BAYESIAN OPTIMIZATION FOR AUTOMATED MODEL SELECTION

Gustavo Malkomes
Chip Schaff
Roman Garnett
Washington University in St. Louis

Metalearning symposium, NIPS 2017, 12/7/17

VISION

Automated active learning

Active learning

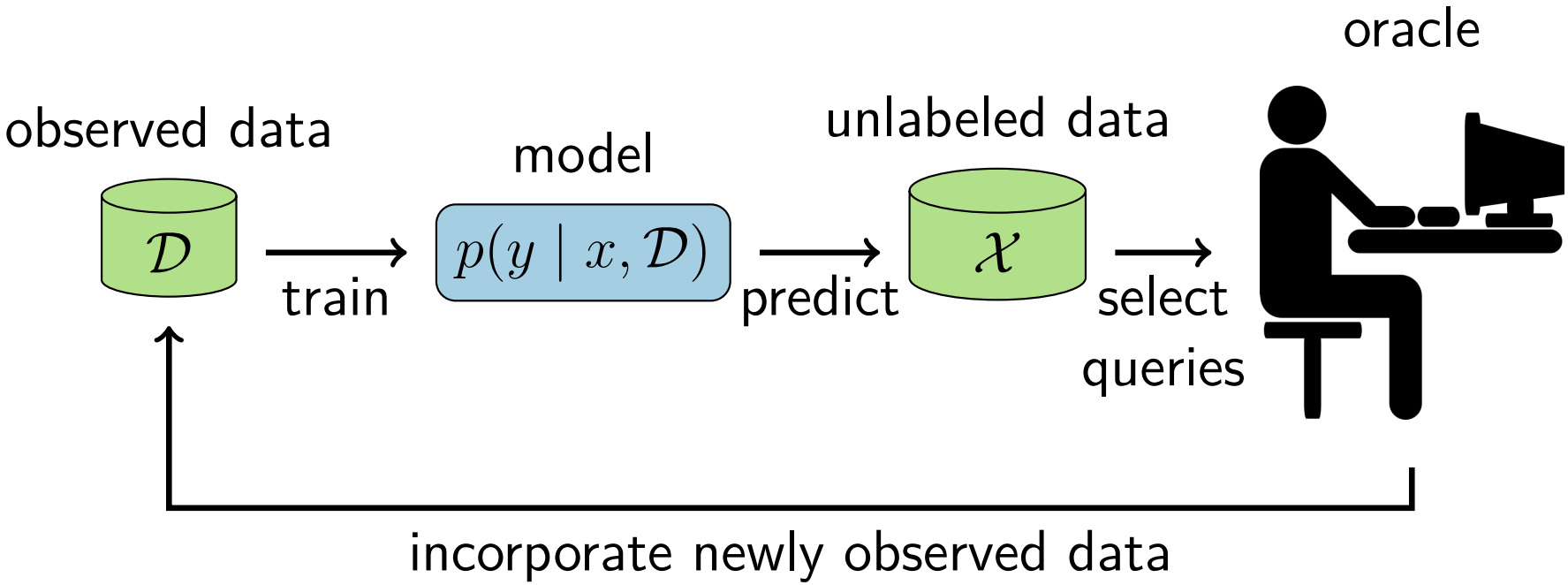
- When obtaining new data is expensive (e.g., requiring human intervention, expensive simulation, or costly experiments), we should *think carefully* about the data we obtain.
- *Active learning* (also Bayesian optimization, active search, etc.) considers how to obtain data to achieve your goals efficiently.

Application areas

Such problems are *pervasive* in the natural sciences and engineering, in machine learning, etc.:

- drug/materials discovery,
- biology,
- astronomy,
- *hyperparameter optimization,* etc.

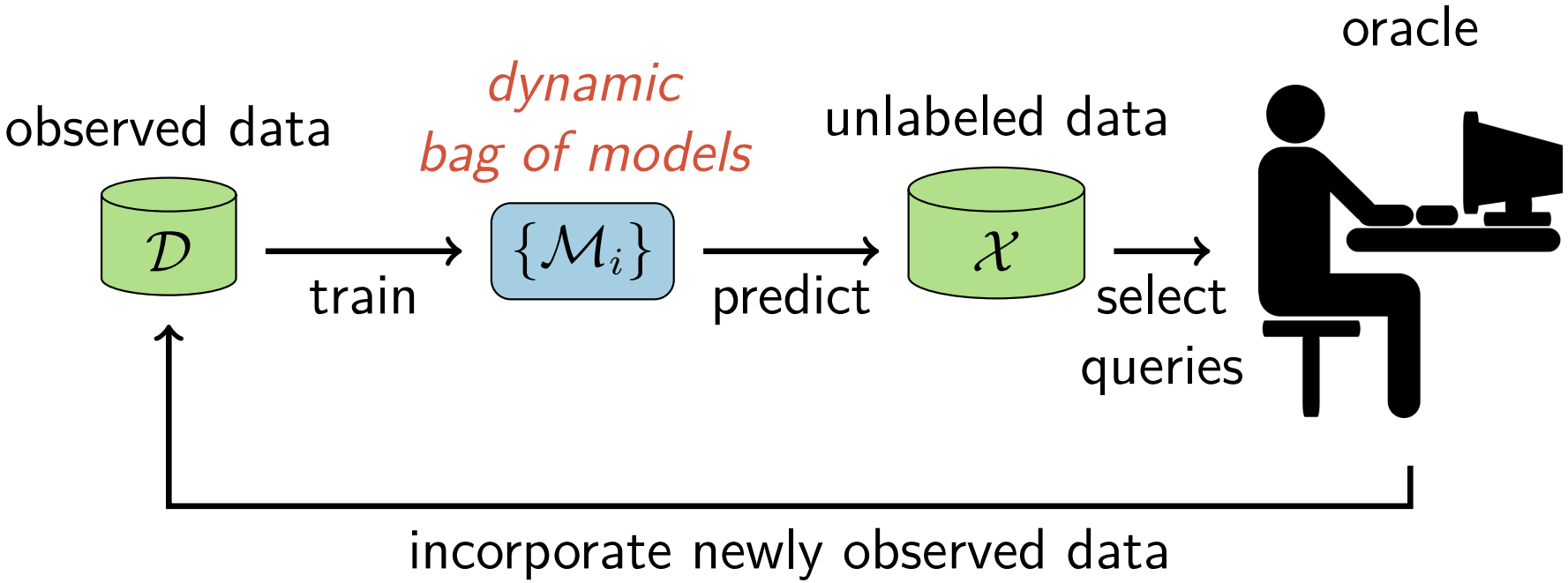
Active learning



Motivating question

- How is a *nonexpert* facing such problems to adopt the tools machine learning is producing?
  - Read the 7100-page NIPS 2017 proceedings?
  - Get a human expert in machine learning?
  - Pick something off the shelf?
- How can we *remove the human from the loop?* By making an *automated* off-the-shelf option.

