Preferential Bayesian Optimization with Hallucination Believer

Shion Takeno
Nagoya Institute of Technology
Aichi, Japan
takeno.s.mllab.nit@gmail.com

Masahiro Nomura
CyberAgent
Tokyo, Japan
nomura_masahiro@cyberagent.co.jp

Masayuki Karasuyama
Nagoya Institute of Technology
Aichi, Japan
karasuyama@nitech.ac.jp

Abstract

We study preferential Bayesian optimization (BO) where reliable feedback is limited to pairwise comparison. An important challenge in preferential BO, which uses the Gaussian process (GP) model to represent preference structure, is that the posterior distribution is computationally intractable. Existing preferential BO methods either suffer from poor posterior approximation ignoring the skewness or require computationally expensive approximation for the exact posterior represented as a skew GP. In this work, we develop a simple and computationally efficient preferential BO algorithm while keeping the superior optimization performance. The basic idea is to use a posterior additionally conditioned by a random sample from the original posterior itself, called hallucination, by which we show that a usual GP-based acquisition function can be used while reflecting the skewness of the original posterior. The numerical experiments on the various benchmark problems demonstrate the effectiveness of the proposed method.

1 Introduction

Preferential Bayesian optimization (BO) has been an attractive approach for solving problems where reliable feedback is limited to *pairwise comparison*, the so-called *duels*. This preference setting often appears in *human-in-the-loop optimization* problems such as visual design optimization [19] and generative melody composition [32] because it is easier for humans to judge which one is better than to give an absolute rating [17]. The system (i.e., the optimization method) in the human-in-the-loop optimization presents choices and receives preferential feedback *interactively*. To reduce the waiting time for users, the system is required to *quickly* present the new options to users by learning from the observed feedback information.

An important challenge in preferential BO, which uses the Gaussian process (GP) model to represent preference structure, is that the posterior distribution is computationally intractable. The existing approaches to this difficulty are twofold: The first approach is the Gaussian approximation (e.g., Laplace approximation and Expectation propagation) [9, 8], which leads to computationally efficient preferential BO algorithms [7, 23, 15, 28, 10]. However, the accuracy of the approximation is often poor [4], as the Gaussian approximation *ignores* the skewness of the exact posterior represented as a skew GP [3–5]. The second approach is to directly employ the skew GP model using the Markov chain Monte Carlo (MCMC) method [3–5]. Although using the skew GP leads to superior optimization performance, the MCMC requires a heavy computational time. Reducing the computational

2022 NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems.

complexity can be critical in preferential BO because, for example, the computational time directly becomes the waiting time for users in applications involving human interactions.

In this work, we develop a simple and computationally efficient preferential BO algorithm while keeping the superior optimization performance coming from skew GP. Our approach mainly relies on the insight that the skew GP additionally conditioned by variables that control the skewness is reduced to the standard GP (Proposition 3.1). Then, the proposed method computes an acquisition function based on the GP conditioned on a random sample from the true posterior, called *hallucination*. This reduction via conditioning gives the proposed method the following two strengths: Firstly, the proposed method can take into account the skewness of the true posterior. Secondly, any powerful acquisition function developed so far in the standard BO literature (e.g., UCB [30], EI [21], and TS [26]) can be integrated. Our experiments show that the proposed method achieves a significant speedup in terms of computational time and at least competitive performance in terms of sample complexity over the state-of-the-art preferential BO approaches [15, 31, 4, 10].

2 Background

We consider that the preferential relation is modeled by a latent function $f : \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X} \in \mathbb{R}^d$ is input domain. Our goal is to maximize the latent function as

$$\boldsymbol{x}_* = \arg\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}),$$

through the dueling feedback, $\boldsymbol{x} \succ \boldsymbol{x}'$, which implies $\boldsymbol{x}$ is preferable to $\boldsymbol{x}'$.

