Supplement to Knowing The What But Not The Where in Bayesian Optimization

Vu Nguyen [1] **Michael A Osborne** [1]

1. Expected Regret Minimization Derivation

We are given an optimization problem $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathscr{X}} f(\mathbf{x})$ where f is a black-box function that we can evaluate pointwise. Let $\mathscr{D}_t = \{\mathbf{x}_i \in \mathscr{X}, y_i \in \mathscr{R}\}_{i=1}^t$ be the observation set including an input $\mathbf{x}_i$, an outcome y_i and $\mathscr{X} \in \mathscr{R}^d$ be the bounded search space. We define the regret function $r(\mathbf{x}) = f^* - f(\mathbf{x})$ where $f^* = \max_{\mathbf{x}\in\mathscr{X}} f(\mathbf{x})$ is the known global optimum value. The likelihood of the regret $r(\mathbf{x})$ on a normal posterior distribution is as follows

$$p(r(\mathbf{x})) = \frac{1}{\sqrt{2\pi}\sigma(\mathbf{x})} \exp\left(-\frac{1}{2}\frac{[\mu(\mathbf{x}) - f^* + r(\mathbf{x})]^2}{\sigma^2(\mathbf{x})}\right). \tag{1}$$

The expected regret can be written using the likelihood function in Eq. (1) to obtain, $\mathbb{E}[r(\mathbf{x})]$

$$= \int_0^\infty \frac{r(\mathbf{x})}{\sqrt{2\pi}\sigma(\mathbf{x})} \exp\left(-\frac{1}{2}\frac{[\mu(\mathbf{x}) - f^* + r(\mathbf{x})]^2}{\sigma^2(\mathbf{x})}\right) dr(\mathbf{x}).$$

As the ultimate goal in optimization is to minimize the regret, we consider our acquisition function to minimize this expected regret as $\alpha^{\text{ERM}}(\mathbf{x}) = \mathbb{E}[r(\mathbf{x})]$. Let $t = \frac{\mu(\mathbf{x}) - f^* + r(\mathbf{x})}{\sigma(\mathbf{x})}$, then $r(\mathbf{x}) = t \times \sigma(\mathbf{x}) - \mu(\mathbf{x}) + f^*$ and $dt = \frac{dr}{\sigma(\mathbf{x})}$. We write $\alpha^{\text{ERM}}(\mathbf{x})$

$$\begin{aligned}
&= \int_{t=\frac{\mu(\mathbf{x})-f^*}{\sigma(\mathbf{x})}}^{\infty} \frac{t\times\sigma(\mathbf{x}) + f^* - \mu(\mathbf{x})}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2) dt \\
&= \sigma(\mathbf{x}) \int_{t=\frac{\mu(\mathbf{x})-f^*}{\sigma(\mathbf{x})}}^{\infty} \frac{t}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2) dt \\
&\quad + [f^* - \mu(\mathbf{x})] \int_{t=\frac{\mu(\mathbf{x})-f^*}{\sigma(\mathbf{x})}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2) dt.
\end{aligned} \tag{2}$$

[1]University of Oxford, UK. Correspondence to: Vu Nguyen <vu@robots.ox.ac.uk>.

Proceedings of the 37th International Conference on Machine Learning, Online, PMLR 119, 2020.

We compute the first term in Eq. (2) as

$$\begin{aligned}
&\sigma(\mathbf{x}) \int_{t=\frac{\mu(\mathbf{x})-f^*}{\sigma(\mathbf{x})}}^{\infty} \frac{t}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2) dt \\
&= \frac{\sigma(\mathbf{x})}{\sqrt{2\pi}} \left[-\exp\left(-\frac{1}{2}\left[\frac{\mu(\mathbf{x}) - f^* + r(\mathbf{x})}{\sigma(\mathbf{x})}\right]^2\right)\right]_{r=0}^{r=\infty} \\
&= \sigma(\mathbf{x}) \mathscr{N}\left(\frac{\mu(\mathbf{x}) - f^*}{\sigma(\mathbf{x})} \mid 0, 1\right).
\end{aligned}$$