Automated active learning proposal



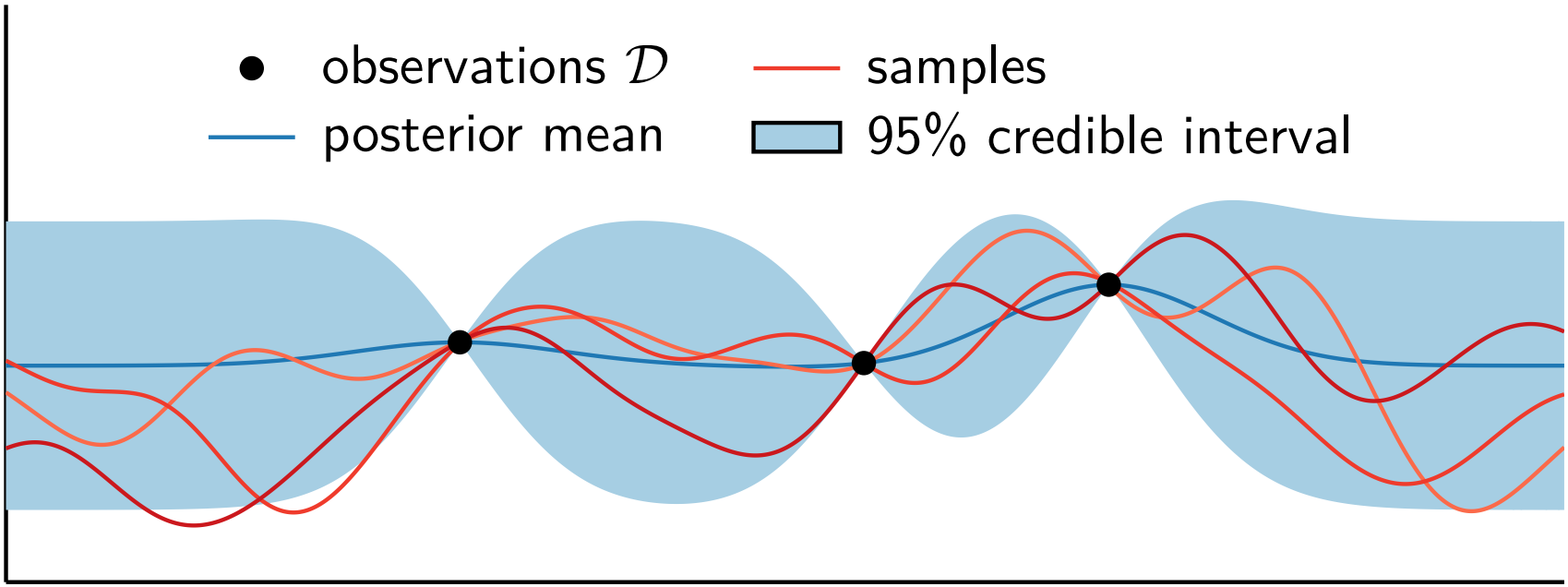
BAYESIAN OPTIMIZATION

Bayesian optimization

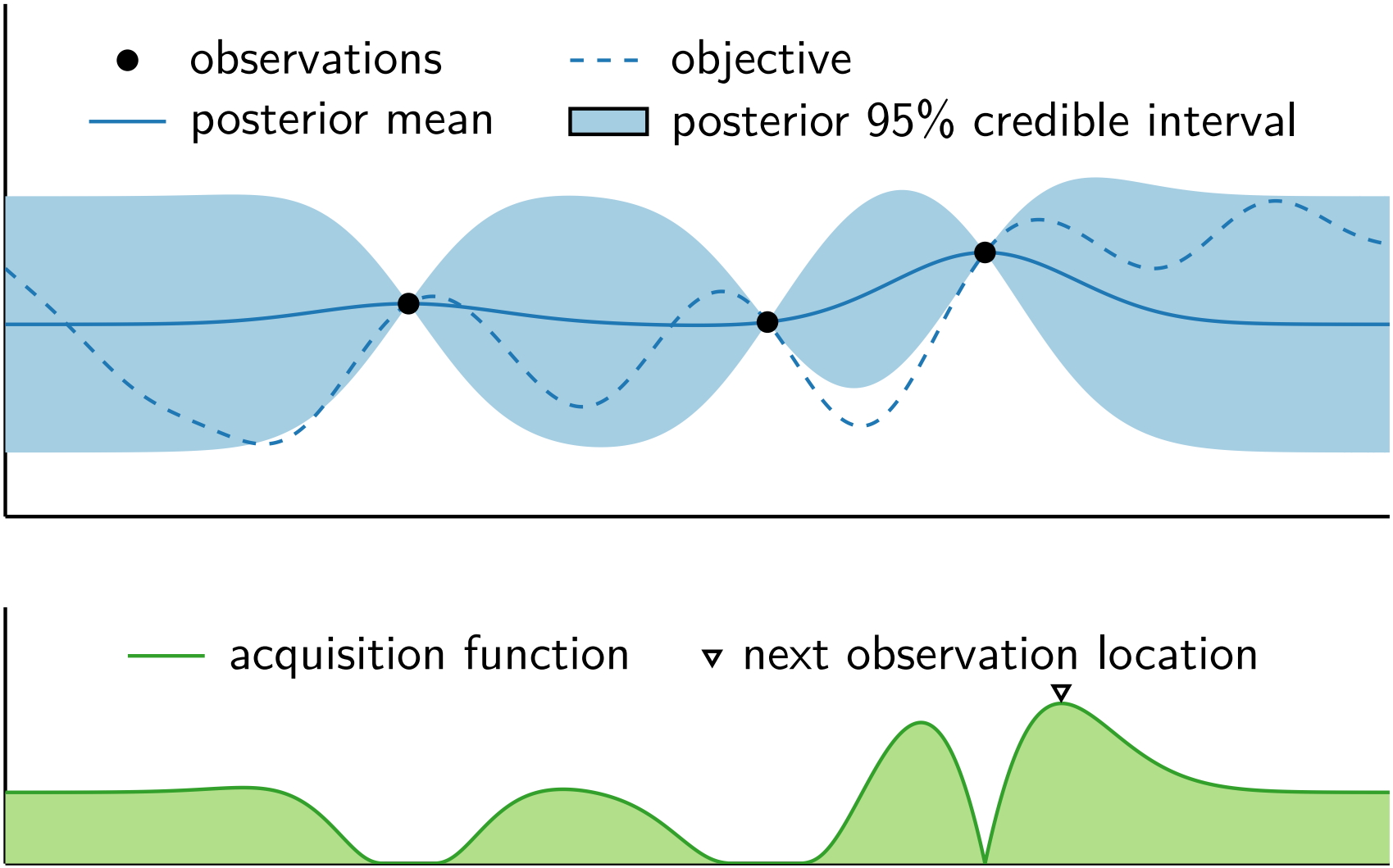
- *Bayesian optimization* is a powerful framework for optimizing *expensive-to-evaluate* functions that has proven successful in many domains.
- With an *informative* model of the objective function, Bayesian optimization can be *extremely sample efficient.*
- Objectives can be
  - nonconvex,
  - observed without gradients,
  - "black boxes."

