Bayesian Optimization with Finite Budget

Presenter: Shibo Li

Motivations

- BO: objective function is expensive to evaluate.
- Most BO algorithms are greedy(myopic): ignores the how the current design selected will affect the future steps: one-step optimal.
- Lookahead:
  - Aware of remaining evaluations and maximize long-term reward over several steps
- Difficulty: DP-essential
  - Uncountable states and uncountable controls.
  - Maximize nested maximization and expectations

Roadmap

Bayesian Optimization with a Finite Budget: An Approximate Dynamic Programming Approach

Remi R. Lam
Massachusetts Institute of Technology
Cambridge, MA
rlam@mit.edu

Karen E. Willcox
Massachusetts Institute of Technology
Cambridge, MA
kwillcox@mit.edu

NeurIPS 2016

BINOCULARS for Efficient, Nonmyopic Sequential Experimental Design

Shali Jiang [*1] **Henry Chai** [*1] **Javier González** [2] **Roman Garnett** [1]

ICML 2020

Efficient Nonmyopic Bayesian Optimization via One-Shot Multi-Step Trees

Shali Jiang*
Washington University
jiang.s@wustl.edu

Daniel R. Jiang*
Facebook
drjiang@fb.com

Maximilian Balandat*
Facebook
balandat@fb.com

Brian Karrer
Facebook
briankarrer@fb.com

Jacob R. Gardner
University of Pennsylvania
gardner.jake@gmail.com

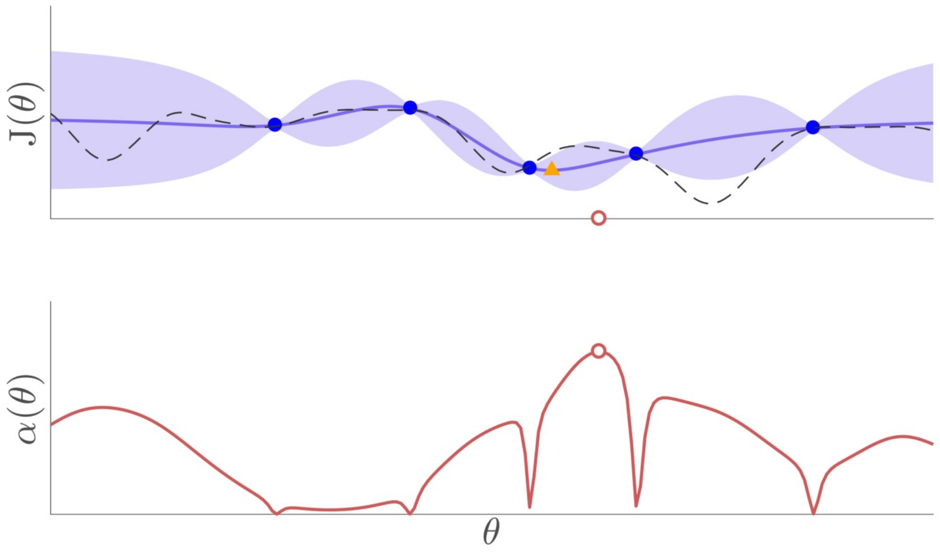
Roman Garnett
Washington University
garnett@wustl.edu

NeurIPS 2020

Bayesian Optimization (BO)

- Probabilistic Surrogate Modeling
  - GP, ABLR, BNN, Random Forest

- Acquisition Function
  - Explicitly/implicitly trade-off between exploration-exploitation
  - PI, EI, ES, UCB, MES



Practically, t is set to the optimums in the querying history

BO Formulation

$$\text{(OP)} \quad \boldsymbol{x}^* = \operatorname{argmin}_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$$

Expensive

$$f \sim \mathcal{G}(m, \kappa)$$

$$\begin{aligned}\overline{\mu}_k(\boldsymbol{x}) &= K(X_k, \boldsymbol{x})^\top [K(X_k, X_k) + \lambda I]^{-1} Y_k, \\ \overline{\sigma}_k^2(\boldsymbol{x}) &= \kappa(\boldsymbol{x}, \boldsymbol{x}) - K(X_k, \boldsymbol{x})^\top [K(X_k, X_k) + \lambda I]^{-1} K(X_k, \boldsymbol{x})\end{aligned}$$

$$\mathcal{S}_{k+1} = \mathcal{S}_k \cup \{(\boldsymbol{x}_{k+1}, y_{k+1})\}$$

$$\text{(AP)} \quad \boldsymbol{x}_{k+1} = \operatorname{argmax}_{\boldsymbol{x}\in\mathcal{X}} U_k(\boldsymbol{x}; \mathcal{S}_k)$$

Utility/Acquisition

Cheap

BO with Finite Budget as DP

- Goal:
    - Statistical model to represent the objective function
    - System dynamics that describes how this statistical model is updated as new information is collected
    - A goal that can be quantified with **long-term** reward

Dynamic Programming

- A discrete-stage dynamic (discrete time) DP
  - State space: $z_k \in \mathcal{Z}_k$
  - Action/Control space: $u_k \in \mathcal{U}_k(z_k)$
  - Noise control consequence $w_k \in \mathcal{W}_k(z_k, u_k)$ $\mathbb{P}(\cdot | z_k, u_k)$
  - Evolves to a new state $z_{k+1} \in \mathcal{Z}_{k+1}$

$$\forall k \in \{1, \cdots, N\}, \forall (z_k, u_k, w_k) \in \mathcal{Z}_k \times \mathcal{U}_k \times \mathcal{W}_k, \quad z_{k+1} = \mathcal{F}_k(z_k, u_k, w_k)$$

  - Policy: $\boldsymbol{\pi} = \{\pi_1, \cdots, \pi_N\}$ $\pi_k : \mathcal{Z}_k \mapsto \mathcal{U}_k$, for $k = 1, \cdots, N$
  - Stage reward: $r_k : \mathcal{Z}_k \times \mathcal{U}_k \times \mathcal{W}_k \mapsto \mathbb{R}$

Dynamic Programming(Continue)

- A discrete-stage dynamic (discrete time) DP
  - Final reward: $r_{N+1} : \mathcal{Z}_{N+1} \mapsto \mathbb{R}$
  - Expected reward (Value) starting the initial state with given policy

$$J_{\boldsymbol{\pi}}(z_1) = \mathbb{E}\left[r_{N+1}(z_{N+1}) + \sum_{k=1}^{N} r_k(z_k, \pi_k(z_k), w_k)\right]$$