Next, we compute the second term in Eq. (2) as

$$\begin{aligned}
&[f^* - \mu(\mathbf{x})] \int_{t=\frac{\mu(\mathbf{x})-f^*}{\sigma(\mathbf{x})}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2) dt \\
&= [f^* - \mu(\mathbf{x})] \left\{\int_{-\infty}^{\infty} \mathscr{N}(t \mid 0,1)\, dt - \int_{-\infty}^{\frac{\mu(\mathbf{x})-f^*}{\sigma(\mathbf{x})}} \mathscr{N}(t \mid 0,1)\, dt\right\} \\
&= [f^* - \mu(\mathbf{x})] \left[1 - \Phi\left(\frac{\mu(\mathbf{x}) - f^*}{\sigma(\mathbf{x})}\right)\right] \\
&= [f^* - \mu(\mathbf{x})] \Phi\left(\frac{f^* - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right).
\end{aligned}$$

Let $z = \frac{f^* - \mu(\mathbf{x})}{\sigma(\mathbf{x})}$, we obtain the acquisition function

$$\alpha^{\text{ERM}}(\mathbf{x}) = \sigma(\mathbf{x})\phi(z) + [f^* - \mu(\mathbf{x})]\Phi(z) \tag{3}$$

where $\phi(z) = \mathscr{N}(z \mid 0, 1)$ is the standard normal p.d.f. and $\Phi(z)$ is the c.d.f. To select the next point, we minimize this acquisition function which is equivalent to minimize the expected regret $\mathbb{E}[r(\mathbf{x})]$

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathscr{X}} \alpha^{\text{ERM}}(\mathbf{x}) = \arg\min_{\mathbf{x}\in\mathscr{X}} \mathbb{E}[r(\mathbf{x})].$$

We can see that this acquisition function is minimized $\alpha^{\text{ERM}}(\mathbf{x}_t) = \mathbb{E}[r(\mathbf{x}_t)] = 0$ when $f^* = \mu(\mathbf{x}_t)$ and $\sigma(\mathbf{x}_t) = 0$. Our chosen point $\mathbf{x}_t$ is the one which offers the smallest expected regret. We aim to find the point with the desired property of $\mathbb{E}[r(\mathbf{x}_t)] = \mathbb{E}[f(\mathbf{x}_t) - f(\mathbf{x}^*)] = 0$.

2. Additional Experiments

We first illustrate the BO with and without the knowledge of f^*. Then, we provide additional information about the deep reinforcement learning experiment in the main paper. Next, we compare the effect of using the vanilla GP and transformed GP with different acquisition functions.

Table 1. Hyperparameters of Advantage Actor Critic.

Variables	Min	Max	Best Parameter $\mathbf{x}^*$
γ discount factor	0.9	1	0.95586
learning rate q model	$1e^{-6}$	0.01	0.00589
learning rate v model	$1e^{-6}$	0.01	0.00037

2.1. Illustration per iteration

We provide the illustration of BO with and without the knowledge of f^* for comparison in Figs. 1 and 2. We show the GP and EI in the left (without f^*) and the transformed GP and ERM in the right (with f^*). As the effect of transformation using f^*, the transformed GP (right) can lift up the surrogate model closer to the true value f^* (red horizontal line) encouraging the acquisition function to select at these potential locations. On the other hand, without f^*, the GP surrogate (left) is less informative. As a result, the EI operating on GP (left) is less efficient as opposed to the transformed GP. We demonstrate visually that using TGP our model can finally find the optimum input within the evaluation budget while the standard GP does not.

2.2. Details of A2C on CartPole problem

We use the advantage actor critic (A2C) () as the deep reinforcement learning algorithm to solve the CartPole problem (). This A2C is implemented in Tensorflow and run on a NVIDIA GTX 2080 GPU machine. In A2C, we use two neural network models to learn $Q(s,a)$ and $V(s)$ separately. In particular, we use a simple neural network architecture with 2 layers and 10 nodes in each layer. The range of the used hyperparameters in A2C and the found optimal parameter are summarized in Table 1.

We illustrate the reward performance over 500 training episodes using the found optimal parameter $\mathbf{x}^* = \underset{\mathbf{x}\in\mathscr{X}}{\text{argmax}} f(\mathbf{x})$ value in Fig. 3. In particular, we plot the raw reward and the average reward over 100 consecutive episodes - this average score is used as the evaluation output. Our A2C with the found hyperparameter will take around 300 episodes to reach the optimum value $f^* = 200$.

2.3. Comparison using vanilla GP and transformed GP

We empirically compare the proposed transformed Gaussian process (using the knowledge of f^*) and the vanilla Gaussian process () as the surrogate model for Bayesian optimization. We then test our ERM and EI on the two surrogate models. After the experiment, we learn that the transformed GP is more suitable for our ERM while it may not be ideal for the EI.

