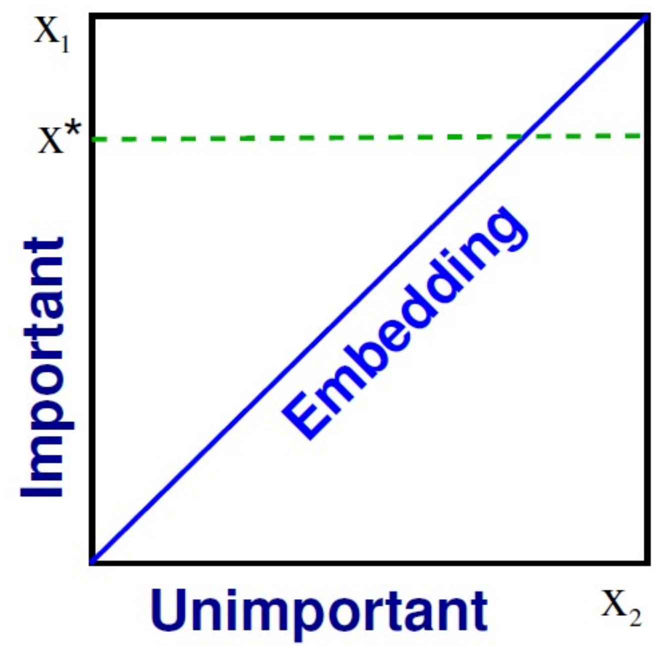
# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Random Embedding Bayesian Optimization

- **Embed** a low dimensional space into the high dimensional one
- Optimize only on the low dimensional space.




[Wang, Zoghi, Matheson, Hutter & dF,
**IJCAI 2013 Distinguished paper award**]

