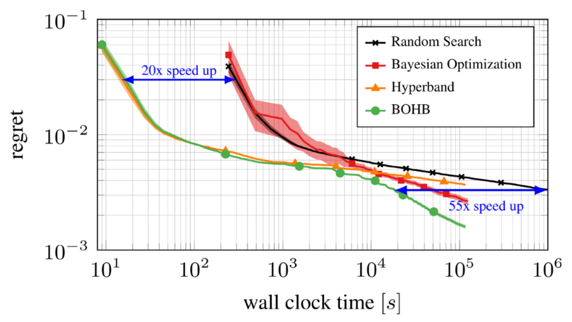
Practical Two-Step Look-Ahead Bayesian Optimization

Jian Wu (Cornell) **Peter I. Frazier** (Cornell, Uber) Funding from AFOSR, NSF

Overview

- **What is BayesOpt?** A class of ML methods for optimizing black-box time-consuming-to-evaluate functions without derivatives. BayesOpt methods use supervised learning (typically a Gaussian process) to model the objective function, and an acquisition function to decide where to evaluate it.
- **Important BayesOpt Applications:**



AutoML
[automl.org]

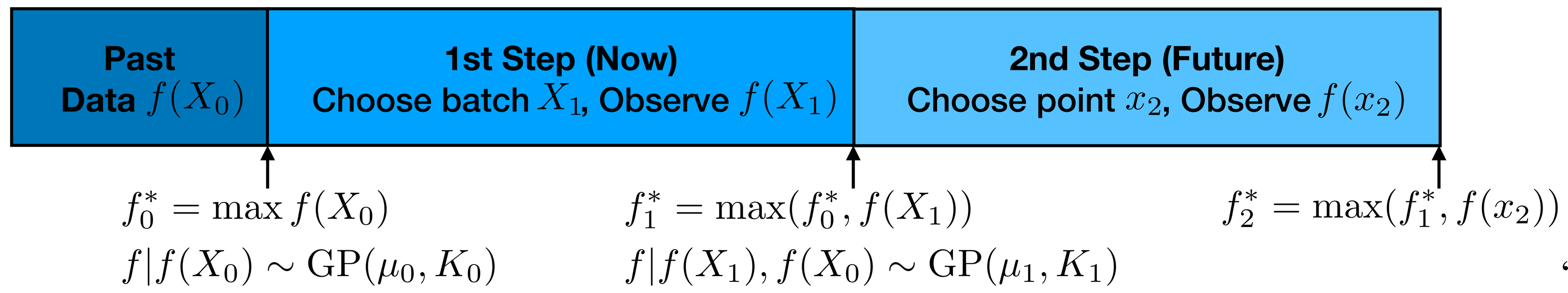
Policy Search in Robotics
[Tesch, Schneider, Choset 2011]

Product Improvement with A/B testing
[botorch.org]

- **Problem:** Acquisition functions fast enough to be used in practice look just 1 step ahead. This hurts their query efficiency.
  - For example, expected improvement, $\mathrm{EI}(x) = E[\max(f(x), f^*)] - f^*$ compares the solution quality now (f^*) to the quality after one more evaluation $E[\max(f(x), f^*]$
  - Predictive entropy search and probability of improvement also look just 1 step ahead. Knowledge gradient looks 1.5 steps ahead.
- **Past Attempts:** Look $>$ 1 steps ahead using general-purpose RL [González, Osborne & Lawrence AISTATS 2016, Lam, Wolcox & Wolpert NIPS 2016]
  - These methods are slow (100s of minutes per evaluation)
  - Approx. errors erase most of the benefit of looking $>$ 1 steps ahead
- **Our Contribution:** A new algorithm that efficiently & accurately optimizes the two-step lookahead acquisition function. It provides:
  - Better query efficiency than 1-step and past multi-step methods
  - It is fast enough to be practical: seconds to at most several minutes per batch; comparable to knowledge-gradient acquisition function; 10x faster than previous multi-step methods
- **Code:**
  - Method: https://github.com/wujian16/Cornell-MOE
  - Experiments: https://github.com/wujian16/TwoStep-BayesOpt

2-OPT Acquisition Function

- **Setting**: Maximize $f(x)$ using noise-free evaluations of f.
- **Approach**: Evaluate the quality of a batch X_1, looking 2 steps ahead.
  - Step 1: Choose a batch X_1 of points to evaluate.
  - Step 2: Assume we'll choose one more point, x_2, based on $f(X_1)$, and our overall solution quality will be the best point seen, f_2^*.