  - Optimal Policy: $J^*(z_1) = J_{\boldsymbol{\pi}^*}(z_1) = \max_{\boldsymbol{\pi} \in \Pi} J_{\boldsymbol{\pi}}(z_1)$
  - Bellman's principle of optimality: DP recursive backward from N to 1

$$J_{N+1}(z_{N+1}) = r_{N+1}(z_{N+1}),$$
$$J_k(z_k) = \max_{u_k \in \mathcal{U}_k} \mathbb{E}[r_k(z_k, u_k, w_k) + J_{k+1}(\mathcal{F}_k(z_k, u_k, w_k))]$$

Formulate BO with Finite Budget as DP

- BO as DP instance:
  - State space: z_k is $\mathcal{S}_k$
  - Action/control space: u_k is $\boldsymbol{x}_{k+1}$
  - Noisy Consequences: possible simulated values f_{k+1} of the objective function at $\boldsymbol{x}_{k+1}$

$$W_k \sim \mathcal{N}\left(\overline{\mu}_k(\boldsymbol{x}_{k+1}), \overline{\sigma}_k^2(\boldsymbol{x}_{k+1})\right)$$

$$\boxed{\begin{array}{c} f_{k+1} \\ y_{k+1} = f(\boldsymbol{x}_{k+1}) \end{array}}$$

  - New state

$$\mathcal{S}_{k+1} = \mathcal{S}_k \cup \{(\boldsymbol{x}_{k+1}, f_{k+1})\} = \mathcal{F}_k(\mathcal{S}_k, \boldsymbol{x}_{k+1}, f_{k+1})$$

  - Reward

$$r_k(\mathcal{S}_k, \boldsymbol{x}_{k+1}, f_{k+1}) = \max\left\{0, f_{min}^{\mathcal{S}_k} - f_{k+1}\right\}$$

Formulate BO with Finite Budget as DP

- BO as DP instance:
  - Expected reward

$$\forall \boldsymbol{x}_{k+1} \in \mathcal{X},\ U_k(\boldsymbol{x}_{k+1}; \mathcal{S}_k) = \boxed{\mathbb{E}[r_k(\mathcal{S}_k, \boldsymbol{x}_{k+1}, f_{k+1})} + J_{k+1}(\mathcal{F}_k(\mathcal{S}_k, \boldsymbol{x}_{k+1}, f_{k+1}))]$$

$$EI(\boldsymbol{x}; \mathcal{S}_k) = \left(f_{min}^{\mathcal{S}_k} - \overline{\mu}_k(\boldsymbol{x})\right) \Phi\left(\frac{f_{min}^{\mathcal{S}_k} - \overline{\mu}_k(\boldsymbol{x})}{\overline{\sigma}_k(\boldsymbol{x})}\right) + \overline{\sigma}_k(\boldsymbol{x})\phi\left(\frac{f_{min}^{\mathcal{S}_k} - \overline{\mu}_k(\boldsymbol{x})}{\overline{\sigma}_k(\boldsymbol{x})}\right)$$

- Key idea(ADP): Approximate the J_{k+1} by simulation

Rollout for BO

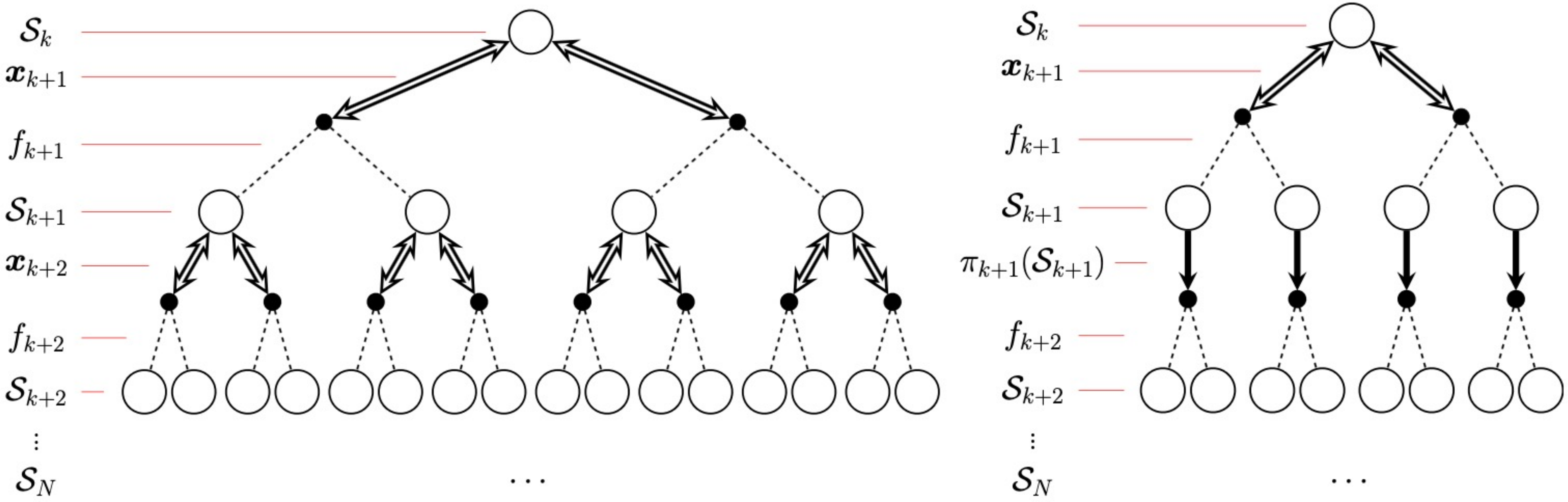
- Rollout relace the requirement to optimally select a design
- Use a suboptimal heuristic to decide which control to apply
- Approximate J_{k+1} by H_{k+1}
- Base policy as heuristic $\boldsymbol{\pi} = (\pi_1, \cdots, \pi_N)$

$$
\begin{aligned}
&H_N(\mathcal{S}_N) = EI(\pi_N(\mathcal{S}_N); \mathcal{S}_N), \\
&H_n(\mathcal{S}_n) = \mathbb{E}\left[r_n(\mathcal{S}_n, \pi_n(\mathcal{S}_n), f_{n+1}) + \gamma H_{n+1}(\mathcal{F}(\mathcal{S}_n, \pi_n(\mathcal{S}_n), f_{n+1}))\right]
\end{aligned}
$$