Model of objective (posterior)

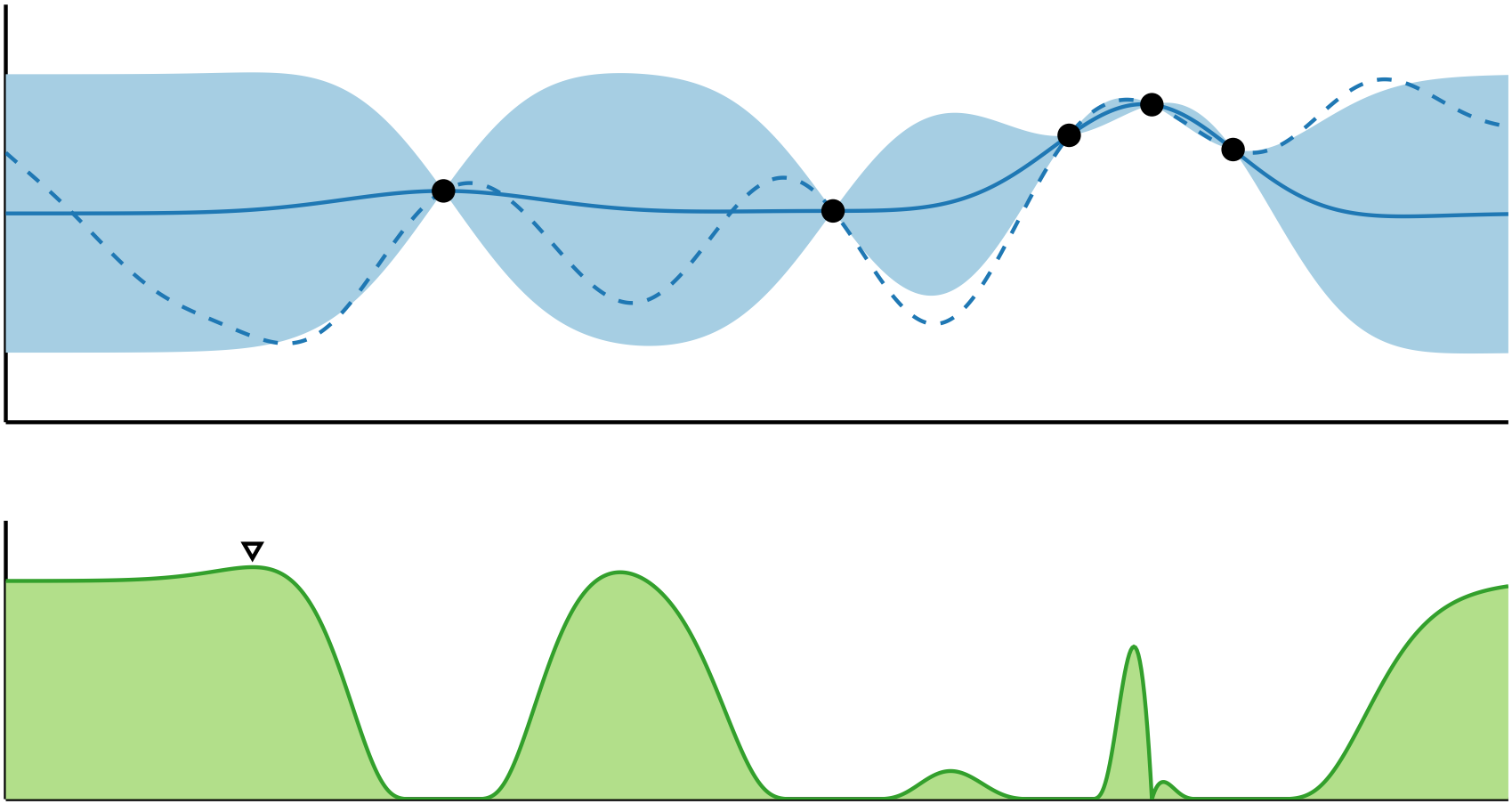
A *Gaussian process model* is typically chosen:



Derive an optimization policy via e.g. *Bayesian decision theory*

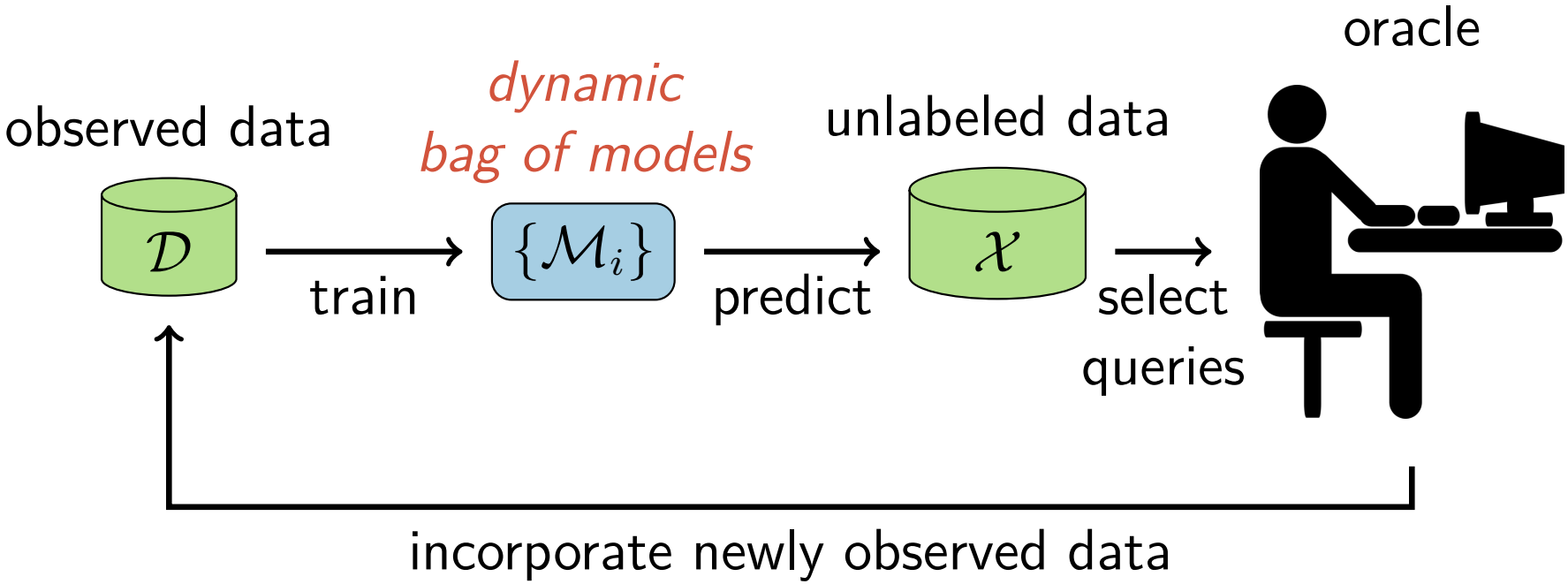


Many policies automatically trade off *exploration/exploitation*



AUTOMATED MODEL SEARCH

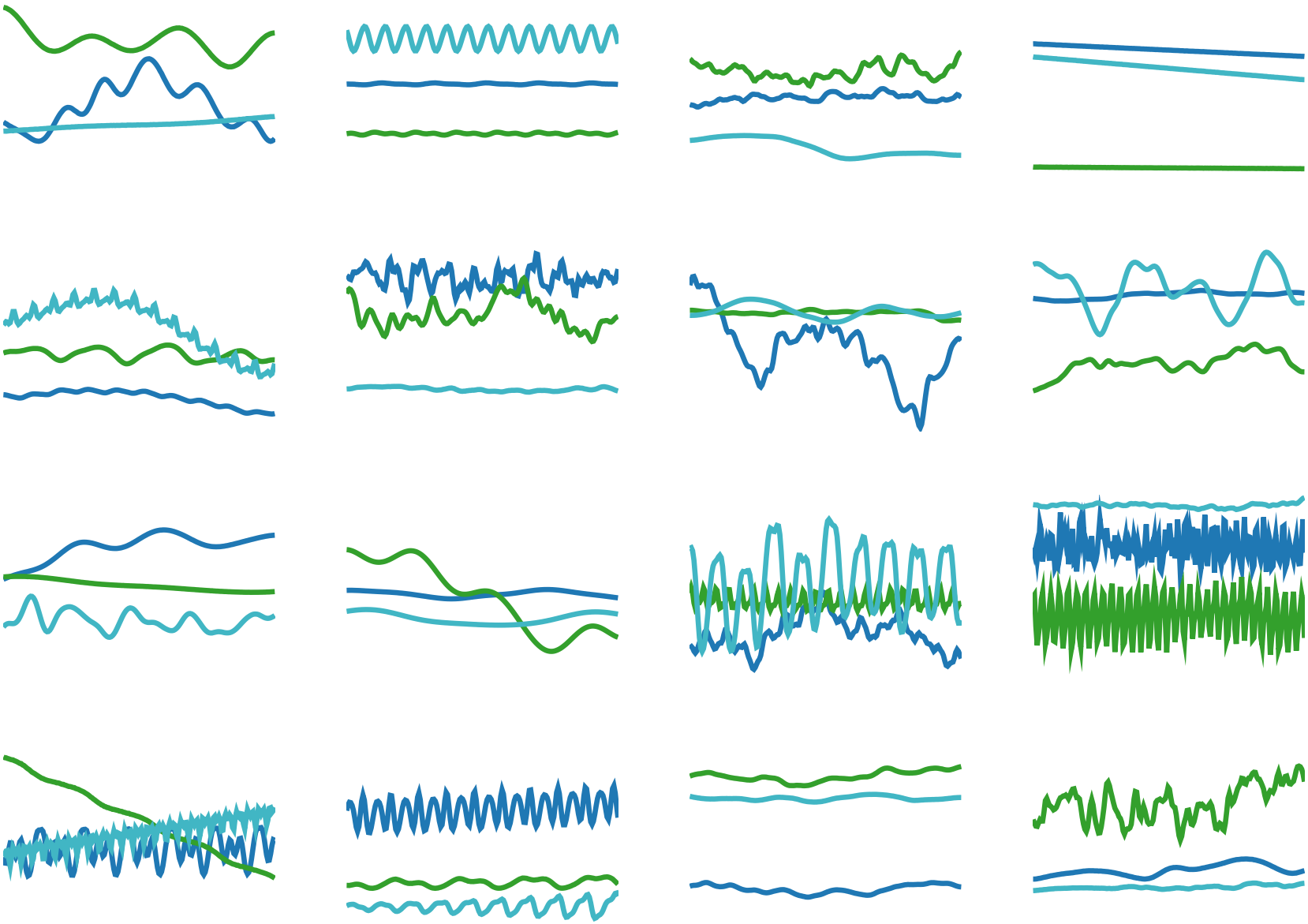
Automated active learning proposal



Structured model spaces

We will automatically search a space of models to find a good explanation of *fixed* dataset.