ERM. We perform experiments on ERM acquisition function using two surrogate models as vanilla Gaussian process (GP) and transformed Gaussian process (TGP). Our acquisition function performs better with the TGP. The TGP exploits the knowledge about the optimum value f^* to construct the surrogate model. Thus, it is more informative and can be helpful in high dimension functions, such as Alpine1 $D = 5$ and gSobol $D = 5$, $D = 10$, in which the ERM on TGP achieves much better performance than ERM on GP. On the simpler functions, such as branin and hartmann, the transformed GP surrogate achieves comparable performances with the vanilla GP. We visualize all results in Fig. 4.

Expected Improvement (EI). We then test the EI acquisition function on two surrogate models of vanilla Gaussian process and our transformed Gaussian process (using f^*) in Fig. 5. In contrast to the case of ERM above, we show that the EI will perform well on the vanilla GP, but not on the TGP. This can be explained by the side effect of the GP transformation as follows. From Eq. (1) in the main paper, when the location has poor (or low) prediction value $\mu(\mathbf{x}) = f^* - \frac{1}{2}\mu_g^2(\mathbf{x})$, we will have large value $\mu_g(\mathbf{x})$. As a result, this large value of $\mu_g(\mathbf{x})$ will make the uncertainty larger $\sigma(\mathbf{x}) = \mu_g(\mathbf{x})\sigma_g(\mathbf{x})\mu_g(\mathbf{x})$ from Eq. (2) in the main paper. Therefore, TGP will make an additional uncertainty $\sigma(\mathbf{x})$ at the location where $\mu(\mathbf{x})$ is low.

Under the additional uncertainty effect of TGP, the expected improvement may spend more iterations to explore these uncertainty area and take more time to converge than the case of using the vanilla GP. We note that this effect will also happen to the GP-UCB and other acquisition functions, which rely on exploration-exploitation trade-off.

In high dimensional function of gSobol $D = 10$, TGP will make the EI explore aggressively due to the high uncertainty effect (described above) and thus result in worse performance. That is, it keeps exploring at poor region in the first 100 iterations (see bottom row of Fig. 5).

Discussion. The transformed Gaussian process (TGP) surrogate takes into account the knowledge of optimum value f^* to inform the surrogate. However, this transformation may create additional uncertainty at the area where function value is low. While our proposed acquisition function ERM and CBM will not suffer this effect, the existing acquisition functions of EI and UCB will. Therefore, we only recommend to use this TGP with our acquisition functions for the best optimization performance.

3. Other known optimum value settings

To highlight the applicability of the proposed model, we list several other settings where the optimum values are known