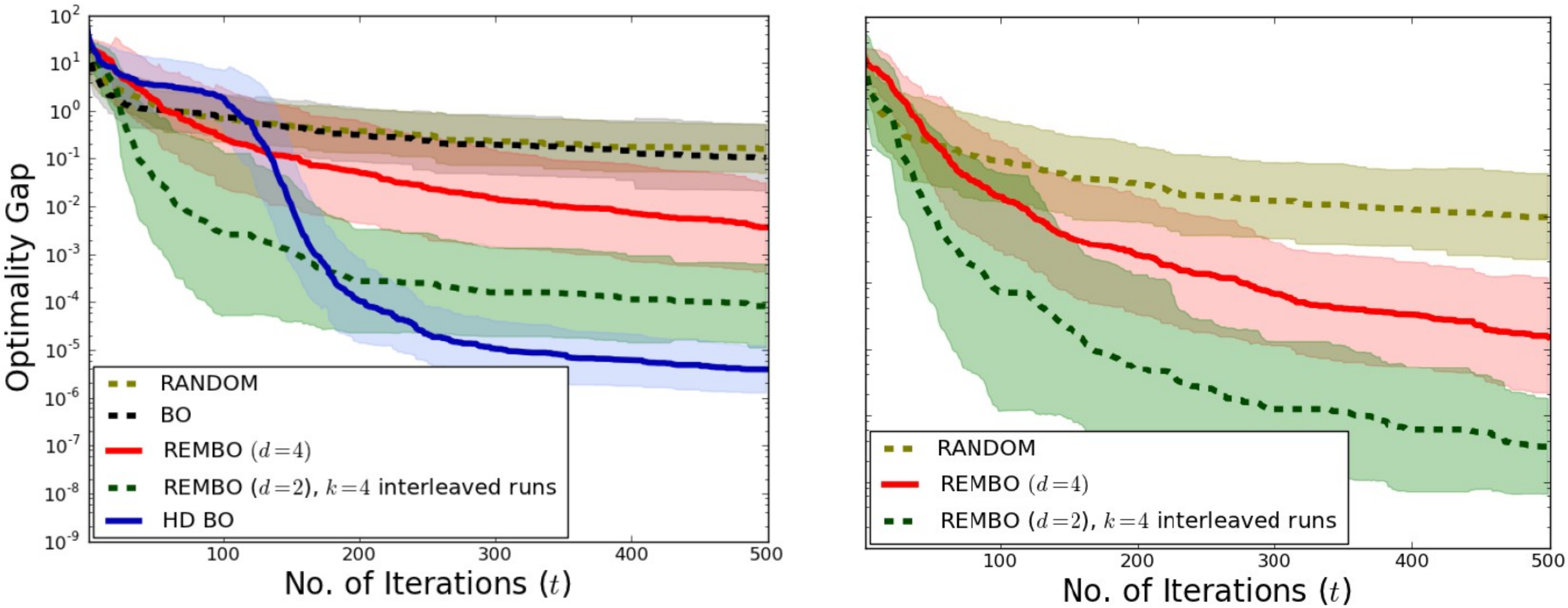
**Algorithm 1** Bayesian Optimization

**for** $t = 1, 2, \ldots$ **do**
    Find $\mathbf{x}_{t+1} \in \mathbb{R}^D$ by optimizing the acquisition function $u$: $\mathbf{x}_{t+1} = \arg\max_{\mathbf{x} \in \mathcal{X}} u(\mathbf{x}|\mathcal{D}_t)$.
    Augment the data $\mathcal{D}_{t+1} = \{\mathcal{D}_t, (\mathbf{x}_{t+1}, f(\mathbf{x}_{t+1}))\}$
**end for**

**Algorithm 2** REMBO: Bayesian Optimization with Random Embedding

Generate a random matrix $\mathbf{A} \in \mathbb{R}^{D \times d}$
Choose the bounded region set $\mathcal{Y} \subset \mathbb{R}^d$
**for** $t = 1, 2, \ldots$ **do**
    Find $\mathbf{y}_{t+1} \in \mathbb{R}^d$ by optimizing the acquisition function $u$: $\mathbf{y}_{t+1} = \arg\max_{\mathbf{y} \in \mathcal{Y}} u(\mathbf{y}|\mathcal{D}_t)$.
    Augment the data $\mathcal{D}_{t+1} = \{\mathcal{D}_t, (\mathbf{y}_{t+1}, f(\mathbf{A}\mathbf{y}_{t+1})\}$
    Update the kernel hyper-parameters.
**end for**

# Scaling to over a billion dimensions



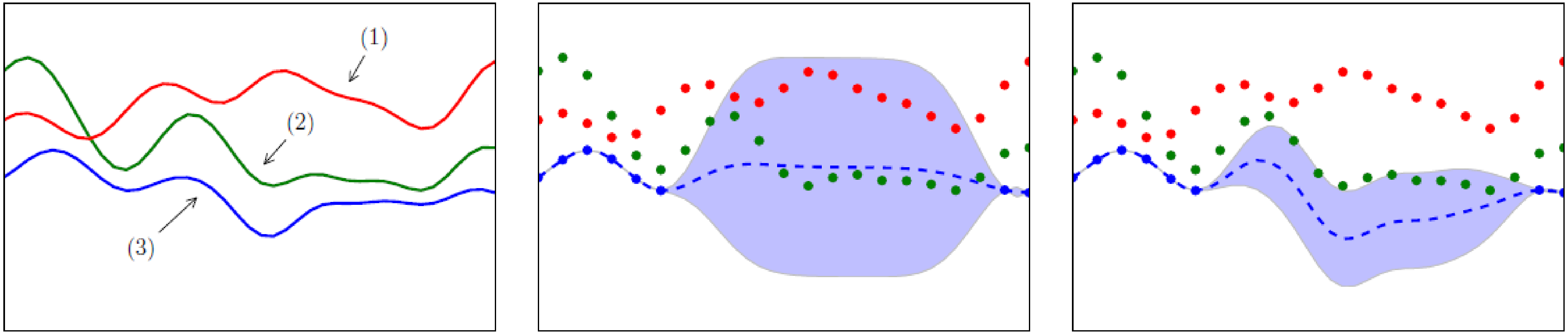

$D = 25$

$D = 1\,000\,000\,000$

# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Multi-task




- Define a kernel over tasks (ouputs) and inputs. E

$$k((\mathbf{x}, m), (\mathbf{x}', m')) = k_{\mathcal{X}}(\mathbf{x}, \mathbf{x}')k_{\mathcal{T}}(m, m')$$