We assume that f is a sample path of $\mathcal{GP}(0, k)$ with some stationary kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. Suppose that we have multiple duels $\mathcal{D}_t \coloneqq \{\boldsymbol{x}_{i,w} \succ \boldsymbol{x}_{i,l}\}_{i=1}^t$, where $\boldsymbol{x}_{i,w}$ is the winner of the duel and $\boldsymbol{x}_{i,l}$ is the loser. Following [9, 8], we assume that the duel is determined as follows:

$$\boldsymbol{x}_{i,w} \succ \boldsymbol{x}_{i,l} \Leftrightarrow f(\boldsymbol{x}_{i,w}) + \epsilon_w > f(\boldsymbol{x}_{i,l}) + \epsilon_l,$$

where i.i.d. additive noise ϵ_w and ϵ_l follow the normal distribution $\mathcal{N}(0, \sigma_{\text{noise}}^2)$. This is equivalent to assuming that the preferences are obtained by which a direct observation $y = f(\boldsymbol{x}) + \epsilon$ is bigger or smaller, where $\epsilon \sim \mathcal{N}(0, \sigma_{\text{noise}}^2)$.

The exact posterior distribution $p(f \mid \mathcal{D}_t)$ is skew GP, as shown in [4, 5]. By the definition, the conditioning by duels $\mathcal{D}_t$ can be rewritten as $\{v_i < 0\}_{i=1}^t$, where $v_i \coloneqq f(\boldsymbol{x}_{i,l}) + \epsilon_l - f(\boldsymbol{x}_{i,w}) - \epsilon_w$. For brevity, we denote $\{v_i < 0\}_{i=1}^t$ as $\boldsymbol{v}_t < \boldsymbol{0}$, where $\boldsymbol{v}_t \coloneqq (v_1, \dots, v_t)^\top$. Then, the posterior density function for all $\boldsymbol{x} \in \mathcal{X}$ is written as,

$$p\left(f(\boldsymbol{x}) \mid \mathcal{D}_t\right) = p\left(f(\boldsymbol{x}) \mid \boldsymbol{v}_t < \boldsymbol{0}\right) = \frac{\Pr\left(\boldsymbol{v}_t < \boldsymbol{0} \mid f(\boldsymbol{x})\right) p\left(f(\boldsymbol{x})\right)}{\Pr(\boldsymbol{v}_t < \boldsymbol{0})}.$$

Then, $\boldsymbol{v}_t$, which we referred to as the *latent truncated variable*, controls the skewness. Since the prior for $\boldsymbol{v}_t$ and $f(\boldsymbol{x})$ is multivariate normal (MVN) distribution, both $\Pr(\boldsymbol{v}_t < \boldsymbol{0})$ and $\Pr\left(\boldsymbol{v}_t < \boldsymbol{0} \mid f(\boldsymbol{x})\right)$ are cumulative distribution function (CDF) of MVN (See Appendix A for the details). Furthermore, statistics of $f(\boldsymbol{x}) \mid \boldsymbol{v}_t < \boldsymbol{0}$, such as mean and variance, is computed using CDF of MVN [2]. However, CDF of MVN [12, 11] is computationally expensive. Thus, Benavoli et al. [4, 5] employed the posterior sampling-based approximation, for which they showed that the sampling from skew GP could be performed through the sampling from truncated MVN.

3 Preferential Bayesian optimization with hallucination believer

The state-of-the-art preferential BO methods are based on Gaussian approximation or MCMC-based posterior approximation. The former suffers poor prediction performance ignoring the skewness of the posterior, and the latter requires heavy computational time. In this work, we propose a computationally efficient preferential BO method while taking into account the skewness in a randomized manner.

Motivation for the proposed method comes from the following observation: The skew GP conditioned by the latent truncated variable is reduced to the standard GP (Proposition 3.1). The important thing is to condition the hallucination from the true posterior, which makes taking into account skewness possible. Algorithm 1 shows the procedure of the proposed method, *hallucination believer* (HB). The