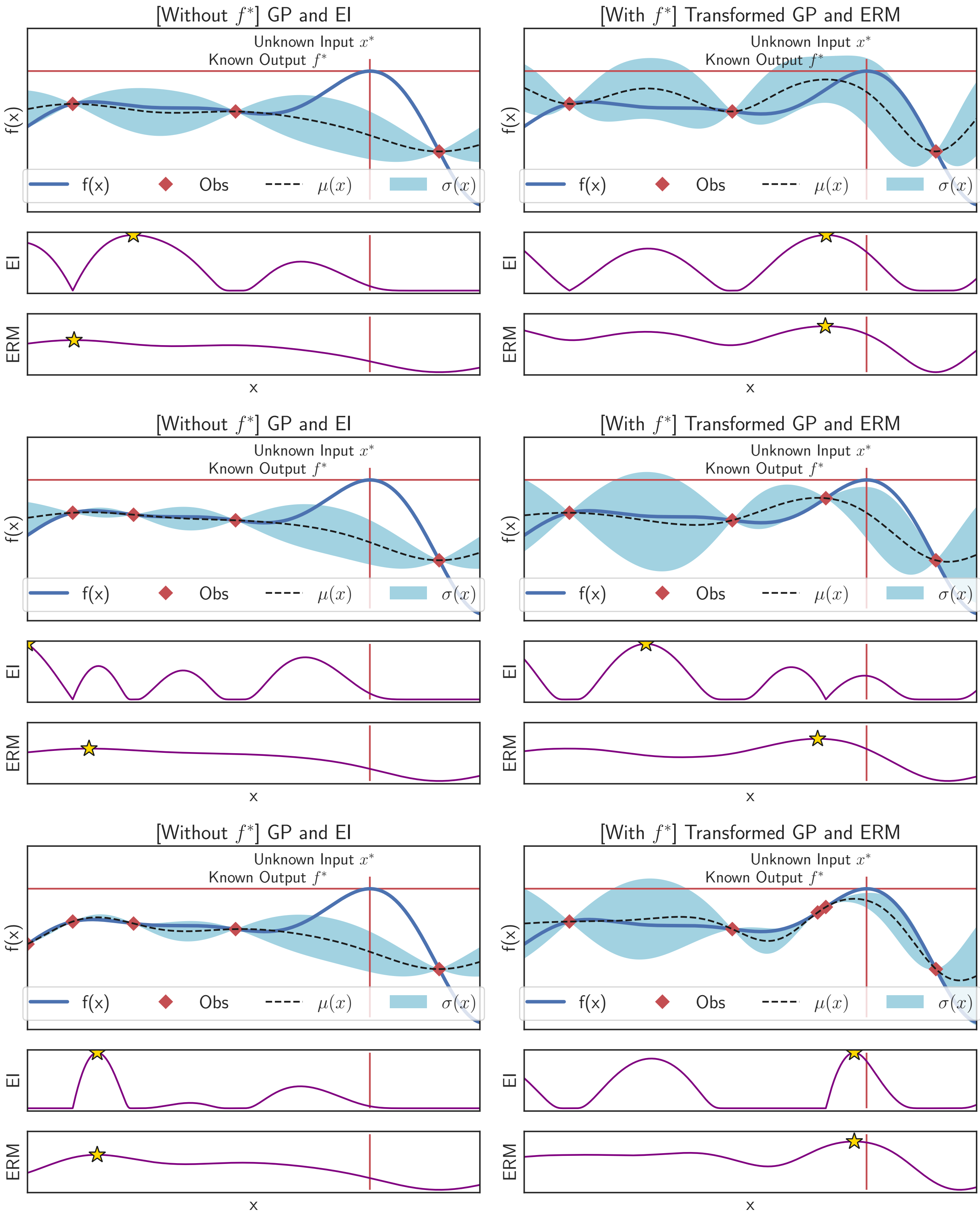

Figure 1. Illustration of the optimization process per iteration $1-3$ starting given the same initialization. Left: BO using GP as surrogate and EI as acquisition function. Right: BO using TGP as surrogate and ERM as acquisition function. Given the known optimum f^* value, the transformed GP can lift up the surrogate model closer to the known value. Then, the ERM will make informed decision given f^*. We also show that the EI may not make the best decision as ERM. To be continue in the next figure.