- Nested expectations are replaced by base policy(Forward manner)

Rollout for BO

- Nested expectations are replaced by base policy(Forward manner)



Rollout for BO

- Approximations
  - Fix rollout horizon to h then the searching interval to $\tilde{N} = \min\{k + h, N\}$
  - Gauss-Hermite quadrature

$$\tilde{H}_{\tilde{N}}(\mathcal{S}_{\tilde{N}}) = EI(\pi_{\tilde{N}}(\mathcal{S}_{\tilde{N}}); \mathcal{S}_{\tilde{N}}),$$

$$\widetilde{H}_n(\mathcal{S}_n) = \sum_{q=1}^{N_q} \alpha^{(q)} \left[r_n \left(\mathcal{S}_n, \pi_n(\mathcal{S}_n), f_{n+1}^{(q)} \right) + \gamma \widetilde{H}_{n+1} \left(\mathcal{F} \left(\mathcal{S}_n, \pi_n(\mathcal{S}_n), f_{n+1}^{(q)} \right) \right) \right]$$

- Utility

$$U_k(\boldsymbol{x}_{k+1}; \mathcal{S}_k) = \sum_{q=1}^{N_q} \alpha^{(q)} \left[r_k \left(\mathcal{S}_k, \boldsymbol{x}_{k+1}, f_{k+1}^{(q)} \right) + \gamma \widetilde{H}_{k+1} \left(\mathcal{F} \left(\mathcal{S}_k, \boldsymbol{x}_{k+1}, f_{k+1}^{(q)} \right) \right) \right]$$

$$U_N(\boldsymbol{x}_{N+1}; \mathcal{S}_N) = EI(\boldsymbol{x}_{N+1}; \mathcal{S}_N)$$

Rollout for BO

- Design of the Base policy:
  - With limited horizon $\boldsymbol{\pi} = \{\pi_{k+1}, \cdots, \pi_{\tilde{N}}\}$
  - At each $n \in \{k+1, \tilde{N}-1\}$ $\pi_n : \mathcal{Z}_n \mapsto \mathcal{X}$, maps a state $z_n = \mathcal{S}_n$

$$\boldsymbol{x}_{n+1} = \underset{\boldsymbol{x} \in \mathcal{X}}{\operatorname{argmax}}\, EI(\boldsymbol{x}; \mathcal{S}_n)$$

$$\boldsymbol{x}_{\tilde{N}+1} = \underset{\boldsymbol{x} \in \mathcal{X}}{\operatorname{argmin}}\, \overline{\mu}_{\tilde{N}}(\boldsymbol{x})$$

- Complexity:
  - Each evaluation of the utility involves $\mathcal{O}\left(N_q^h\right)$ of a heuristic
  - At each heuristic involves optimizing $\mathcal{O}\left(|\mathcal{S}_k|^2\right)$ of work

Experiment

$$G = \frac{f_{min}^{S_1} - f_{min}^{S_{N+1}}}{f_{min}^{S_1} - f(\boldsymbol{x}^*)}$$

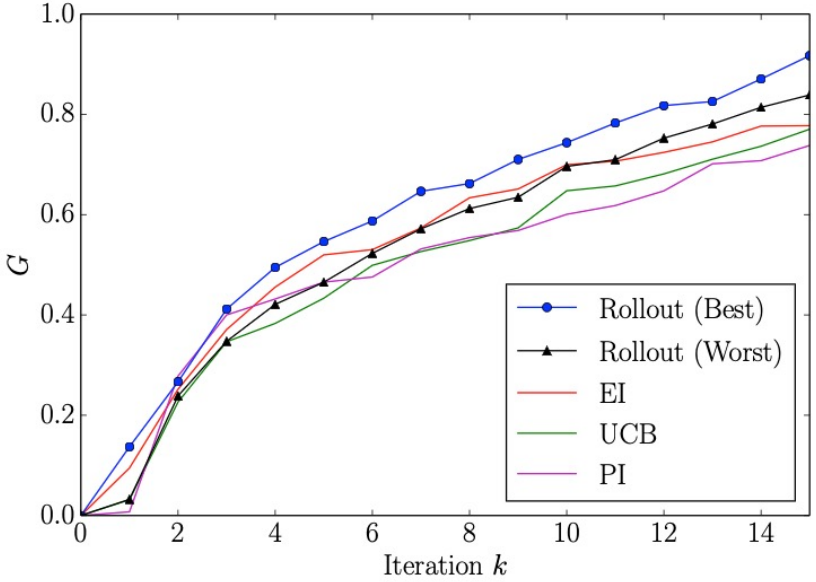
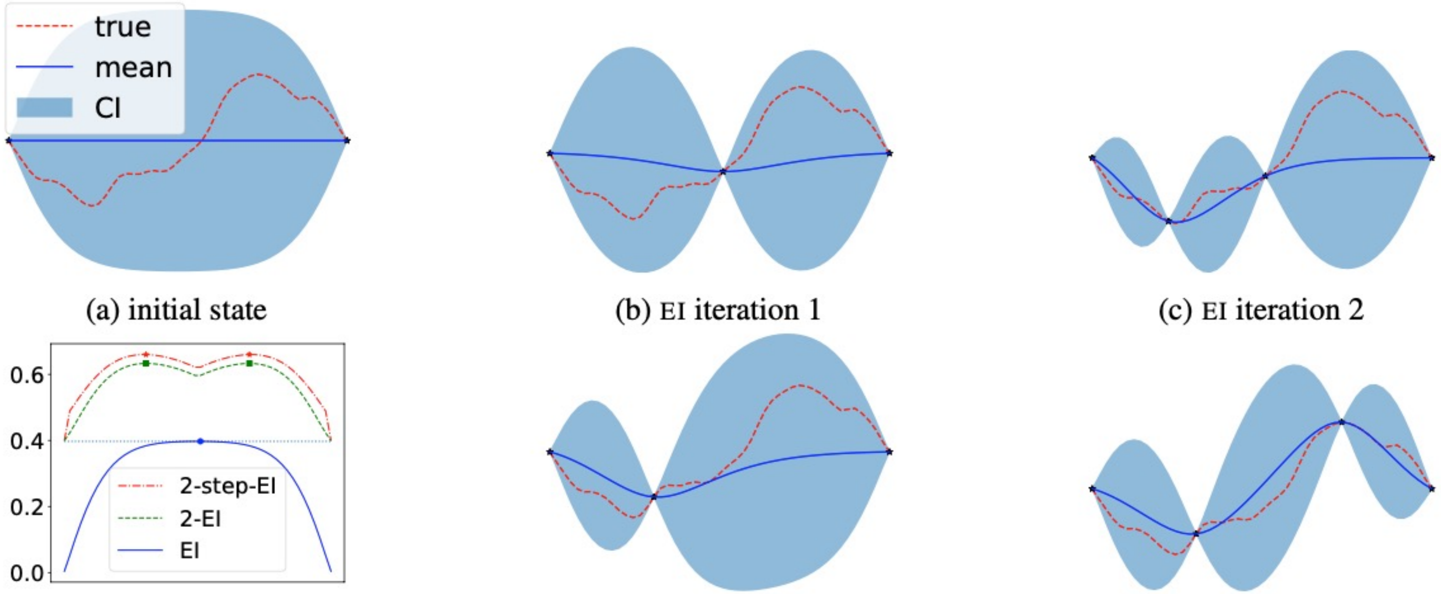