Here we focus on *Gaussian process models,* which are useful for active learning and can express rich *structure* in data.



Objective function

In the Bayesian formalism, given a dataset $\mathcal{D}$, we measure the quality of a model $\mathcal{M}$ using the (log) *model evidence:*

$$g(\mathcal{M}; \mathcal{D}) = \log \int p(\mathbf{y} \mid \mathbf{X}, \theta, \mathcal{M}) p(\theta \mid \mathcal{M}) \, \mathrm{d}\theta.$$

Optimization problem

We may now frame model search as an *optimization problem.* We seek

$$\mathcal{M}^* = \arg\max_{\mathcal{M} \in \mathbb{M}} g(\mathcal{M}; \mathcal{D}).$$

Where $\mathbb{M}$ is some appropriate space of models, e.g.:

- compositional kernel grammar (Duvenaud, et al. ICML 2013)
- additive decompositions (Kandasamy, et al. ICML 2013)

OBSTACLES

Why this is a hard problem

The objective is nonlinear and nonconvex

- The mapping from models to evidence is *highly complex!*
- Even seemingly "similar" models can offer *vastly different* explanations of the data.
- . . . and this similarity depends on the *geometry* of the data!
- Imagine a bunch of isolated points. . .

The objective is expensive

Even estimating the model evidence is *very expensive.*

Easily $\mathcal{O}(1000|\mathcal{D}|^3)$!

The domain is discrete

Another problem is that the space of models is *discrete;* therefore we can't compute *gradients* of the objective.

BAYESIAN OPTIMIZATION?

Why not?

A case for Bayesian optimization!

We have a

- nonlinear,
- gradient-free,
- expensive,
- optimization problem...

...*Bayesian optimization!*

Overview of approach

We model the (log) model evidence function with a *Gaussian process in model space:*

$$p\big(g(\mathcal{M};\mathcal{D})\big) = \mathcal{GP}(g;\mu_g,K_g).$$

Then use a posterior belief to derive an efficient policy trading off exploration and exploitation in model space.

Evidence model

We need to construct an *informative prior* over the log model evidence function:

$$p\big(g(\mathcal{M};\mathcal{D})\big) = \mathcal{GP}(g;\mu_g,K_g).$$

For the mean, we simply take a constant...

...what about the covariance?

The "kernel kernel"

We consider two kernels to be "similar" for a given dataset $\mathcal{D}$ *if they offer similar explanations for the latent function at the observed locations.*

The "kernel kernel"

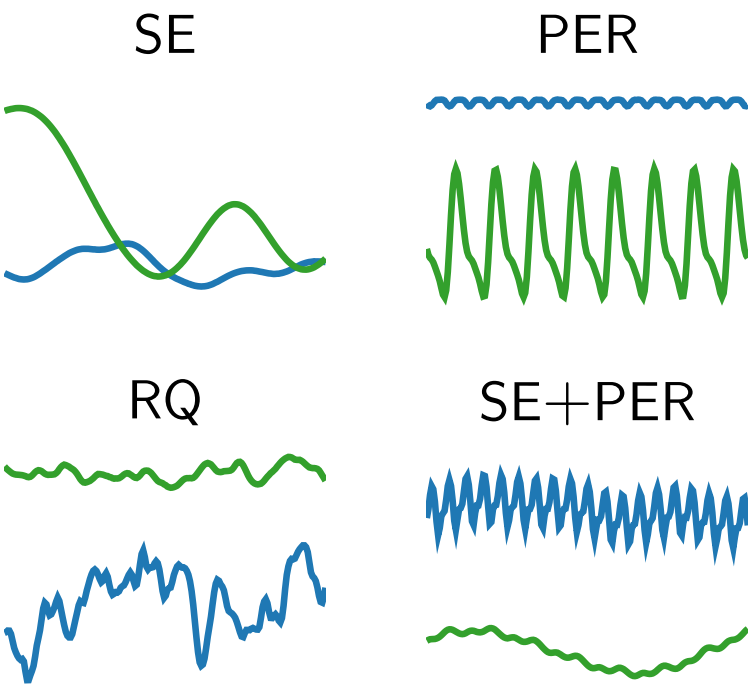
Given input locations $\mathbf{X}$, we suggest two models $\mathcal{M}$ and $\mathcal{M}'$ should be similar when the latent explanations

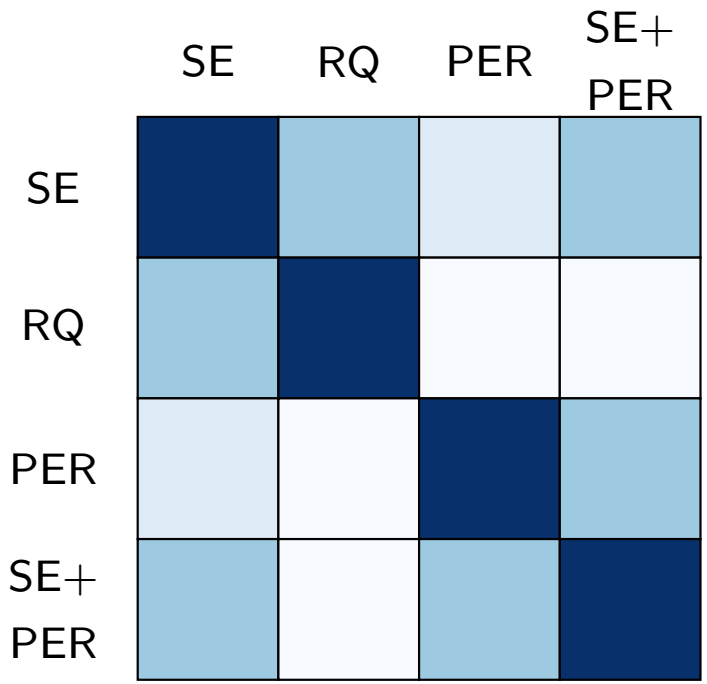
$$p(\mathbf{f} \mid \mathbf{X}, \mathcal{M}) \qquad p(\mathbf{f} \mid \mathbf{X}, \mathcal{M}')$$

are similar; i.e., they have *high overlap.*

Many details omitted!