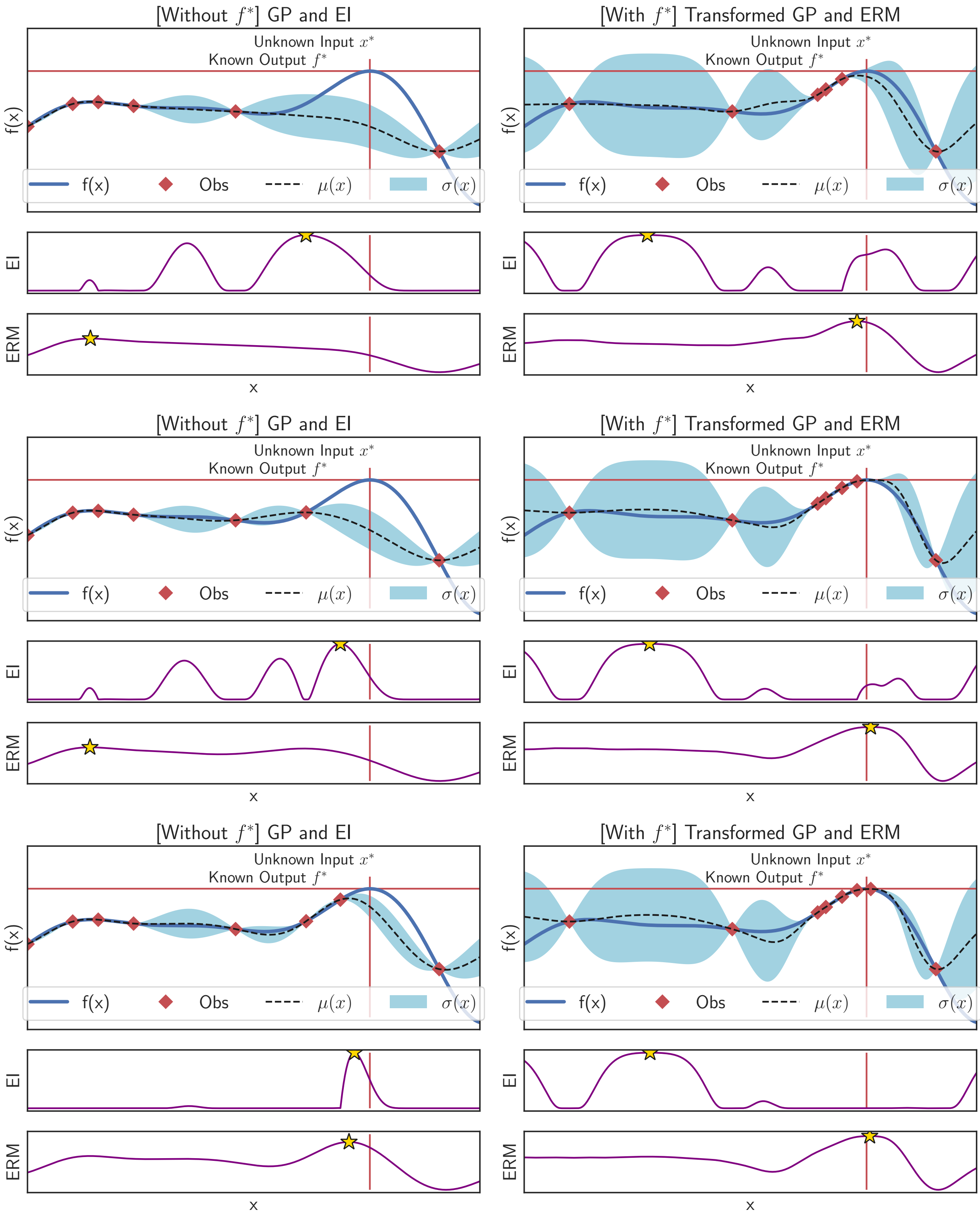

Figure 2. Continuing from the previous figure. Illustration of the optimization process per iteration $4-6$ starting given the same initialization. Left: BO using GP as surrogate and EI as acquisition function. Right: BO using TGP as surrogate and ERM as acquisition function. Given the known optimum f^* value, the transformed GP can lift up the surrogate model closer to the known value. Then, the ERM will make informed decision given f^*. We also show that the EI may not make the best decision as ERM.

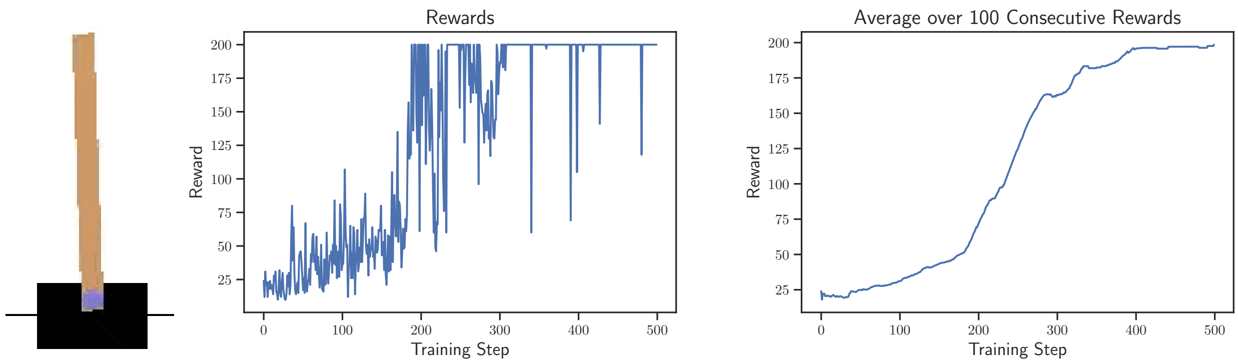

Figure 3. Left: visualization of a CartPole. Middle and Right: visualization of the reward curve using the best found parameter value $\mathbf{x}^*$. We have used the Advantage Actor Critic (A2C) algorithm to solve the CartPole problem. The known optimum value is $f^* = 200$.

Table 2. Examples of known optimum value settings.