Table 2: Mean and median gap G over 40 initial guesses.

Function name		PI	EI	UCB	PES	GLASSES	R-4-9	R-4-10	R-5-9	R-5-10
Branin-Hoo	Mean	0.847	*0.818*	0.848	0.861	0.846	**0.904**	0.898	0.887	0.903
	Median	0.922	*0.909*	0.910	**0.983**	*0.909*	0.959	0.943	0.921	0.950
Goldstein-Price	Mean	0.873	0.866	*0.733*	0.819	0.782	**0.895**	0.784	0.861	0.743
	Median	0.983	0.981	*0.899*	0.987	0.919	**0.991**	0.985	0.989	0.928
Griewank	Mean	*0.827*	0.884	0.913	**0.972**	1[2]	0.882	0.885	0.930	0.867
	Median	*0.904*	0.953	0.970	**0.987**	1[2]	0.967	0.962	0.960	0.954
Six-hump Camel	Mean	0.850	**0.887**	0.817	*0.664*	0.776	0.860	0.825	0.793	0.803
	Median	0.893	**0.970**	0.915	*0.801*	0.941	0.926	0.900	0.941	0.907

BINOCULARS

- BINOCULARS(**B**atch-**I**nformed **NO**nmyopic **C**hoices **U**sing **L**ong-horizons for **A**daptive, **R**apid **S**ED)



(a) initial state (b) EI iteration 1 (c) EI iteration 2
(d) EI, 2-EI and 2-step-EI (e) 2-EI iteration 1 (f) 2-EI iteration 2

BINOCULARS

- Contribution
  - Formulate the adaptive policy from the batch optimization
  - Optimizing any single element of the batch is equivalent to optimize the lower bound of the adaptive policy
- Cons:
  - Eliminate the procedures how to conduct the batch optimization

BINOCULARS

- Notations
  - Design space $\mathcal{X}$ response space $\mathcal{Y}$ dataset $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$ probabilistic model $p(y \mid x, \mathcal{D})$
  - Marginal gain utility $u(y \mid x, \mathcal{D}) = u\big(\mathcal{D} \cup (x, y)\big) - u(\mathcal{D})$
  - Expected utility after observing dataset and k steps remaining $Q_k(x \mid \mathcal{D})$

$$
\begin{aligned}
Q_k(x \mid \mathcal{D}) = \mathbb{E}_y[u(y \mid x, \mathcal{D})] + \\
\mathbb{E}_y\left[\max_{x'} Q_{k-1}\big(x' \mid \mathcal{D} \cup \{(x, y)\}\big)\right]
\end{aligned}
$$

  - Optimal policy at i-th step with horizon T

$$
x^* = \operatorname*{argmax}_x Q_{T-i}(x \mid \mathcal{D}_i)
$$

BINOCULARS

- A batch view of adaptive policy design
  - Suppose **simultaneously** design T experiments $X = \{x_1, \ldots, x_T\}$ given $\mathcal{D}$
  - The expected marginal utility over the joint of $Y = \{y_1, \ldots, y_T\}$, $p(Y \mid X, \mathcal{D})$

$$Q(X \mid \mathcal{D}) = \mathbb{E}_Y[u(Y \mid X, \mathcal{D})]$$

  - Decomposing $X_{-j} = X \setminus \{x_j\}$

$$Q(X \mid \mathcal{D}) = \mathbb{E}_{y_j}[u(y_j \mid x_j, \mathcal{D})] + \\ \mathbb{E}_{y_j}\Big[Q\big(X_{-j} \mid \mathcal{D} \cup \{(x_j, y_j)\}\big)\Big]$$

BINOCULARS

- Batch View
  - Let $X^* \in \arg\max_X Q(X \mid \mathcal{D})$
  - For any $x_j^* \in X^*$

$$\mathbb{E}_{y_j^*}\left[Q\left(X_{-j}^* \mid \mathcal{D} \cup \{(x_j^*, y_j^*)\}\right)\right] = \max_{X_{-j}} \mathbb{E}_{y_j^*}\left[Q\left(X_{-j} \mid \mathcal{D} \cup \{(x_j^*, y_j^*)\}\right)\right]$$

  - Choosing any $x^* \in X^*$ is equivalent to solve $x^* \in \arg\max_x B(x \mid \mathcal{D})$

