Supplemental Materials: Bayesian Optimization for Policy Search via Online-Offline Experimentation

Online Appendix 1: Factors behind MTGP performance

Section 4 of the main text evaluated cross-validation mean squared error for a single-task GP and an MTGP on 99 separate experiment outcomes. Fig. 4 showed that for many experiment outcomes, use of the simulator and the MTGP significantly improved prediction ability over the single-task GP. Section 6 showed that the key quantity in MTGP generalization is the squared inter-task correlation, ρ^2. Consistent with that, Fig. S1 shows that most outcomes had high values of ρ^2, but the outcomes with high MTGP MSE nearly all have low values of ρ^2.

Fig. S2 directly shows the relationship between ρ^2 and cross-validation MSE for those same 99 experiment outcomes in Fig. S1. There is a clear correlation between the two, and values of ρ^2 greater than 0.75 always produced low MTGP MSE.

There are two potential sources of low inter-task correlation. The first is a high level of observation noise: If the level of observation noise is high relative to the effect size, the correlation will be low regardless of how similar the true response surfaces are. The second source of low inter-task correlation is poor fidelity of the simulator. To help identify the source of the low correlations, Fig. S2 also compares cross validation error to the noise standard deviation. These results tell a mixed story: Many of the outcomes with low values of ρ^2 have high noise levels, suggesting noise as a likely cause, but others do not, suggesting poor simulator fidelity as the likely cause.

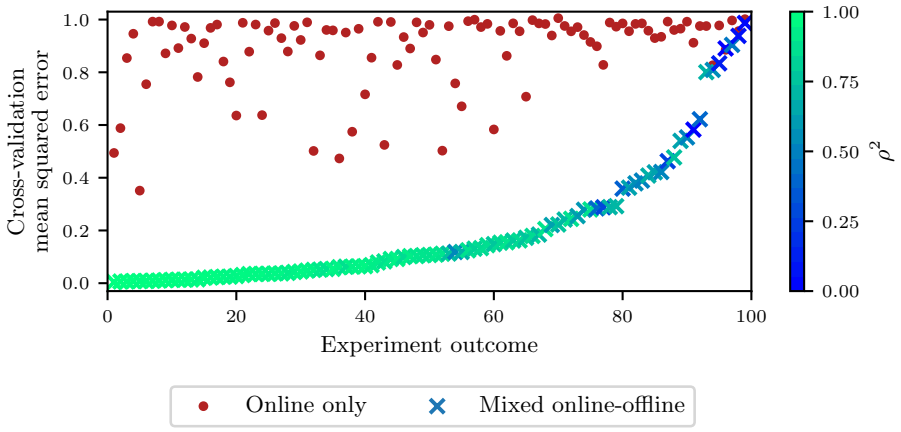

Figure S1: The same results as Fig. 4 from the main text, with the MTGP markers colored according to the squared inter-task correlation, ρ^2. Outcomes with high MTGP error have moderate or low values of ρ^2. Low inter-task correlation is the factor behind high MTGP error.