Algorithm 1 Hallucination believer for preferential BO

Require: $\mathcal{D}_0 = \{\boldsymbol{x}_{0,w}, \boldsymbol{x}_{0,l}\}$, $\mathcal{X}$
1: **for** $t = 1, \dots$ **do**
2: $\quad \boldsymbol{x}_t^{(1)} \leftarrow \boldsymbol{x}_{t-1,w}$
3: $\quad$ Generate $\tilde{\boldsymbol{v}}_{t-1}$ from $p(\boldsymbol{v}_{t-1} \mid \boldsymbol{v}_{t-1} < \mathbf{0})$ and $\tilde{y}_t^{(1)}$ from $p(y_t^{(1)} \mid \tilde{\boldsymbol{v}}_{t-1})$, respectively
4: $\quad \boldsymbol{x}_t^{(2)} \leftarrow \arg\max_{\boldsymbol{x} \in \mathcal{X}} \alpha(\boldsymbol{x})$ based on the GP $f \mid \tilde{\boldsymbol{v}}_{t-1}, \tilde{y}_t^{(1)}$
5: $\quad$ Set $\boldsymbol{x}_{t,w}$ and $\boldsymbol{x}_{t,l}$ as the winner and loser of the duel between $\boldsymbol{x}_t^{(1)}$ and $\boldsymbol{x}_t^{(2)}$, respectively
6: **end for**

proposed method iteratively selects a pair of inputs by the following two steps: (i) Select the winner of the past duels as the first point[1] (line 2). (ii) Using the posterior distribution conditioned by the hallucination, select the point that maximizes the acquisition function as the second point (lines 3-4). We can use an arbitrary computationally efficient and powerful acquisition function for the usual BO due to the reduction via the conditioning.

3.1 Hallucination believer

The following proposition shows that conditioning the latent truncated variables reduce the skew GP to the standard GP. The proof is shown in Appendix A.

Proposition 3.1. *Exact posterior distribution additionally conditioned by $\boldsymbol{v}_{t-1}$ is $p(f \mid \boldsymbol{v}_{t-1} < \mathbf{0}, \boldsymbol{v}_{t-1}) = p(f \mid \boldsymbol{v}_{t-1})$, which is a GP.*

Still, there remains the question of what value should be conditioned to the skew GP. For example, if we condition on a constant value, such as the posterior mean, the proposed method ignores skewness and the optimization performance will deteriorate. To reflect the skewness of the true posterior, we condition the hallucination $\tilde{\boldsymbol{v}}_{t-1}$ generated from the posterior $p(\boldsymbol{v}_{t-1} \mid \boldsymbol{v}_{t-1} < \mathbf{0})$. Thus, the skewness is incorporated in a randomized manner by $\tilde{\boldsymbol{v}}_{t-1}$ affecting to the GP $p(f \mid \boldsymbol{v}_{t-1} = \tilde{\boldsymbol{v}}_{t-1})$ according to the true posterior. In addition, we conditioned $y_t^{(1)} := f(\boldsymbol{x}_t^{(1)}) + \epsilon$ so that a more exploratory $\boldsymbol{x}_t^{(2)}$ should be selected. Note that $\tilde{y}_t^{(1)}$ can be easily generated from the normal distribution $p(y_t^{(1)} \mid \tilde{\boldsymbol{v}}_{t-1})$. See Appendix A for the derivation of GP $p(f \mid \tilde{\boldsymbol{v}}_{t-1}, \tilde{y}_t^{(1)})$.

We employed Gibbs sampling [20] for the sampling from the truncated MVN $p(\boldsymbol{v}_{t-1} \mid \boldsymbol{v}_{t-1} < \mathbf{0})$, whose details are shown in Appendix C. Although the sampling $\tilde{\boldsymbol{v}}_{t-1}$ needs MCMC, HB needs only one sample, whose sampling is sufficiently fast (See Appendix E.1 for a computational time of MCMC). Thus, HB is highly efficient compared to [4, 5], which need many MC samples. HB is related to the kriging believer [27], which is a well-known heuristic in parallel BO literature, in the sense that some additional conditioning to the posterior is performed. We provide the detailed discussion in Appendix B.