Environment	f^*	Source
Pong	18	Gym.OpenAI
Frozen Lake	0.79	Gym.OpenAI
Inverted Pendulum v1	135.91	Gym.OpenAI
CartPole	200	Gym.OpenAI

in Table 2.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

Andrew G Barto, Richard S Sutton, and Charles W Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, (5):834–846, 1983.

Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

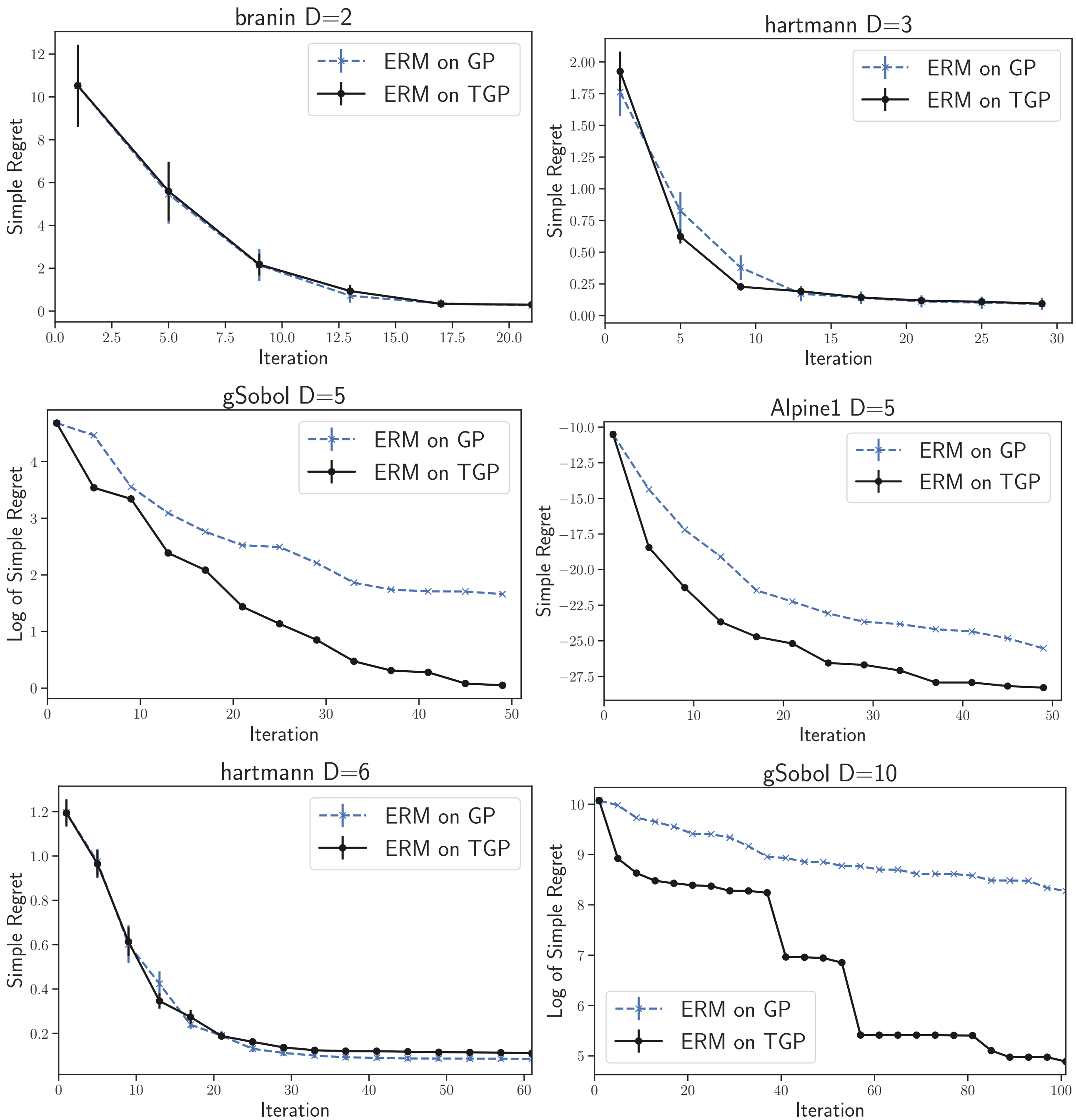

Figure 4. Experiments with ERM acquisition function on vanilla Gaussian process (GP) and transformed Gaussian process (TGP). Our acquisition function using the transformed GP consistently performs better than using the vanilla GP. Particularly, the TGP will be more useful in high-dimensional functions of Alpine1 $D = 5$ and gSobol $D = 5$, $D = 10$ functions. In these functions, ERM on TGP will outperform ERM on GP by a wide margin.