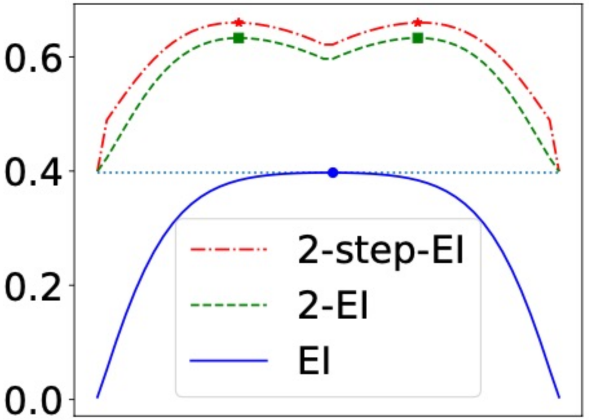
$$B(x \mid \mathcal{D}) = \mathbb{E}_y[u(y \mid x, \mathcal{D})] + \max_{X' : |X'| = T-1} \mathbb{E}_y\left[Q\left(X' \mid \mathcal{D} \cup \{(x, y)\}\right)\right]$$

$$\boxed{Q_k(x \mid \mathcal{D}) = \mathbb{E}_y[u(y \mid x, \mathcal{D})] + \mathbb{E}_y\left[\max_{x'} Q_{k-1}\left(x' \mid \mathcal{D} \cup \{(x, y)\}\right)\right]}$$

BINOCULARS

- Batch View
  - Lower bound of the true expected utility

$$
\begin{aligned}
&\max_{X':|X'|=T-1} \mathbb{E}_y\Big[Q\big(X' \mid \mathcal{D}\cup\{(x,y)\}\big)\Big] \\
&\leq \mathbb{E}_y\left[\max_{X':|X'|=T-1} Q\big(X' \mid \mathcal{D}\cup\{(x,y)\}\big)\right] \\
&\leq \mathbb{E}_y\left[\max_{x'} Q_{T-1}\big(x' \mid \mathcal{D}\cup\{(x,y)\}\big)\right].
\end{aligned}
$$



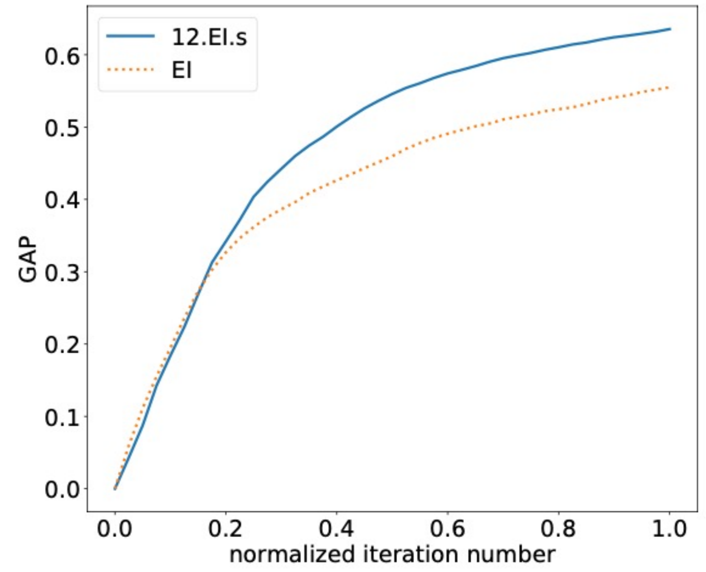
Algorithm 1 BINOCULARS

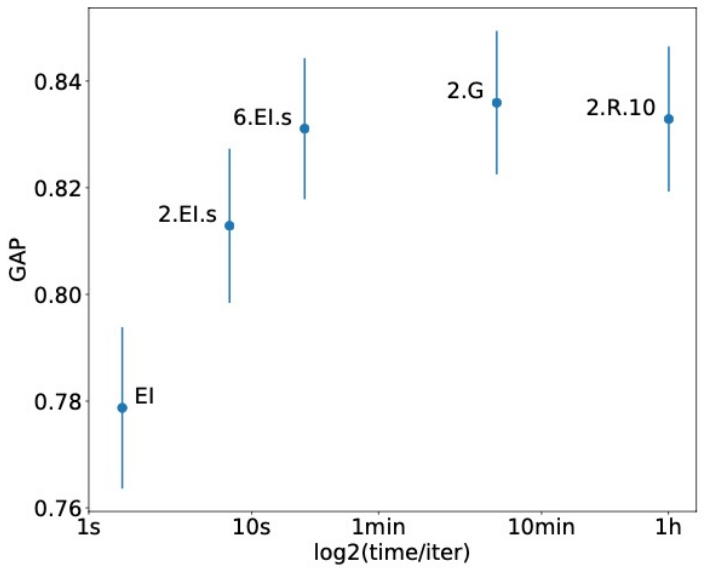
Input: design space $\mathcal{X}$, response space $\mathcal{Y}$, model $p(y \mid x, \mathcal{D})$, utility function $u(y \mid x, \mathcal{D})$, budget T
Output: $\mathcal{D}$, a sequence of experiments and observations
for $i \leftarrow 0$ to $T-1$ **do**
  Compute the optimal batch X^* of size $T-i$
  Pick an experiment $x^* \in X^*$ and observe response y^*
  Augment $\mathcal{D} = \mathcal{D} \cup \{(x^*, y^*)\}$

BINOCULARS

- Experiment

	EI	2.EI.s	3.EI.s	4.EI.s	6.EI.s	8.EI.s	2.G	3.G	2.R.10	3.R.3
SVM	0.738	*0.913*	**0.940**	*0.911*	*0.937*	0.834	*0.881*	0.898	*0.930*	*0.928*
LDA	0.956	**1.000**	*0.996*	*0.993*	0.982	*0.995*	*1.000*	*0.999*	*0.999*	*1.000*
LogReg	0.963	*0.998*	*1.000*	*0.999*	*0.999*	**1.000**	0.989	0.911	0.965	0.948
NN Boston	*0.470*	*0.467*	*0.478*	*0.460*	*0.502*	*0.467*	0.455	**0.512**	*0.503*	*0.482*
NN Cancer	0.665	0.627	0.654	0.686	0.700	0.686	**0.806**	*0.755*	0.708	0.698
Robot pushing 3d	0.928	*0.960*	*0.962*	*0.957*	**0.962**	*0.961*	*0.955*	0.951	*0.955*	*0.954*
Robot pushing 4d	*0.730*	0.726	0.695	0.695	0.736	0.697	*0.765*	**0.786**	*0.770*	*0.745*
Average	0.779	*0.813*	*0.818*	0.815	*0.831*	0.806	**0.836**	*0.830*	*0.833*	*0.822*





Efficient Nonmyopic Bayesian Optimization

- Contribution
  - One-shot multiple-step trees: **jointly** optimize all decision variables in one-shot fashion
  - Fast-fantasies and parallelism: **LOVE** cache with **GPytorch**
  - Improved performance over myopic EI and BINOCULARS

