

Choosing a Suitable Acquisition Function for Batch Bayesian Optimization: Comparison of Serial and Monte Carlo Approaches

Imon Mia,^a Mark Lee,^b Weijie Xu,^a William Vandenberghe,^a and Julia W. P. Hsu^{* a}

^a*Department of Materials Science and Engineering, The University of Texas at Dallas, USA.*

^b*Department of Physics, The University of Texas at Dallas, USA.*

^{*}*Email: jwhsu@utdallas.edu*

Note 1: Empirical Perovskite Power Conversion Efficiency Model Background

The empirical power conversion efficiency (PCE) model is based on our previously reported Bayesian optimization (BO) experiment to maximize the PCE of flexible perovskite solar cells.¹ These perovskite solar cells were made using a novel annealing process called photonic curing. Photonic curing uses a flash xenon lamp to deliver intense photon pulses (20 μ s–100 ms) with a broadband spectrum (200–1500 nm).² The perovskite precursor films absorb radiant energy from the light pulses and become crystallized.

To use BO to maximize the PCE, we selected as input features the four most critical synthesis parameters that determine the perovskite solar cell characteristics: (1) perovskite precursor concentration (in M), (2) additive diiodomethane volume (in μ L), (3) pulse voltage (in V), which determines the light intensity, and (4) pulse length (in ms). The ranges for each of these features are given in Table S1. In the BO posterior model computation, each feature is normalized to the interval [0, 1], with 0 corresponding to the minimum and 1 corresponding to the maximum value of each feature's range. The normalized feature values are labeled x_1, x_2, x_3, x_4 , respectively, so the input predictors are 4-D vectors $\mathbf{X} = (x_1, x_2, x_3, x_4)$, whose domain is the hypercube [0, 1]⁴. The output objective is the predicted solar cell PCE.

Experimental data from 48 solar cells were used to train a Random Forest ensemble model with four base learners: Gradient Boosting, Gaussian Process, Random Forest, and Neural Network, which gave us best predicting capability among other base learners. This ensemble model serves as the ground truth function for the acquisition functions tested in the main text. To evaluate the PCE model at the new predictor points recommended by the acquisition functions, the model is gridded into a look-up table with the (unnormalized) intervals below. These intervals correspond to the step size used to vary each feature in the real experiment.

Table S1 Four-dimensional input parameters space

Input Parameters (Unit)	Dimension	Range (Interval)*
Perovskite Precursor Concentration (M)	x_1	1.2 - 1.6 M (0.05 M)
Additive Diiodomethane Volume (μ L)	x_2	0 - 250 μ L (5 μ L)
Pulse Voltage (V)	x_3	200 - 440 V (5 V)
Pulse Length (ms)	x_4	1 - 100 ms (1 ms)

*Before normalization

References

- 1 W. Xu, Z. Liu, R. T. Piper and J. W. P. Hsu, Bayesian Optimization of photonic curing process for flexible perovskite photovoltaic devices, *Sol. Energy Mater. Sol. Cells*, 2023, **249**, 112055.
- 2 K. A. Schroder, Mechanisms of photonic curingTM: Processing high temperature films on low temperature substrates, *Tech. Proc. 2011 NSTI Nanotechnol. Conf. Expo, NSTI-Nanotech 2011*, 2011, **2**, 220–223.

Note 2: Computational Time and Memory Usage

All code was implemented in Python and run in normal mode (CPU only) on the Lonestar6 system of the Texas Advanced Computing Center. The Emukit package was used for UCB/LP and logEI/LP, and the BoTorch package was used for all Monte Carlo parallel batch acquisition functions.

Each 50-iteration Bayesian optimization campaign used 2 GB of random-access memory.