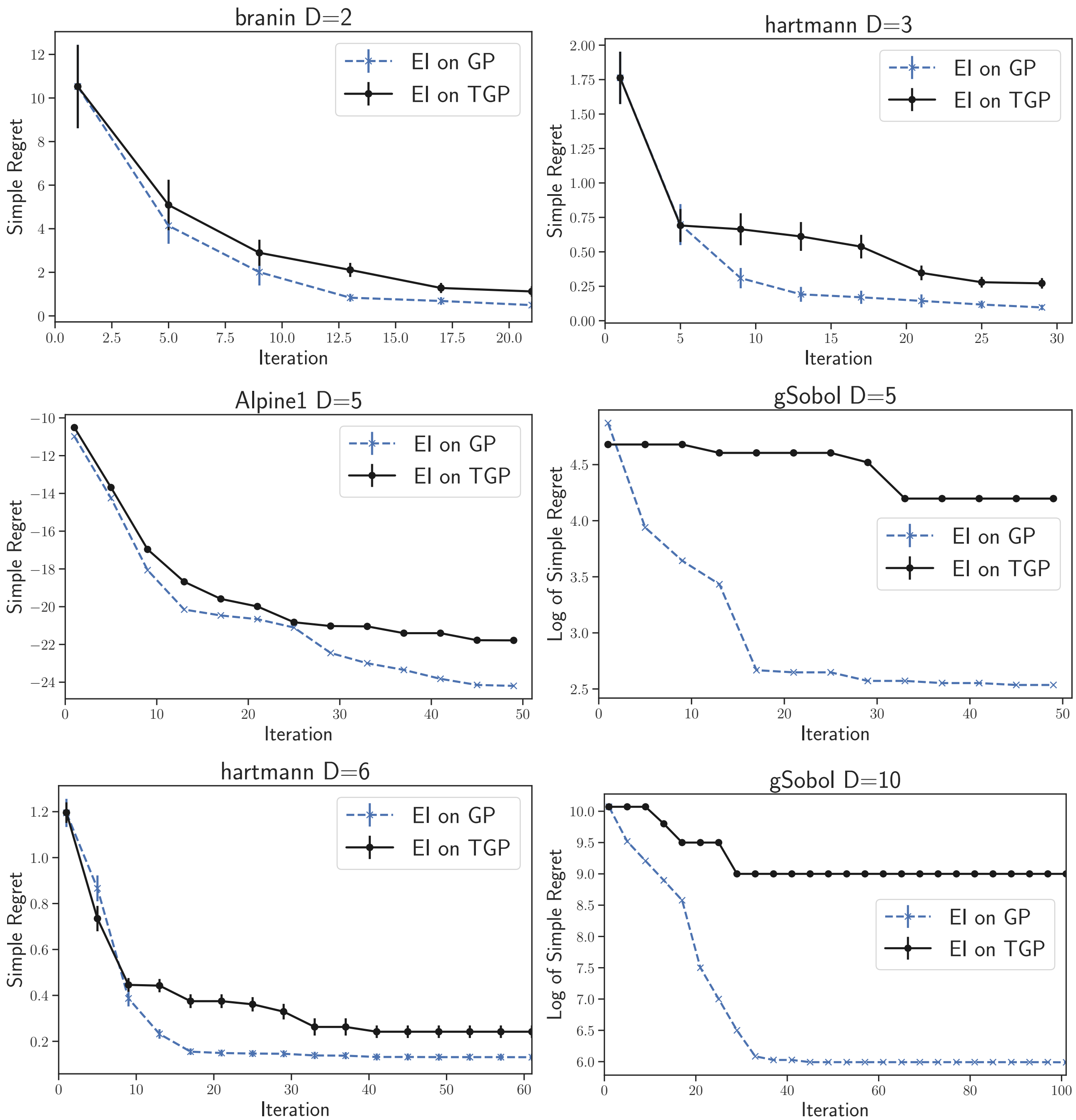

Figure 5. Experiments with EI acquisition function using the surrogate models as GP and TGP. Although the TGP exploits the knowledge about the optimum value f^* to construct the informed surrogate model, it brings the side effect of transformation in making additional GP predictive uncertainty. As a result, the EI will explore more aggressively using TGP and thus obtain worse performance comparing to the case of using vanilla GP.