Efficient Nonmyopic Bayesian Optimization

- Bayesian Optimal Policy
  - OP: $x^* \in \arg\max_{x \in \mathcal{X}} f(x)$
  - Utility of decision horizon k $u(\mathcal{D}_k) = \max_{(x,y) \in \mathcal{D}_k} y$
  - Define: $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$ recursively $\mathcal{D}_i = \mathcal{D}_{i-1} \cup \{(x_i, y_i)\}$
  - Policy: a collection of decision functions $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_k)$
  - Objective: $\sup_{\boldsymbol{\pi}} \mathbb{E}[u(\mathcal{D}_k^{\boldsymbol{\pi}})]$
  - One step marginal:

$$v_1(x \mid \mathcal{D}) = \mathbb{E}_y \left[u(\mathcal{D} \cup \{(x, y)\}) - u(\mathcal{D}) \mid x, \mathcal{D} \right]$$

ENO

- Bayesian Optimal Policy
  - k-steps marginal by Bellman's recursion:

$$v_t(x \mid \mathcal{D}) = v_1(x \mid \mathcal{D}) + \mathbb{E}_y[\max_{x'} v_{t-1}(x' \mid \mathcal{D} \cup \{(x, y)\})]$$

  - Batch marginal per iteration(q-EI)

$$V_1^q(X \mid \mathcal{D}) = \mathbb{E}_{y^{(1)},\ldots,y^{(q)}}\left[u(\mathcal{D} \cup \{(x^{(1)}, y^{(1)}), \ldots, (x^{(q)}, y^{(q)})\}) - u(\mathcal{D}) \mid X, \mathcal{D}\right]$$

Relaxation

Nested Expectation-Maximization Problem

- Expand the k-steps problem

$$v_k(x \,|\, \mathcal{D}) = v_1(x \,|\, \mathcal{D}) + \mathbb{E}_y\Big[\max_{x_2}\Big\{v_1(x_2 \,|\, \mathcal{D}_1) + \mathbb{E}_{y_2}\big[\max_{x_3}\big\{v_1(x_3 \,|\, \mathcal{D}_2) + \cdots\Big]$$

- Replace the expectation by *fantasy* samples from model's posterior

$$\bar{v}_k(x \,|\, \mathcal{D}) = v_1(x \,|\, \mathcal{D}) + \frac{1}{m_1}\sum_{j_1=1}^{m_1}\Big[\max_{x_2}\Big\{v_1(x_2 \,|\, \mathcal{D}_1^{j_1}) + \frac{1}{m_2}\sum_{j_2=1}^{m_2}\big[\max_{x_3}\big\{v_1(x_3 \,|\, \mathcal{D}_2^{j_1 j_2}) + \cdots\Big]$$

$$\mathcal{D}_1^{j_1} = \mathcal{D} \cup \{(x, y^{j_1})\}$$

$$\mathcal{D}_t^{j_1 \dots j_t} = \mathcal{D}_{t-1}^{j_1 \dots j_{t-1}} \cup \{(x_t^{j_1 \dots j_{t-1}}, y_t^{j_1 \dots j_t})\}$$

$$y_t^{j_1 \dots j_t} \sim p(y_t \,|\, x_t^{j_1 \dots j_t}, \mathcal{D}_{t-1}^{j_1 \dots j_{t-1}})$$

Reparameterization Trick

- Reparametrize $\bar{v}_k(x \mid \mathcal{D})$

$$y = h_{\mathcal{D}}(x, z)$$

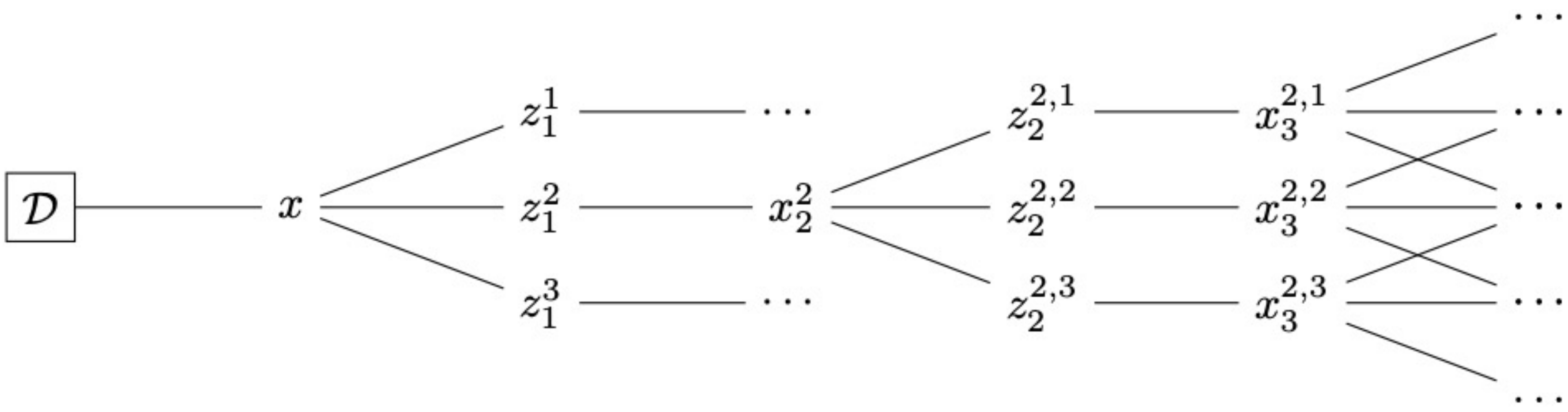
z is a random variable independent of both x and $\mathcal{D}$ $\quad z \sim \mathcal{N}(0, I)$

$$h_{\mathcal{D}}(x, z) = \mu_{\mathcal{D}}(x) + L_{\mathcal{D}}(x) z$$

$$L_{\mathcal{D}}(x) L_{\mathcal{D}}^T(x) = \Sigma_{\mathcal{D}}(x)$$

Base samples

Scenario Tree



$$\bar{v}_k(x \mid \mathcal{D}) = v_1(x \mid \mathcal{D}) + \frac{1}{m_1}\sum_{j_1=1}^{m_1}\Big[\max_{x_2}\Big\{v_1(x_2 \mid \mathcal{D}_1^{j_1}) + \frac{1}{m_2}\sum_{j_2=1}^{m_2}\big[\max_{x_3}\big\{v_1(x_3 \mid \mathcal{D}_2^{j_1 j_2}) + \cdots\Big]$$