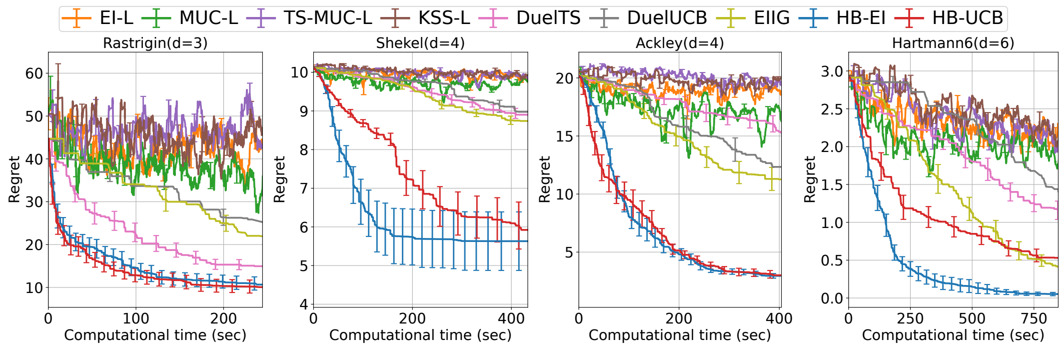
4 Experiments

We investigate the effectiveness of the proposed method through comprehensive numerical experiments. We employed the eight benchmark functions. In this section, we show the results for Rastrigin, Shekel, Ackley, and Hartmann6 functions, and others are shown in Appendix E. We performed the proposed methods combining expected improvement (EI) [21] and upper confidence bound (UCB) [30], denoted as HB-EI and HB-UCB, respectively. We employed the baseline methods, EI [7], MUC [10], TS-MUC[2] [15], KSS [31], DuelTS [4], DuelUCB [4], and EIIG[3] [4], where DuelTS, DuelUCB, and EIIG are based on skew GP and others employed Laplace approximation. As a performance measure, we used the regret defined as $f(\boldsymbol{x}_*) - f(\tilde{\boldsymbol{x}}_t)$, where $\tilde{\boldsymbol{x}}_t$ is a recommendation point

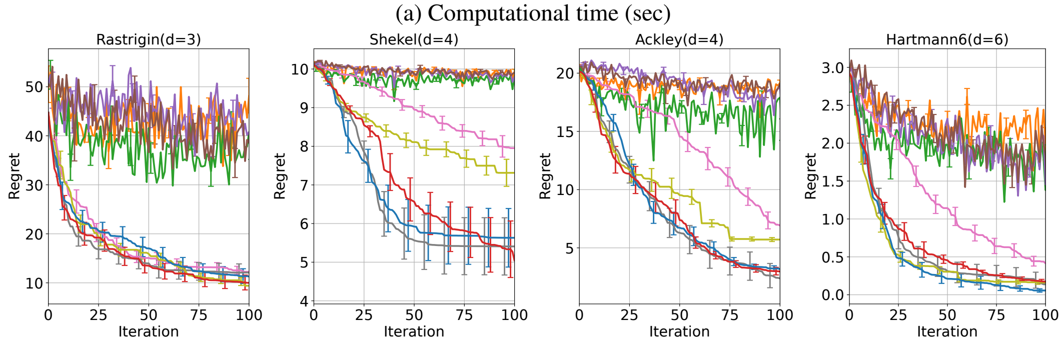
[1] Some prior works [7, 23, 10] employed the maxima of the posterior mean as the first input. In contrast, we employ the winner so far as the first input, as the posterior mean of skew GP is computationally expensive.

[2] This method was originally proposed as dueling TS, whose name is same as [4]. To distinguish them, we denote TS-MUC since this method selects the first input by TS and the second input by MUC.

[3] Benavoli et al. [4, 5] proposed the acquisition function that EI minus information gain (IG). We conjecture that this is a typo since both EI and IG should be large. Thus, we used a modified one that EI plus IG.



(a) Computational time (sec)



(b) Iteration

Figure 1: Performance comparison of the proposed method (HB-EI and HB-UCB) with the state-of-the-art preferential BO methods. The horizontal axis represents (a) the computational time (sec) and (b) the number of iterations. The vertical axis represents the regret, which is the smaller, the better it is. Suffix "-L" in the method name indicates using Laplace approximation.

at t-th iteration. We report the mean and standard error of the regret over 20 random initialization, where the initial duel is obtained as a uniformly random input pair. The details for settings of the kernel function, the baseline methods, and the recommendation point $\tilde{\boldsymbol{x}}_t$ are shown in Appendix D.