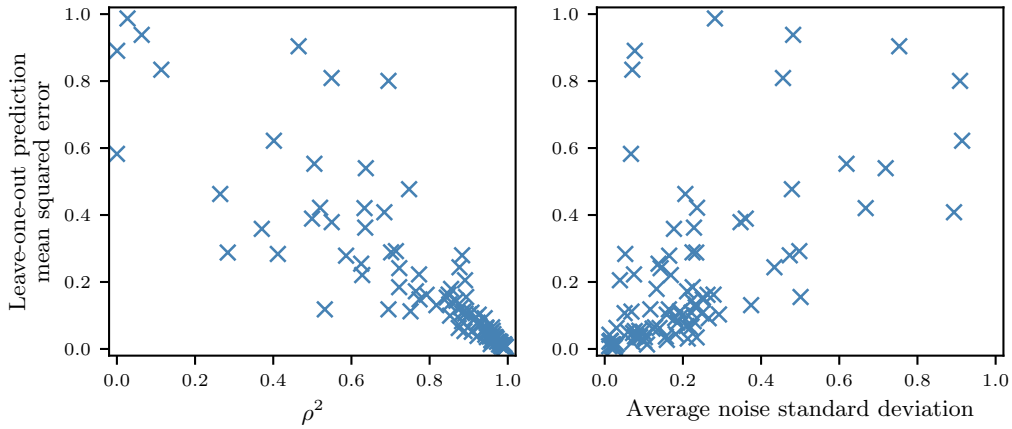

Figure S2: (Left) ρ^2 vs. cross-validation error for the 99 experiment outcomes in Fig. S1. As expected from theory, MTGP performance depends strongly on ρ^2. (Right) Cross-validation error vs. noise level for the same outcomes. Error tends to be higher for cases where the noise standard deviation is high, although there are some instances of high error with low noise level.

Online Appendix 2: MTBO performance on a synthetic problem

The results in the main text focused on an empirical analysis of MTBO performance on real value model tuning experiments. Here we also provide a synthetic problem with characteristics similar to that of the real problems discussed in the text. The code to produce the results in this section is available at `http://ax.dev`.

The simulator contributes to the optimization in Algorithm 1 in two ways. The first is that after the random initialization of both online and offline observations (Line 2), we have seen that the MTGP has much lower prediction error than can be obtained without the simulator. Lower prediction error in the first optimized batch means that the acquisition function will select better points and reach the optimum more quickly. After the initialization, the simulator is used by interleaving offline and online batches. Optimized points selected by the acquisition function are tested first on the simulator, and then a filtered set based on those results are actually deployed online. This helps to increase the quality of the optimized batch prior to being observed online, which further accelerates the optimization.

The main goal of these tests on the synthetic problem are to gain insight into the relative contributions of these two factors. We thus compared three models: a single-task GP using only online points; an MTGP using Algorithm 1 as described in the main text, which used the offline task as a look-ahead by evaluating a larger set of candidates offline, updating the model, and launching online the subset that maximized utility; and finally an MTGP that used offline data for the initialization but did not interleave offline and online batches after that. That is, in the third model lines 6 and 7 of Algorithm 1 were skipped, and the optimization proceeded exactly as with the single-task GP, except for the offline observations in the initialization.

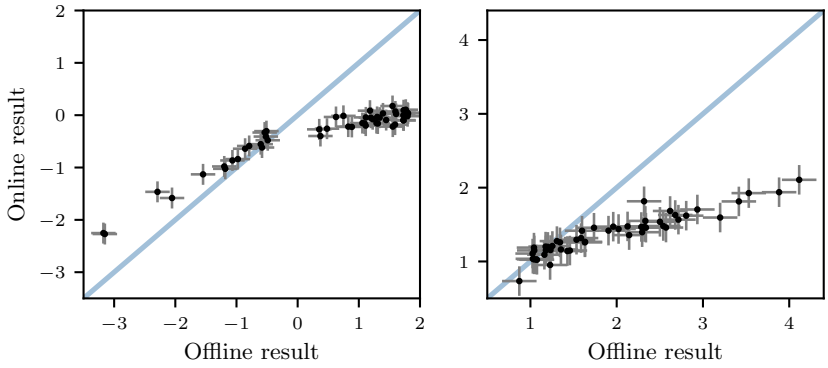

Figure S3: A comparison between "offline" and "online" results for the synthetic problem objective (left) and constraint (right). The offline evaluations produce a biased estimate of the true, online surfaces. Observations from both tasks have noise.

For the true, online surface we use the Hartmann 6 problem, a classic 6-parameter optimization test problem over the domain $x \in [0,1]^6$. We also simulated an additional outcome $g(x) = ||x||_2$ to be used as a constraint. As in the real application described in the main text, observations were made in batch either "online" or "offline." Online observations were direct function evaluations of the Hartmann 6 function $f(x)$ and constraint $g(x)$, with normally distributed noise added to each (standard deviation 0.1). Offline observations were generated by applying a nonlinear transform to the Hartmann 6 surface:

$$\tilde{f}(x) = \begin{cases} \alpha_1(x-m)+m, & \text{if } x \le m, \\ \alpha_2(x-m)+m, & \text{otherwise,} \end{cases}$$

where $m = 0.75$, $\alpha_1 = 0.4$, and $\alpha_2 = 0.8$. Offline observations for the constraint, $\tilde{g}(x)$ were generated via the same transform with $m = 1.25$, $\alpha_1 = 0.8$, and $\alpha_2 = 4$. Offline observations also had normally distributed noise added to them, with the same standard deviation 0.1.