Problem	Acquisition Function	Mean “wall clock” time for one 50-iteration Bayesian optimization campaign (sec)
Ackley 6D (needle-in-haystack)	UCB/LP	105
	q UCB	80
	q logEI	105
	q logEI+Turbo	100
Hartmann 6D no noise (false maximum)	UCB/LP	90
	q UCB	85
	q logEI	110
Hartmann 6D with noise (false maximum)	UCB/LP	90
	q UCB	75
	q logEI	100
	q logNEI	110
PCE Model 4D (flat landscape, real noise/errors)	UCB/LP	65
	q UCB	70
	q logNEI	85

Note 3: Rationale for Main Bayesian Optimization Setting Choices

Setting Choice	Reason
ARD Matern 5/2 kernel for Gaussian Process regression	The ARD Matern 5/2 function is a standard GP regression kernel widely used when modeling real data. ARD (Automatic Relevance Detection) allows for different characteristic length scales in each input dimension, ¹ which is always the case for real experimental inputs. Because the Matern 5/2 function is discontinuous in its second derivative, it is better able to model possibly non-smooth functions that often occur in real-world problems when compared to the radial basis function (Gaussian) kernel more favored in theoretical studies.
Kernel hyperparameter tuning by maximizing likelihood	Maximizing statistical likelihood (or log likelihood) is a standard and widely used method to tune the Gaussian Process regression kernel’s hyperparameters. This is the default hyperparameter tuning method used in most GP software packages in Python (including BoTorch and Emukit used for this paper) and Matlab.
$\beta = 2$ for the exploration-exploitation parameter value for UCB/LP and q UCB	Integer values of β between 1 and 7 were tried with only minor differences that do not affect the conclusions presented in this paper. This is consistent with Diessner, <i>et al.</i> , ² who found performance of UCB-type acquisition functions to be insensitive to β values between 1 and 5. We settled on $\beta = 2$ because it gave somewhat better, but representative, overall results. Since this work is not meant as a study of the effects of β , it was decided to fix $\beta = 2$ for all UCB-related acquisition functions.
Acquisition function optimization methods	Used default acquisition function optimization methods in Python packages Emukit and BoTorch.
Batch size $q = 4$	Common experimental batch size in new materials synthesis and processing experiments.
TuRBO for domain narrowing on the Ackley function in 6-dimensions	Publicly available Python code implementing TuRBO with BoTorch ³ is available and was suitable for our needs.

¹J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms, *Adv Neural Info Proc Sys* **25**, 2951-2959 (2012).

²M. Diessner, J. O’Connor, A. Wynn, S. Laizet, Y. Guan, K. Wilson and R. D. Whalley, Investigating Bayesian optimization for expensive-to-evaluate black box functions: Application in fluid dynamics, *Front Appl Math Stat* **8**, 1076296 (2022) DOI:10.3389/fams.2022.1076296.