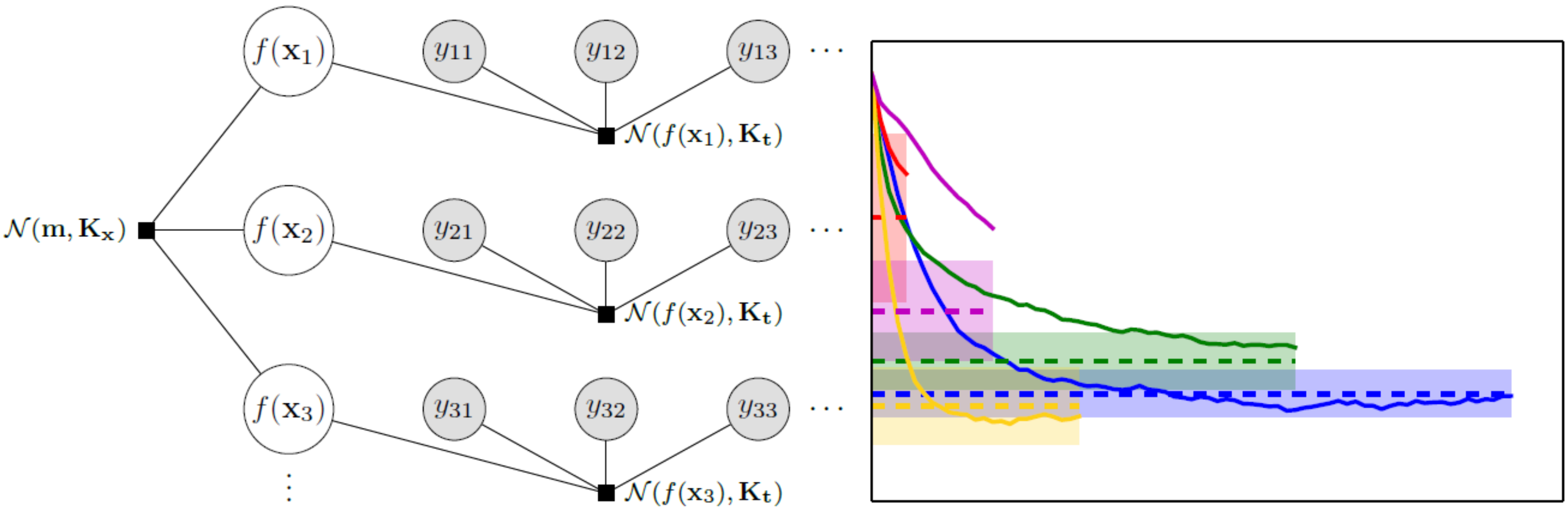
- Can also do BayesOpt with many parametric approaches to contextual / multi-task regression. E.g. context could be features of problem at hand

# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Freeze-thaw – learning curve