“Kernel kernel:” Illustration

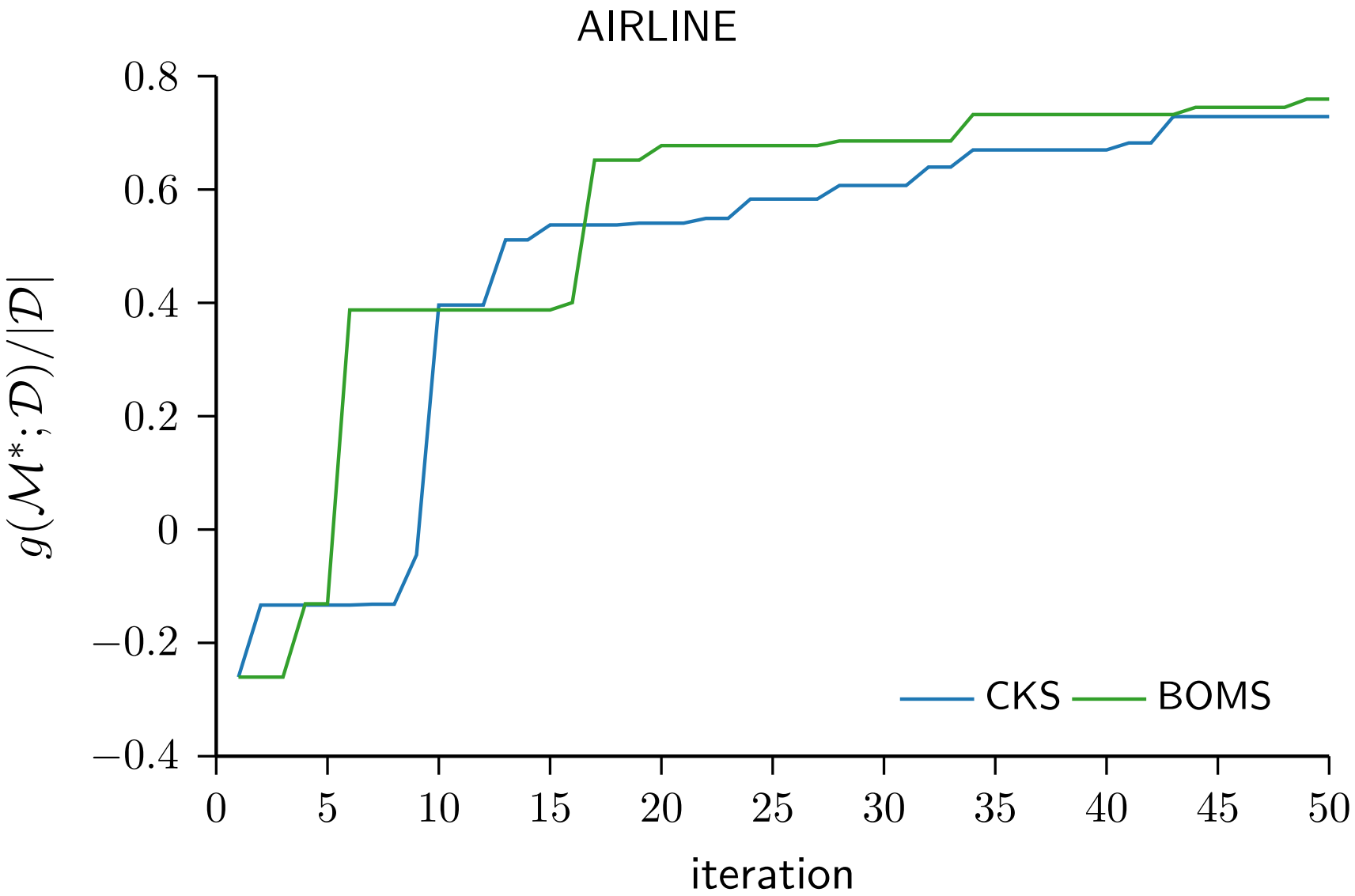




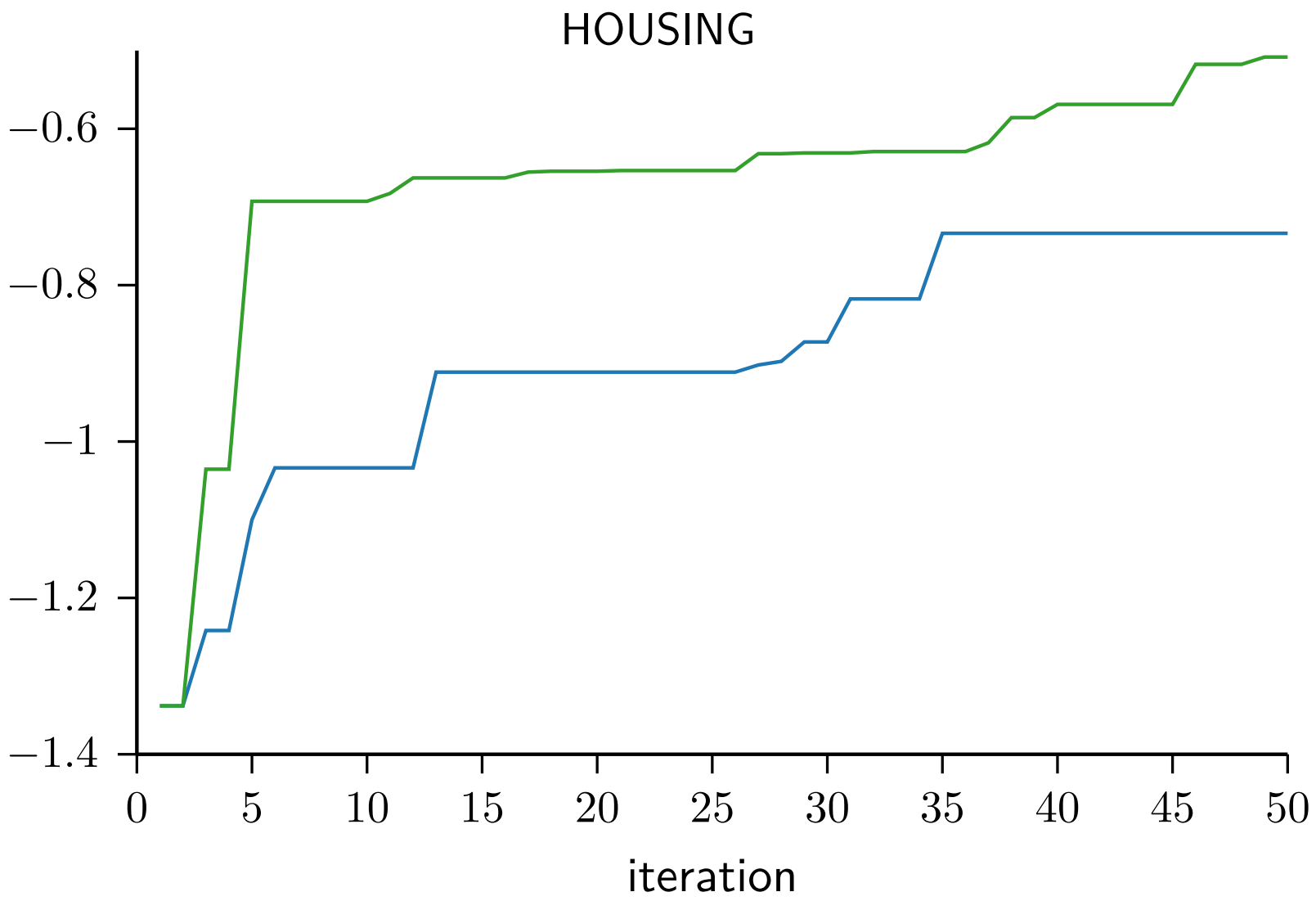
Experimental setup

- We compare our method (*Bayesian optimization for model selection, BOMS*) against the greedy search method from Duvenaud, et al. ICML 2013.
- Everything is the same: estimate of evidence, model space, etc.
- Budget of 50 model evidence computations.

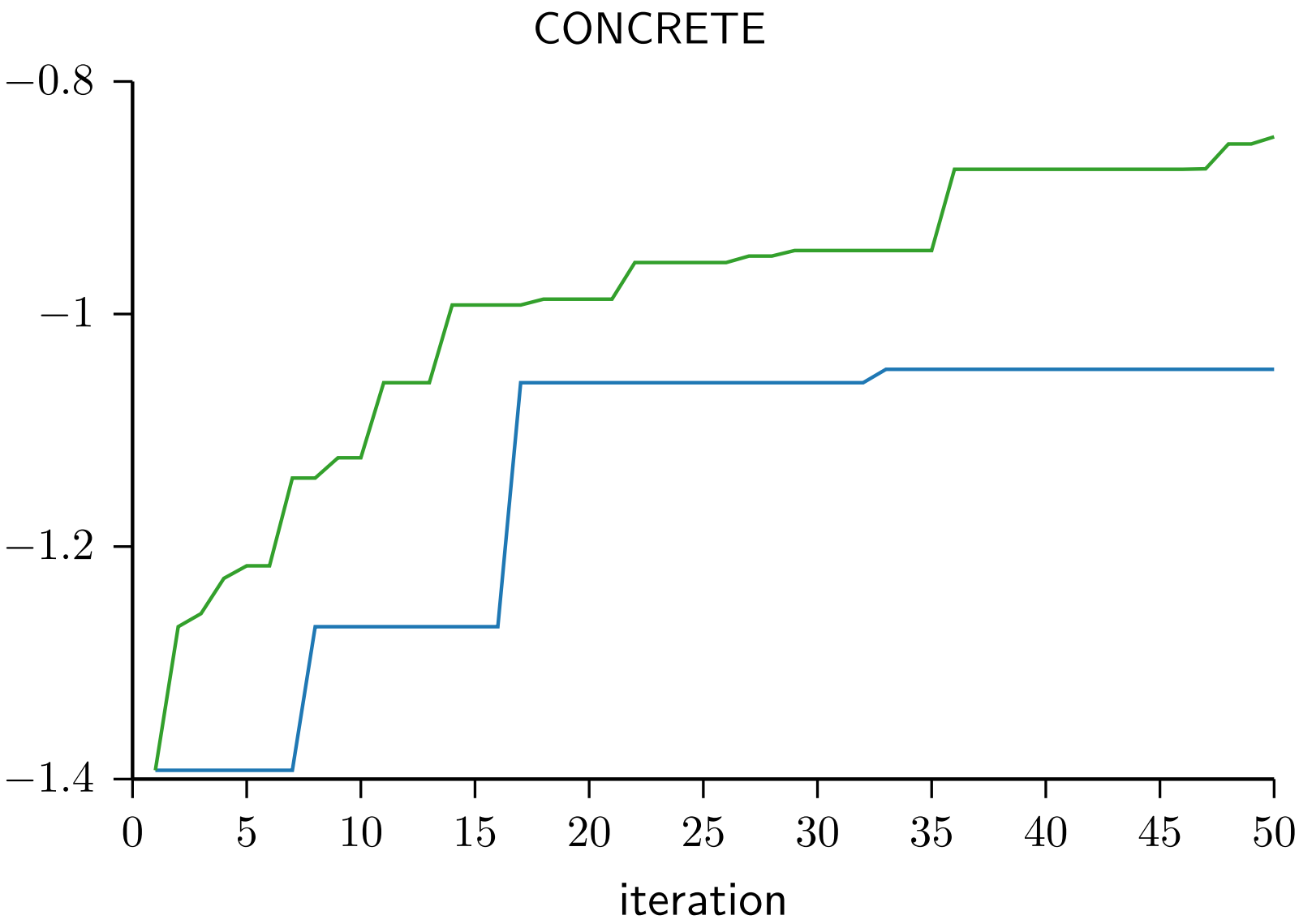
Results: Time series



Results: High-dimensional data



Results: High-dimensional data



Notes

- Our procedure is able to *rapidly* and *efficiently locate promising models* for fixed datasets compared to previous approaches (Duvenaud, et al. ICML 2013).
- *Much faster* in more-complex model spaces due to exploration/exploitation tradeoff.
- We offer some advice for automatically selecting reasonable *hyperparameter priors* for given data.