³https://botorch.org/docs/tutorials/turbo_1/

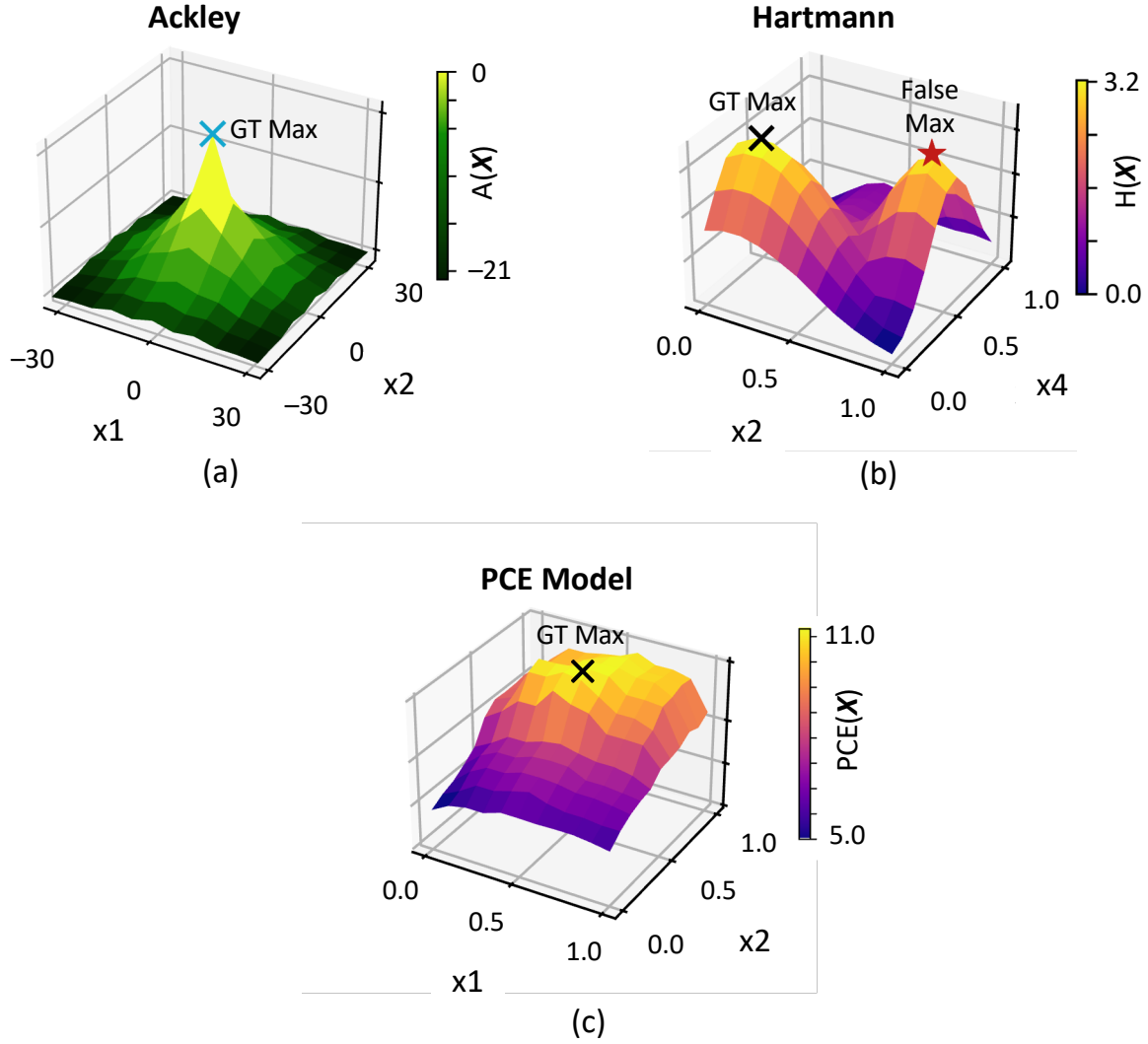


Fig. S1. Surface contour plots of ground truth (a) Ackley function $A(X)$, (b) Hartmann function $H(X)$, and (c) PCE model $PCE(X)$, projected onto one coordinate plane. The input vector $X = (x_1, x_2, x_3, x_4, x_5, x_6)$ for Ackley and Hartmann functions and $X = (x_1, x_2, x_3, x_4)$ for the PCE model. The z-axis values form a surface defined by the maximum value associated with each (x_1, x_2) for $A(X)$ and $PCE(X)$ or each (x_2, x_4) $H(X)$ projected out from all other components of X . In all plots “x” indicates the ground truth maximum (GT Max). In (b) a red star indicates the false maximum.