Differentiable Scenario Tree

- Jointly optimize all decision variables is equivalent to optimize the decision variable of the k-steps problem

Proposition 1. *Fix a set of base samples and consider* $\bar{v}_k(x \mid \mathcal{D})$. *Let* $x_t^{j_1 \cdots j_{t-1}}$ *be an instance of* x_t *for each realization of* $\mathcal{D}_{t-1}^{j_1 \cdots j_{t-1}}$ *and let*

$$x^*, \mathbf{x}_2^*, \mathbf{x}_3^*, \ldots, \mathbf{x}_k^* = \operatorname*{arg\,max}_{x, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_k} \Bigg\{ v_1(x \mid \mathcal{D}) + \frac{1}{m_1} \sum_{j_1=1}^{m_1} v_1(x_2^{j_1} \mid \mathcal{D}_1^{j_1}) + \cdots + \frac{1}{\prod_{\ell=1}^{k-1} m_\ell} \sum_{j_1=1}^{m_1} \cdots \sum_{j_{k-1}=1}^{m_{k-1}} v_1(x_k^{j_1 \cdots j_{k-1}} \mid \mathcal{D}_{k-1}^{j_1 \cdots j_{k-1}}) \Bigg\}, \quad (6)$$

Differentiable

where we compactly represent $\mathbf{x}_2 = \{x_2^{j_1}\}_{j_1=1 \ldots m_1}$, $\mathbf{x}_3 = \{x_3^{j_1 j_2}\}_{j_1=1 \ldots m_1, j_2=1 \ldots m_2}$, *and so on. Then,* $x^* = \arg\max_x \bar{v}_k(x \mid \mathcal{D})$.

One-step multi-step approach

Fast-Fantasies

- Lanczos Variance Estimates

Constant-Time Predictive Distributions for Gaussian Processes

Geoff Pleiss [1] Jacob R. Gardner [1] Kilian Q. Weinberger [1] Andrew Gordon Wilson [1]

Table 1. Asymptotic complexities of predictive (co)variances (n training points, m inducing points, k Lanczos/CG iterations) and sampling from the predictive distribution (s samples, t test points).

Method	**Pre-computation** (time)	(storage)	**Computing variances** (time)	**Drawing s samples** (time)
Standard GP	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	$\mathcal{O}(tn^2 + t^2(n+s) + t^3)$
SGPR	$\mathcal{O}(nm^2)$	$\mathcal{O}(m^2)$	$\mathcal{O}(m^2)$	$\mathcal{O}(tm^2 + t^2(m+s) + t^3)$
KISS-GP	–	–	$\mathcal{O}(k(n+m\log m))$	$\mathcal{O}(kt(n+m\log m)+t^2(m+s)+t^3)$
KISS-GP (w/ LOVE)	$\mathcal{O}(k(n+m\log m))$	$\mathcal{O}(km)$	$\mathcal{O}(k)$	$\mathcal{O}(ks(t+m))$

Fast-Fantasies

- LOVE Cache

$$k_{f|\mathcal{D}}(x_i^*, x_j^*) = k_{x_i^* x_j^*} - \mathbf{k}_{Xx_i^*}^\top (K_{XX} + \Sigma)^{-1} \mathbf{k}_{Xx_j^*}$$

$$\tilde{K}_{XX} = RR^\top, \quad \tilde{K}_{XX} := K_{XX} + \Sigma \in \mathbb{R}^{n \times n}$$

QR decompose R

LOVE Cache

Rank-1 Cholesky update

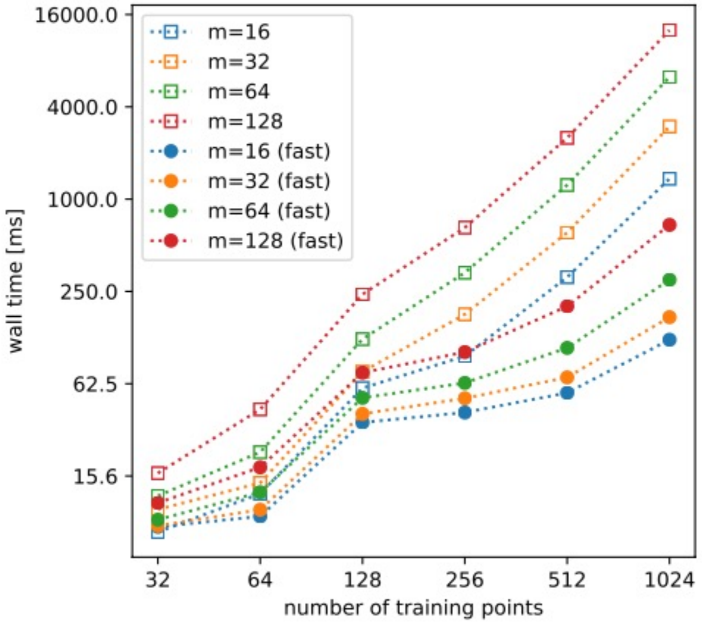
$$k_{f|\mathcal{D}}(x_i^*, x_j^*) = k_{x_i^* x_j^*} - \mathbf{k}_{Xx_i^*}^\top R^{-\top} R^{-1} \mathbf{k}_{Xx_j^*}$$

Pseudo-inverse

Fast-Fantasies

- LOVE Cache

Proposition 2. *Suppose* $(K_{XX} + \Sigma)^{-1}$ *has been decomposed using LOVE into* $R^{-\top} R^{-1}$, *with* $R^{-1} \in \mathbb{R}^{n \times r}$. *Suppose we wish to augment* X *with* q *data points, thereby augmenting* K_{XX} *with* q *rows and columns, yielding* $K_{\hat{X}\hat{X}}$. *A rank* $r + q$ *decomposition* $\hat{R}^{-1}$ *of the inverse,* $\hat{R}^{-\top} \hat{R}^{-1} \approx (K_{\hat{X}\hat{X}} + \Sigma)^{-1}$, *can be computed from* R *in* $\mathcal{O}(nrq)$ *time.*



Special Instances (linear to k)