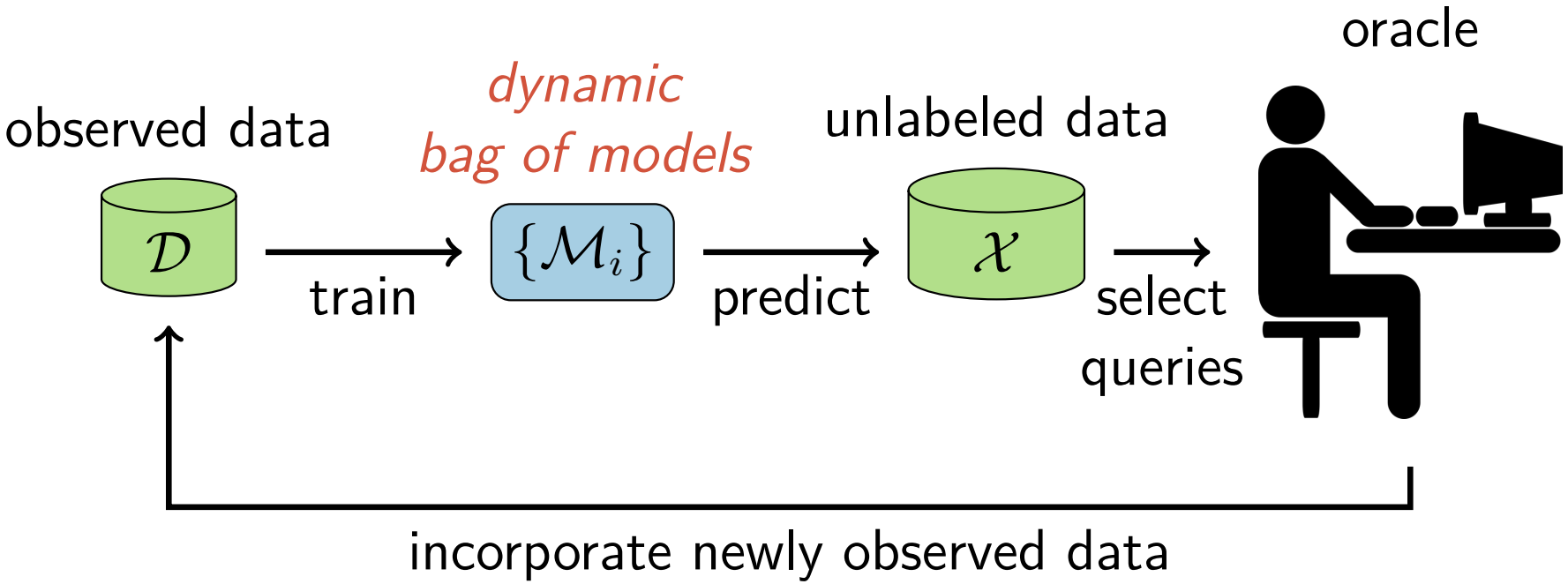
RETURNING TO THE VISION

Automating active learning

Looking forward

These results are promising, but the real promise of such methods is in the *inner loop* of another active learning procedure (e.g., Bayesian optimization)!

Active learning/Bayesian optimization



Initial study

We have completed an initial study of using our framework for *fully automated Bayesian optimization.*

Compared proposed method with off-the-shelf model (SE) and a *fixed* bag of models (BOM).

BAYESIAN OPTIMIZATION FOR AUTOMATING BAYESIAN OPTIMIZATION

(talk about meta)

Results: Average gap (high = better)

test function	SE	BOM	proposed
Ackley	0.79	0.37	**1.00**
Branin	0.86	0.96	**0.99**
drop-wave	0.37	0.43	**0.49**
McCormick	**1.00**	**1.00**	**1.00**
six-hump camel	0.70	0.85	**0.87**

Model-uncertainty aware methods (BOM and proposed) typically *outperform* off-the-shelf methods, with the proposed method performing the best on all functions.

Table 2: Results for the average GAP performance measure over different test functions. Experiments with high dimension functions ($d > 3$) were replicated 10 times, whereas the results for lower dimensional functions are average across 20 repetitions. In all experiments, five initial points were selected at random and the number of function evaluations was limited to 10 times the dimension d of the input domain. The best result for each function is bolded. The mean and median GAP performance over all functions are shown at the bottom.

Functions	d	SE	SE_{ARD}	RQ	BOM	MCMC	ABO
Ackley 2-D	2	0.758	0.724	**0.995**	0.959	0.977	0.992
Beale	2	**0.717**	0.643	0.679	0.666	0.69	0.652
Branin	2	0.841	0.919	0.867	0.982	0.927	**0.989**
Bukin N6	2	0.65	0.728	**0.871**	0.814	0.833	0.825
Six-Hump Camel	2	0.699	0.608	0.793	**0.811**	0.735	0.788
Drop-Wave	2	**0.652**	0.559	0.301	0.458	0.578	0.442
Eggholder	2	0.542	0.537	**0.576**	0.531	0.508	0.54
Goldstein-Price	2	**0.822**	0.691	0.774	0.682	0.703	0.648
Griewank 2-D	2	0.859	**0.916**	0.875	0.915	**0.916**	0.902
Mccormick	2	1	1	1	1	1	1
Rastrigin	2	0.454	0.325	0.815	0.886	0.886	**0.892**
Rosenbrock	2	0.744	0.954	0.784	0.992	0.73	**0.995**
Shubert	2	0.34	**0.503**	0.21	0.322	0.231	0.49
Hartmann 3-D	3	0.97	0.979	0.959	0.984	**0.999**	0.995
Levy	3	0.744	0.612	0.917	0.859	0.9	**0.938**
Shekel M-10	4	661	0.580	0.509	**0.630**	0.613	0.460
Ackley 5-D	5	0.652	0.668	0.844	0.953	0.921	**0.987**
Griewank 5-D	5	0.986	0.986	0.988	0.978	**0.989**	0.972
Hartmann 6-D	6	0.963	0.983	0.978	0.977	0.985	**0.986**
Mean GAP		0.740	0.732	0.776	0.810	0.796	**0.815**
Median GAP		0.744	0.691	0.844	0.886	0.886	**0.902**

THANK YOU!

Questions?

But random search...

- With an *informative* model of the objective function, Bayesian optimization can be *extremely sample efficient.*
- With an *weakly informative* model of the objective function, Bayesian optimization can devolve to a slightly smarter random search.
- For example, an off-the-shelf kernel in high dimensions is going to face the *curse of dimensionality.*
- But this is just an argument for finding better models! (Low-dimensional structure, additive structure, etc.), which can give *massive* (exponential) speedups.