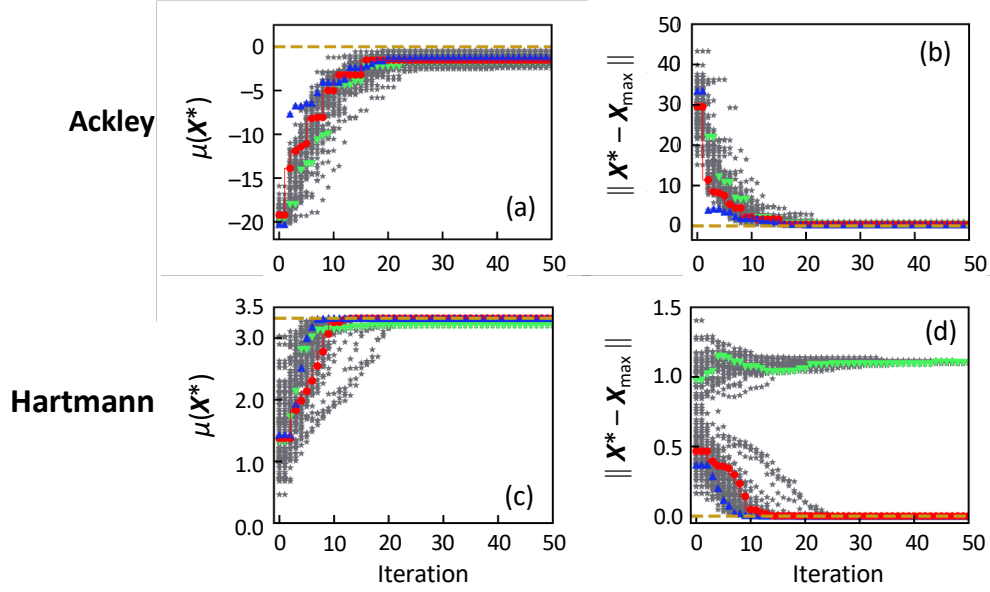


Fig. S2. Learning progression data on the noiseless Ackley (top row) and Hartmann (bottom row) test functions for serial batch acquisition function logEI/LP. $\mu(X^*)$ and $\|X^* - X_{\max}\|$ have the same meanings as in Fig. 2 in the main text. The ground truth y_{\max} and X_{\max} are indicated by yellow dashed lines. Gray points show the spread in learning progress of the 99 campaigns starting from the 99 different initial data sets. Green, red, and blue points indicate the runs ranked in the top 25th, 50th, and 75th percentile, respectively, as defined in the main text.