$$k(t,t') = \int_0^\infty e^{-\lambda t} e^{-\lambda t'} \psi(d\lambda)$$




(a) Graphical Model (b) Training curve predictions

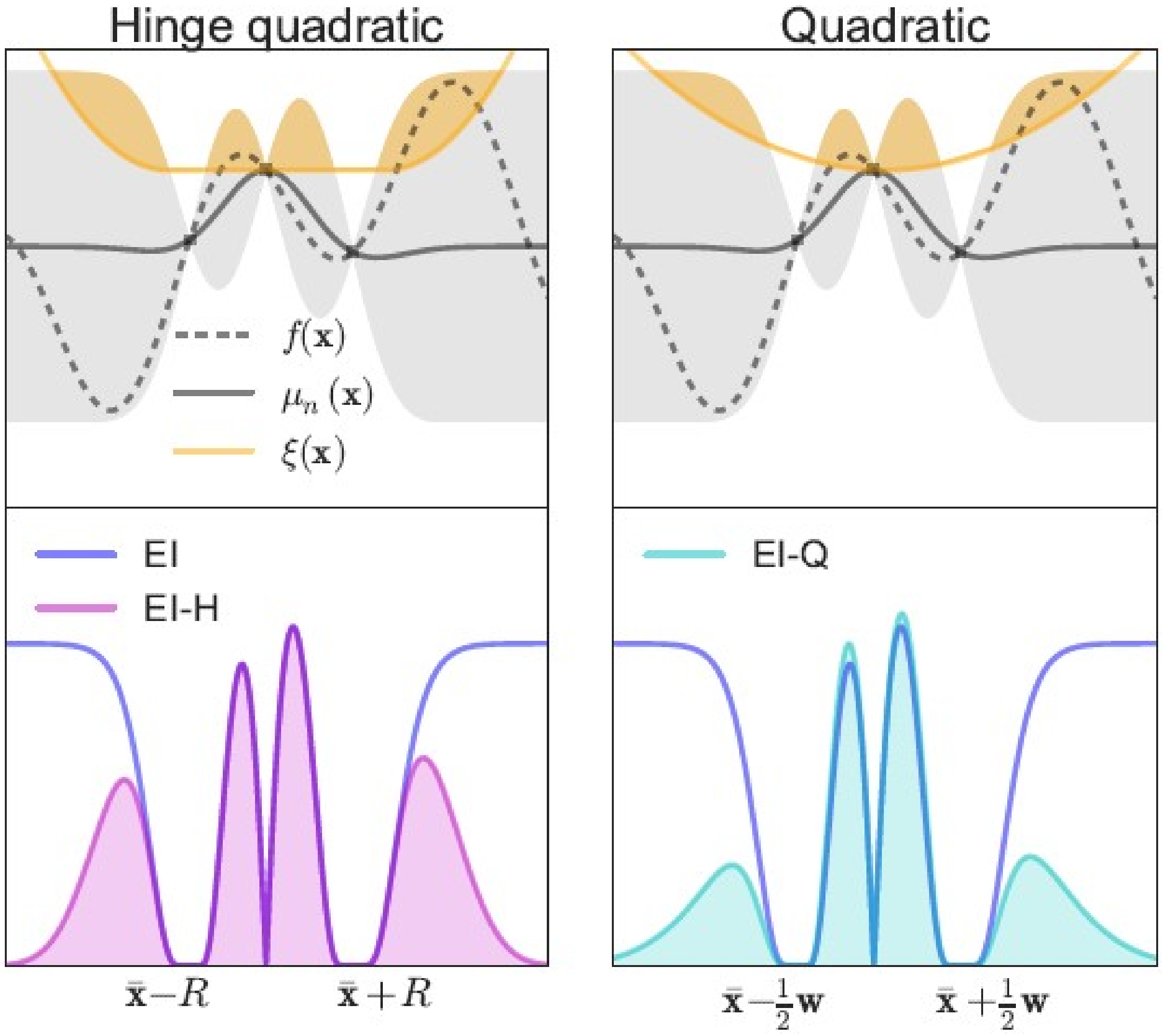
$$P(\{\mathbf{y}_n\}_{n=1}^N \mid \{\mathbf{x}_n\}_{n=1}^N) = \int \left[ \prod_{n=1}^N \mathcal{N}(\mathbf{y}_n; f_n \mathbf{1}_n, \mathbf{K}_{tn}) \right] \mathcal{N}(\mathbf{f}; \mathbf{m}, \mathbf{K}_\mathrm{x})\, d\mathbf{f}$$

[Kevin Swersky et al, See also work of David Ginsbourger and Frank Hutter]