- **2-OPT Acquisition Function**: The expected 2-step improvement $E[f_2^* - f_0^*]$ for a batch X_1 is

$$\text{2-OPT}(X_1) = \mathrm{EI}_0(X_1) + \mathbb{E}_0\left[\max_{x_2} \mathrm{EI}_1(x_2)\right],$$

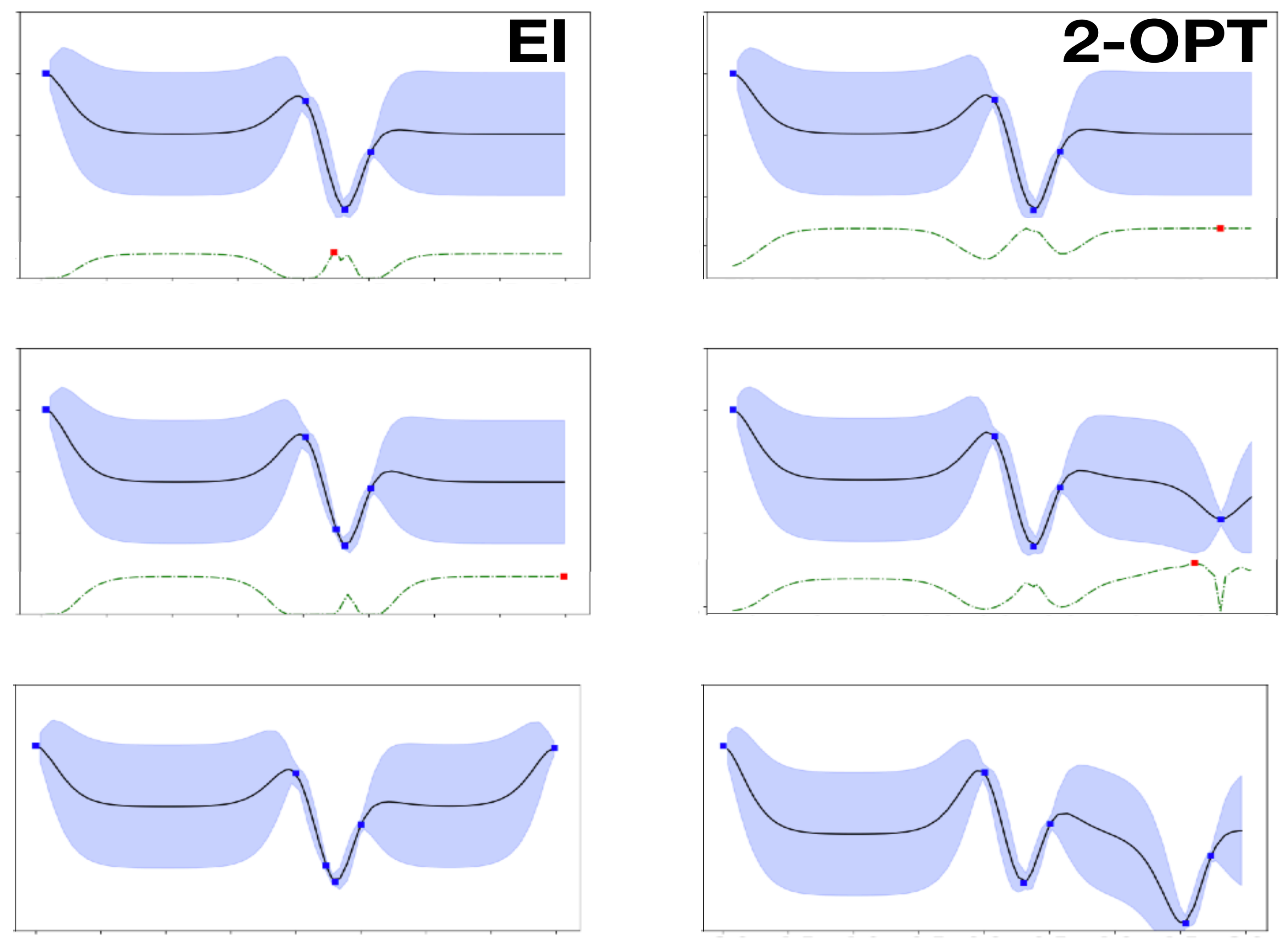
  - $\mathrm{EI}_n(x) = \mathrm{EI}(\mu_n(x) - f_n^*, K_n(x,x))$ is the expected improvement of evaluating at x under the step n posterior.
  - $\mathrm{EI}(m, v) = m\Phi(m/\sqrt{v}) + \sqrt{v}\varphi(m/\sqrt{v})$.

We Can Maximize 2-OPT Quickly

- **Approach:** Multistart SGD, where SGD uses a novel fast unbiased Monte Carlo estimator of ∇2-OPT(X_1)
- **Novel fast Monte Carlo estimator of the gradient of 2-OPT**(X_1)