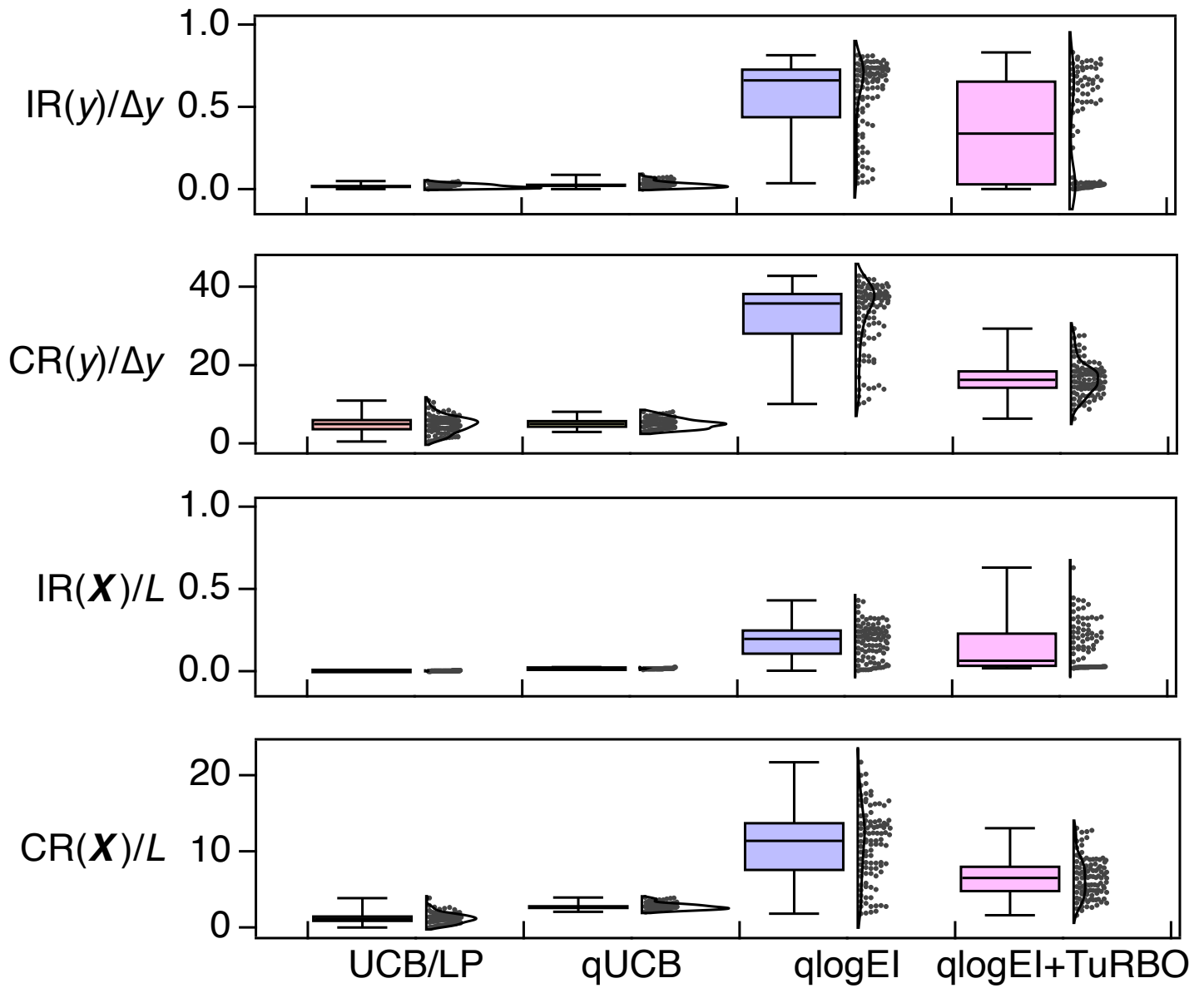


Fig. S3a. Box and violin plots showing the distribution of instantaneous regret (IR) and cumulative regret (CR) data corresponding to Table 1 in the main text for the four batch acquisition functions used on the Ackley test function. Each dot represents the IR or CR result of a 50-iteration batch Bayesian optimization campaign starting from one of the 99 initialization conditions selected by Latin hypercube sampling. Here, y is the objective value, and X is the input of the function being optimized.