- Multi-Step (deterministic) Path
  - Single path when $m_t = 1 \text{ for } t \geq 2$
  - Equivalent to single point quadrature (mean of the Gaussian)
  - Certainty equivalent conol
  - Works surprisingly well in practice
- Non-Adaptive Approximation (ENO)
  - replace the adaptive value function v_{k-1} by V_1^{k-1}

$$\max_{x, X^{(1)}, \ldots, X^{(m_1)}} v_1(x \mid \mathcal{D}) + \frac{1}{m_1} \sum_{i=1}^{m_1} V_1^{k-1}(X^{(i)} \mid \mathcal{D}_1^{(i)})$$

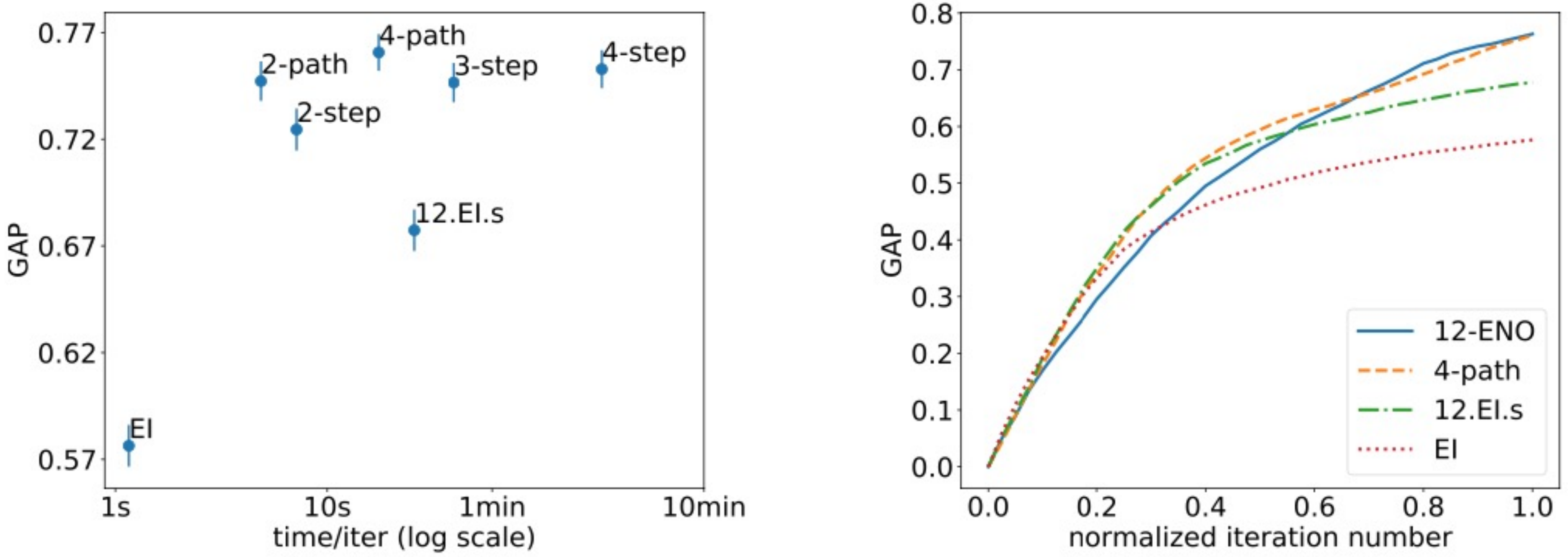
Tightness of Lowerbound

Method	Acquisition Function
multi-step (ours)	$v_1(x \mid \mathcal{D}) + \mathbb{E}_y[\max_{x'} v_{k-1}(x' \mid \mathcal{D}_1)]$
ENO (ours)	$v_1(x \mid \mathcal{D}) + \mathbb{E}_y[\max_X V_1^{k-1}(X \mid \mathcal{D}_1)]$
BINOCULARS [15]	$v_1(x \mid \mathcal{D}) + \max_X \mathbb{E}_y[V_1^{k-1}(X \mid \mathcal{D}_1)]$
GLASSES [11]	$v_1(x \mid \mathcal{D}) + \mathbb{E}_y[V_1^{k-1}(X_g \mid \mathcal{D}_1)]$
rollout [18]	$r_k(x \mid \mathcal{D}) = r_1(x \mid \mathcal{D}) + \mathbb{E}_y[r_{k-1}(\pi(\mathcal{D}_1) \mid \mathcal{D}_1)]$
two-step [32]	$v_1(x \mid \mathcal{D}) + \mathbb{E}_y[\max_{x'} v_1(x' \mid \mathcal{D}_1)]$
one-step [21]	$v_1(x \mid \mathcal{D}) + 0$
relationships (when $k \geq 2$)	multi-step $\geq$ ENO $\geq$ BINOCULARS $\geq$ GLASSES $\geq$ one-step; multi-step $\geq$ rollout $\geq$ two-step $\geq$ one-step; ENO $\geq$ two-step.

Experiment

	EI	ETS	12.EI.s	2-step	3-step	4-step	4-path	12-ENO
eggholder	0.627	0.647	**0.736**	0.478	0.536	0.577	0.567	0.661
dropwave	0.429	0.585	0.606	0.545	0.600	0.635	**0.731**	0.673
shubert	0.376	*0.487*	*0.515*	0.476	*0.507*	**0.562**	*0.560*	*0.494*
rastrigin4	0.816	0.495	0.790	**0.851**	*0.821*	*0.826*	*0.837*	*0.837*
ackley2	0.808	0.856	*0.902*	*0.870*	*0.895*	*0.888*	**0.931**	0.847
ackley5	0.576	0.516	0.703	0.786	0.793	0.804	**0.875**	*0.856*
bukin	0.841	0.843	0.842	**0.862**	*0.862*	*0.861*	*0.852*	0.836
shekel5	0.349	0.132	0.496	*0.827*	**0.856**	*0.847*	0.718	0.799
shekel7	0.363	0.159	0.506	*0.825*	*0.850*	0.775	0.776	**0.866**
Average	0.576	0.524	0.677	0.725	*0.747*	*0.753*	*0.761*	**0.763**
Ave. time	1.157	1949.	25.74	7.163	39.53	197.7	17.50	15.61

Experiment



(a) (b)