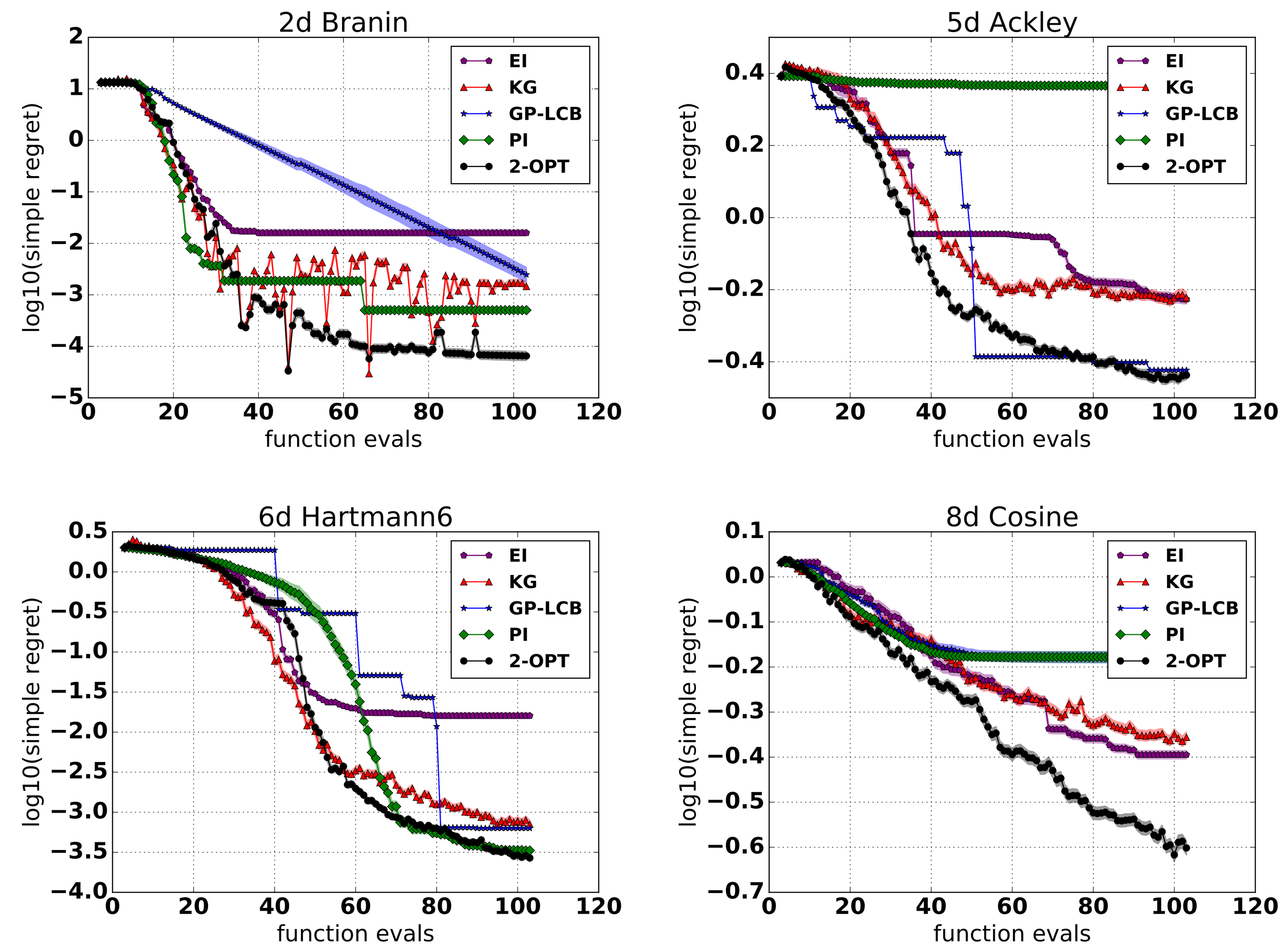
1. Use the reparameterization trick for unbiased Monte Carlo estimation of 2-OPT, where $C_0(X_1)$ is the Cholesky decomposition of $K_0(X_1, X_1)$, $Z \sim N(0, 1)$,
$$\widehat{\text{2-OPT}}(X_1, Z) = \max(f_0^* - \mu_0(X_1) - C_0(X_1)Z)^+ + \max_{x_2} \mathrm{EI}_2(X_1, x_2, Z).$$
2. Use infinitesimal perturbation analysis to exchange gradient and expectation:
$$\nabla_{X_1} \text{2-OPT}(X_1) = \mathbb{E}_0\left[\nabla_{X_1} \widehat{\text{2-OPT}}(X_1, Z)\right]$$
$$= \mathbb{E}_0\left[\nabla \max(f_0^* - \mu_0(X_1) - C_0(X_1)Z)^+ + \nabla \max_{x_2} \mathrm{EI}_2(X_1, x_2, Z)\right]$$
3. Use the envelope theorem to compute the last term quickly
$$\nabla_{X_1} \max_{x_2} \mathrm{EI}_2(X_1, x_2, Z) = \nabla_{X_1} \mathrm{EI}_2(X_1, x_2^*, Z),$$
where we can ignore the dependence of $x_2^* \in \mathrm{argmax}_{x_2} \mathrm{EI}_2(X_1, x_2, Z)$ on X_1 when calculating ∇_{X_1}.

- **2-OPT Explores More than EI**:



Numerical Results

- Synthetic functions, 90% quantile of log10 simple regret vs. common 1-step heuristics. 2-OPT is substantially more robust.



- 2-OPT is much faster than GLASSES, comparable to KG.