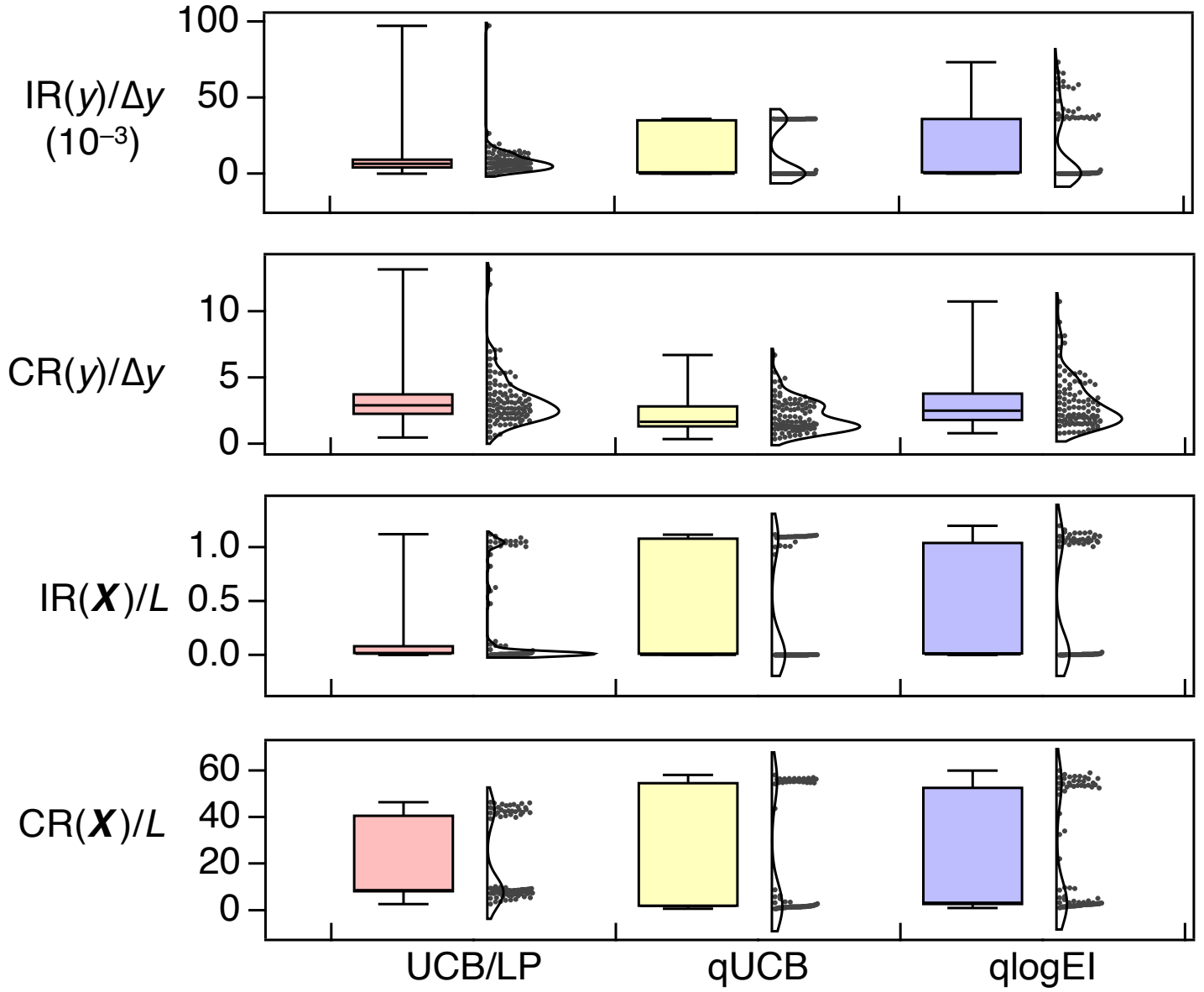


Fig. S3b. Box and violin plots showing the distribution of instantaneous regret (IR) and cumulative regret (CR) data corresponding to Table 2 in the main text for the three batch acquisition functions used on the Hartmann test function without noise. Each dot represents the IR or CR result of a 50-iteration batch Bayesian optimization campaign starting from one of the 99 initialization conditions selected by Latin hypercube sampling. Here, y is the objective value, and \mathbf{X} is the input of the function being optimized.