We first compare the Laplace approximation-based methods (denoted by Suffix "-L") with the skew GP-based methods including the proposed method. Figure 1(b) shows that the Laplace approximation-based methods are outperformed by the skew GP-based methods, which demonstrates the poor quality of the Laplace approximation. Next, among the skew GP-based methods, we compare the proposed methods with the MCMC-based methods [4, 5]. Figure 1(a) shows that the proposed method outperforms the other baseline methods including MCMC-based methods, which demonstrates the computational efficiency of the proposed method compared to MCMC-based methods. Furthermore, from the result of Figure 1(b), the proposed method is at least competitive in terms of iteration (i.e., sample complexity). These results indicate that the proposed method is highly efficient in practice compared to the existing preferential BO algorithms.

5 Conclusion

In this work, we developed a simple and computationally efficient preferential Bayesian optimization method, called hallucination believer (HB). We reduce the skew GP to the standard GP by conditioning the hallucination for the latent truncated variables, which leads to computational efficiency and superior optimization performance coming from the skew GP. The numerical experiments demonstrate the effectiveness of the proposed method in terms of both computational time and sample complexity. Interestingly, HB with a roughly approximated prediction by the hallucination achieves the performance that is comparable to the MCMC-based methods based on the exact posterior. Therefore, conditioning the hallucination can be more important than just making accurate predictions in preferential BO, and investigating why the conditioning can achieve such good sample complexity is one of the interesting future directions.

References

[1] Robert J Adler, Jose Blanchet, and Jingchen Liu. Efficient simulation for tail probabilities of Gaussian random fields. In *2008 Winter Simulation Conference*, pages 328–336. IEEE, 2008.

[2] Adelchi Azzalini. *The Skew-Normal and Related Families*. Institute of Mathematical Statistics Monographs. Cambridge University Press, 2013.

[3] Alessio Benavoli, Dario Azzimonti, and Dario Piga. Skew Gaussian processes for classification. *Machine Learning*, 109(9):1877–1902, 2020.

[4] Alessio Benavoli, Dario Azzimonti, and Dario Piga. Preferential Bayesian optimisation with skew Gaussian processes. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1842–1850, 2021.

[5] Alessio Benavoli, Dario Azzimonti, and Dario Piga. A unified framework for closed-form nonparametric regression, classification, preference and mixed problems with skew gaussian processes. *Machine Learning*, 110(11):3095–3133, 2021.

[6] J.A. Breslaw. Random sampling from a truncated multivariate normal distribution. *Applied Mathematics Letters*, 7(1):1–6, 1994. ISSN 0893-9659.

[7] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv.1012.2599*, 2010.

[8] Wei Chu and Zoubin Ghahramani. Extensions of Gaussian processes for ranking: semisupervised and active learning. In *Proceedings of the Learning to Rank workshop at the 18th Conference on Neural Information Processing Systems*, pages 29–34, 2005.

[9] Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 137–144, 2005.

[10] Tristan Fauvel and Matthew Chalk. Efficient exploration in binary and preferential Bayesian optimization. *arXiv preprint arXiv:2110.09361*, 2021.

[11] Marc G. Genton, David E. Keyes, and George Turkiyyah. Hierarchical decompositions for the computation of high-dimensional multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, pages 268–277, 2017.

[12] Alan Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–150, 1992.

[13] Alexandra Gessner, Oindrila Kanjilal, and Philipp Hennig. Integrals over Gaussians under linear domain constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 2764–2774. PMLR, 2020.

[14] John Geweke. Efficient simulation from the multivariate normal and student-t distributions subject to linear constraints and the evaluation of constraint probabilities. In *Computing science and statistics: Proceedings of the 23rd symposium on the interface*, pages 571–578, 1991.

[15] Javier González, Zhenwen Dai, Andreas Damianou, and Neil D Lawrence. Preferential bayesian optimization. In *International Conference on Machine Learning*, pages 1282–1291. PMLR, 2017.