Fig. S3 shows a comparison between online and offline observations for this synthetic problem evaluated on a collection of points in the design space. The bias constructed in this problem is modeled after that seen in the real application in Fig. 1 in the main text.

Bayesian optimization to minimize $f(x)$ subject to $g(x) \le 5/4$ was done with the three models described above. All methods began with the same $n_T = 5$ online observations at points from a Sobol sequence. The MTGPs were additionally given $n_S = 20$ offline points for their initialization. After that, all methods made online observations in batches of 5. For the MTGP using the loop in Algorithm 1, these were selected from an optimized batch of $n_o = 20$ points that were first tested offline. The optimization was done for a total of 20 online observations (4 batches of 5 points each), and was repeated 30 times, each with independent observation noise.

Fig. S4 shows the results of the optimization for each model, as the mean (over 30 runs) best feasible point tested by each iteration (online observation). These results show that both uses of the simulator have significant value. Comparing the single-task GP to

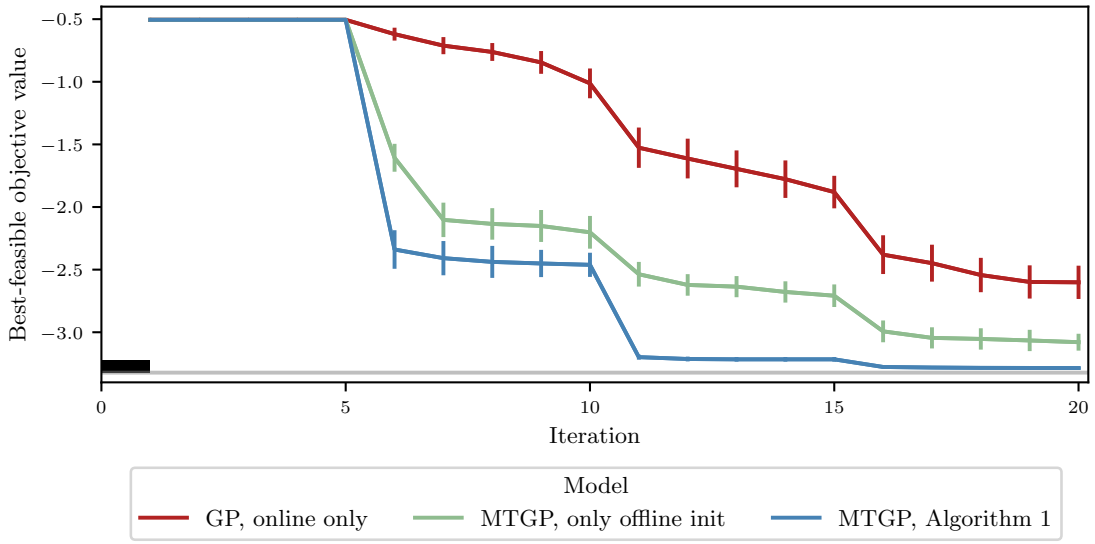

Figure S4: Bayesian optimization performance (best feasible objective value by each iteration) on the synthetic minimization problem for three methods: A GP making only online observations, an MTGP that uses offline observations only for the initialization, and an MTGP using Algorithm 1 and interleaving offline and online observations throughout the optimization. Iterations are online observations, and values shown are the mean and two standard errors across 30 repeated optimizations. Horizontal gray line is the global optimum. Offline observations accelerate the optimization compared to optimization with only online observations, and there is significant value to interleaving offline observations throughout the optimization.

the MTGP with offline observations in the initialization, we see that the improved model predictions allow for better points to found much earlier in the optimization. Comparing the two MTGP models, we see that while the initialization plays an important role early on, using offline observations as a pre-test before online observations, as is done in Algorithm 1, leads to significantly better performance at all iterations.