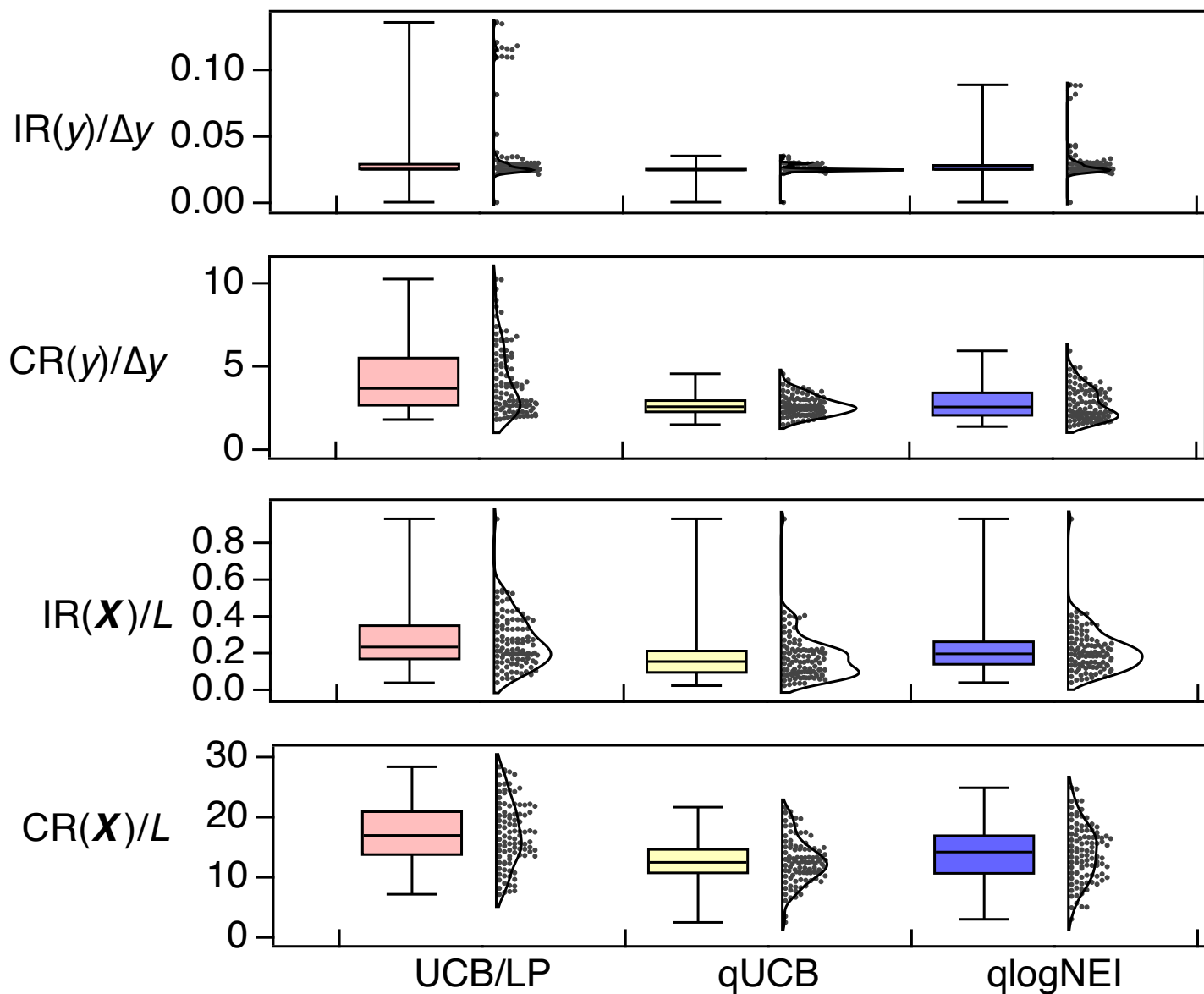


Fig. S3c. Box and violin plots showing the distribution of instantaneous regret (IR) and cumulative regret (CR) data corresponding to Table 3 in the main text for the three batch acquisition functions used on the PCE model test function. Each dot represents the IR or CR result of a 50-iteration batch Bayesian optimization campaign starting from one of the 99 initialization conditions selected by Latin hypercube sampling. Here, y is the objective value, and \mathbf{X} is the input of the function being optimized.