[16] William C. Horrace. Some results on the multivariate truncated normal distribution. *Journal of Multivariate Analysis*, 94(1):209–221, 2005.

[17] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific, 2013.

[18] Jayesh H Kotecha and Petar M Djuric. Gibbs sampling approach for generation of truncated multivariate Gaussian random variables. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 3, pages 1757–1760. IEEE, 1999.

[19] Yuki Koyama, Issei Sato, and Masataka Goto. Sequential gallery for interactive visual design optimization. *ACM Transactions on Graphics (TOG)*, 39(4):88–1, 2020.

[20] Yifang Li and Sujit K Ghosh. Efficient sampling methods for truncated multivariate normal and student-t distributions subject to linear inequality constraints. *Journal of Statistical Theory and Practice*, 9(4):712–732, 2015.

[21] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

[22] Iain Murray, Ryan Adams, and David MacKay. Elliptical slice sampling. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 541–548. JMLR Workshop and Conference Proceedings, 2010.

[23] Jens Brehm Bagger Nielsen, Jakob Nielsen, and Jan Larsen. Perception-based personalization of hearing aids using Gaussian processes and active learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(1):162–173, 2015.

[24] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[25] Christian P Robert. Simulation of truncated normal variables. *Statistics and computing*, 5(2): 121–125, 1995.

[26] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

[27] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104 (1):148–175, 2016.

[28] Eero Siivola, Akash Kumar Dhaka, Michael Riis Andersen, Javier González, Pablo García Moreno, and Aki Vehtari. Preferential batch Bayesian optimization. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2021.

[29] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc., 2012.

[30] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 1015–1022. Omnipress, 2010.

[31] Yanan Sui, Vincent Zhuang, Joel W Burdick, and Yisong Yue. Multi-dueling bandits with dependent arms. *arXiv preprint arXiv:1705.00253*, 2017.

[32] Yijun Zhou, Yuki Koyama, Masataka Goto, and Takeo Igarashi. Interactive exploration-exploitation balancing for generative melody composition. In *26th International Conference on Intelligent User Interfaces*, pages 43–47, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We believe that our study does not have any potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Appendix A
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We did not include the code since this is still a workshop paper and we will extend the study.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and Appendix D

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] Our experiments can be performed in usual laptop.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Details of skew GP

Let us consider the prediction with the test points $\boldsymbol{X}_{\text{tes}} := (\boldsymbol{x}_{1,\text{tes}}, \dots, \boldsymbol{x}_{n_{\text{tes}},\text{tes}})^\top$, where $n_{\text{tes}} \in \mathbb{N}$. We define the matrix of the inputs $\boldsymbol{X} := \left(\boldsymbol{x}_{1,\text{tes}}, \dots, \boldsymbol{x}_{n_{\text{tes}},\text{tes}}, \boldsymbol{x}_{1,w}, \dots, \boldsymbol{x}_{t,w}, \boldsymbol{x}_{1,l}, \dots, \boldsymbol{x}_{t,l}\right)^\top \in \mathbb{R}^{(n_{\text{tes}}+2t)\times d}$, where d is the input dimension. Furthermore, we denote the i-th row of $\boldsymbol{X}$ as $\boldsymbol{X}_i$. Then, the prior for $\boldsymbol{f} := (f(\boldsymbol{X}_0), \dots, f(\boldsymbol{X}_{n_{\text{tes}}+2t}))^\top \in \mathbb{R}^{n_{\text{tes}}+2t}$ is,