# Outline

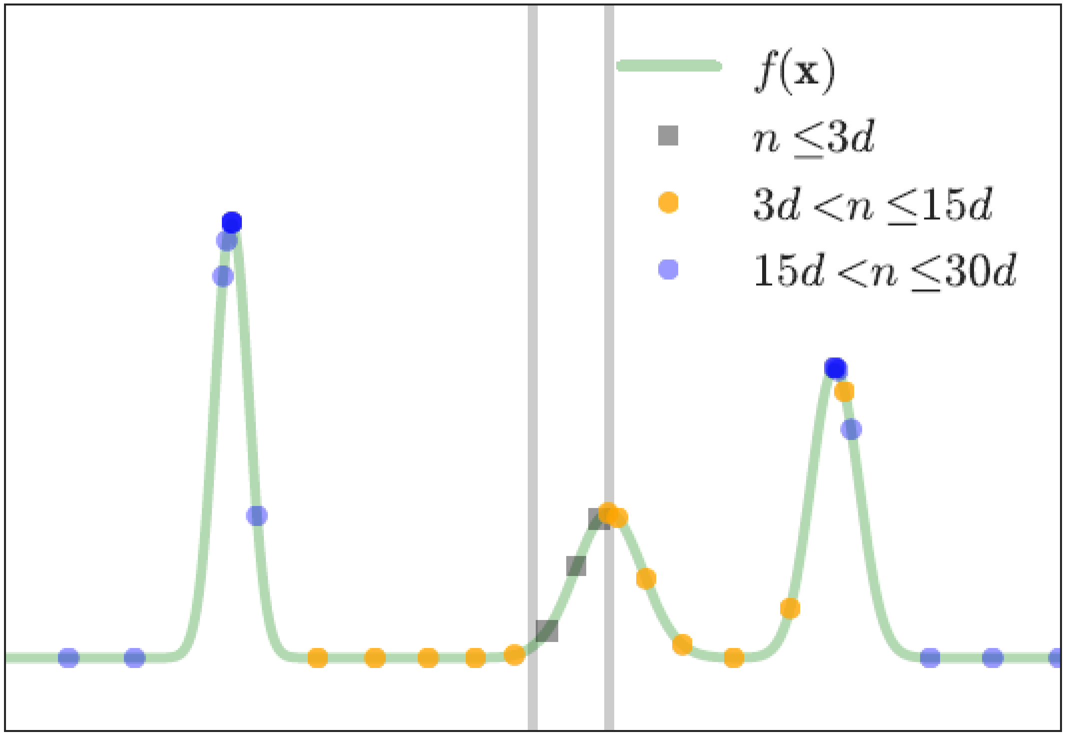
- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

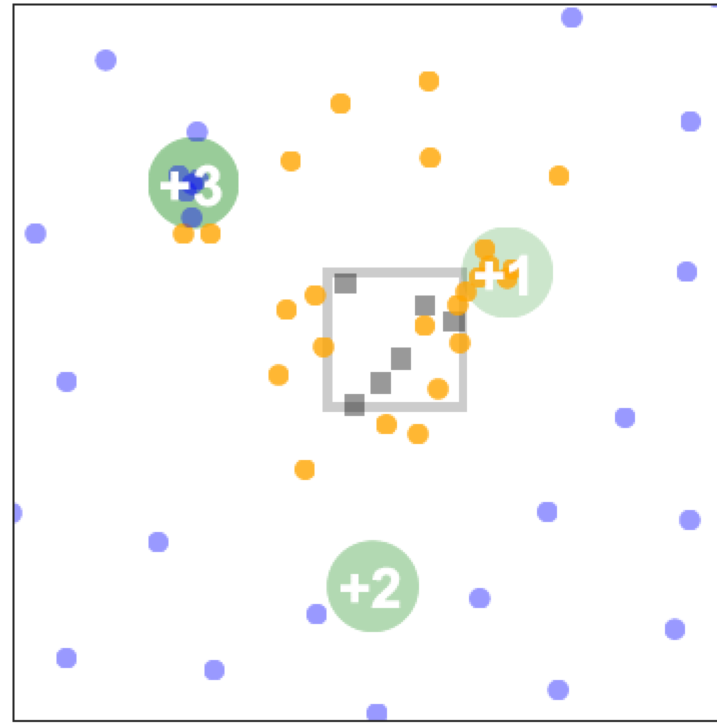
# Unknown optimization boundary




[Bobak Shahriari et al, 2015

# Unknown optimization boundary




[Bobak Shahriari et al, 2015

# Outline

- **Some applications**
- **Parametric Bayesian optimization**
  - Beta-Bernoulli bandit models
  - Linear bandit models
  - Neural network and other feature-based models
- **Non-Parametric Bayesian optimization**
  - Gaussian processes
  - Random forests
- **Acquisition functions**
- **A huge bag of problems**
  - Hyper-parameters and robustness
  - Optimizing acquisition functions
  - Conditional spaces
  - Non-stationarity
  - Parallelization
  - Constraints and cost sensitivity
  - High-dimensions
  - Multi-task / context
  - Freeze-thaw / early stopping
  - Unknown optimization regions
  - Empirical hardness models and variants

# Empirical Hardness Models

- Formally, given a set of particular problem instances, $p \in P$, and an algorithm, $a \in A$ (with free-parameters), the objective of performance prediction is to build a model that predicts the performance of $a$ when run on an arbitrary $p \in P$. See work of Kevin Leyton-Brown and Holger Hoos.

# Thank you

joinus@deepmind.com