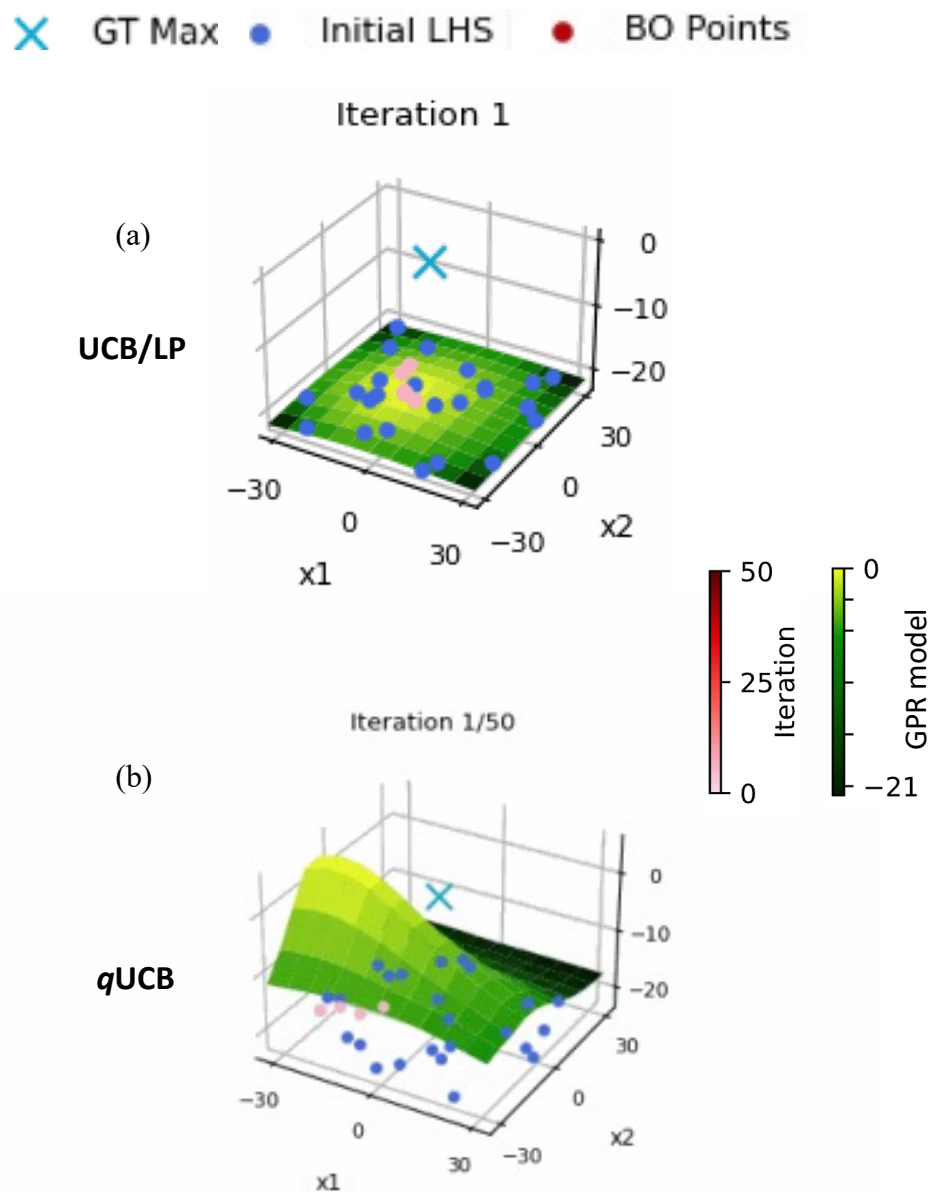


Fig. S4. Time lapse video clips showing the bBO learning progression on the Ackley function using (a) serial UCB/LP and (b) MC q UCB bAFs. The z-axis surfaces are the maximum value the GPR model gives for each (x_1, x_2) projected out from all other components of X , after each iteration. The ground truth maximum (GT Max) is indicated by a blue “X”. Blue circles indicate the initial set of 24 points picked by Latin hypercube sampling. New batch points picked by the bAF are shown circles progressing in color from light pink (early iterations) to dark red (later iterations).