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}),$$

where (i,j)-th element of $\boldsymbol{K}$ is $k(\boldsymbol{X}_i, \boldsymbol{X}_j)$. Then, $\boldsymbol{f}_{\text{tes}}$ and $\boldsymbol{v}_t$ are linear combinations of $\boldsymbol{f}$, the noise ϵ_w, and ϵ_l. Therefore, we see that,

$$\begin{bmatrix} \boldsymbol{f}_{\text{tes}} \\ \boldsymbol{v}_t \end{bmatrix} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\text{tes},\boldsymbol{v}}), \tag{1}$$

where

$$\boldsymbol{A} := \begin{bmatrix} \boldsymbol{I}_{n_{\text{tes}}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I}_t & \boldsymbol{I}_t \end{bmatrix} \in \mathbb{R}^{(n_{\text{tes}}+t)\times(n_{\text{tes}}+2t)},$$

$$\boldsymbol{B} := \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma^2_{\text{noise}} \boldsymbol{I}_{2t} \end{bmatrix} \in \mathbb{R}^{(n_{\text{tes}}+2t)\times(n_{\text{tes}}+2t)},$$

$$\boldsymbol{\Sigma}_{\text{tes},\boldsymbol{v}} := \boldsymbol{A}\big(\boldsymbol{K} + \boldsymbol{B}\big)\boldsymbol{A}^\top \in \mathbb{R}^{(n_{\text{tes}}+t)\times(n_{\text{tes}}+t)},$$

and $\boldsymbol{I}_i \in \mathbb{R}^{i\times i}$ is the identity matrix.

Then, we revisit that the posterior $p(\boldsymbol{f}_{\text{tes}} \mid \boldsymbol{v}_t < \boldsymbol{0})$ is obtained as

$$p\left(\boldsymbol{f}_{\text{tes}} \mid \boldsymbol{v}_t < \boldsymbol{0}\right) = \frac{\Pr\left(\boldsymbol{v}_t < \boldsymbol{0} \mid \boldsymbol{f}_{\text{tes}}\right) p\left(\boldsymbol{f}_{\text{tes}}\right)}{\Pr(\boldsymbol{v}_t < \boldsymbol{0})}.$$

From the prior (1), $\boldsymbol{v}_t$ and $\boldsymbol{v}_t \mid \boldsymbol{f}_{\text{tes}}$ follow MVN. Therefore, $\Pr\left(\boldsymbol{v}_t < \boldsymbol{0} \mid \boldsymbol{f}_{\text{tes}}\right)$ and $\Pr(\boldsymbol{v}_t < \boldsymbol{0})$ are CDF of MVN. Consequently, we can see that this posterior is a multivariate unified skew normal distribution [2, 4], with latent skewness dimension t.

Next, we consider the posterior additionally conditioned by $\boldsymbol{v}_t$ and prove the Proposition 3.1. First, for convenience, we define the submatrices as follows:

$$\boldsymbol{\Sigma}_{\text{tes},\boldsymbol{v}} := \begin{bmatrix} \boldsymbol{\Sigma}_{\text{tes},\text{tes}} & \boldsymbol{\Sigma}_{\text{tes},\boldsymbol{v}} \\ \boldsymbol{\Sigma}_{\text{tes},\boldsymbol{v}}^\top & \boldsymbol{\Sigma}_{\boldsymbol{v},\boldsymbol{v}} \end{bmatrix},$$

where $\mathbf{\Sigma}_{\mathrm{tes},\mathrm{tes}} \in \mathbb{R}^{n_{\mathrm{tes}} \times n_{\mathrm{tes}}}$, $\mathbf{\Sigma}_{\mathrm{tes},\boldsymbol{v}} \in \mathbb{R}^{n_{\mathrm{tes}} \times t}$, $\mathbf{\Sigma}_{\boldsymbol{v},\boldsymbol{v}} \in \mathbb{R}^{t \times t}$. The joint posterior distribution of $\boldsymbol{f}_{\mathrm{tes}}$ and $\boldsymbol{v}_t$ is truncated MVN, in which $\boldsymbol{v}_t$ is truncated above at $\mathbf{0}$. From the property of truncated MVN, Conclusion 5 in [16], the conditional distribution of truncated MVN is truncated MVN keeping the original truncation, in which the parameters can be computed as with usual MVN. More precisely, by using equations above Conclusion 5 in [16], we see that

$$p(\boldsymbol{f}_{\mathrm{tes}} \mid \boldsymbol{v}_t, \boldsymbol{v}_t < \mathbf{0}) = \frac{p(\boldsymbol{f}_{\mathrm{tes}}, \boldsymbol{v}_t)}{\int_{\mathbb{R}^{n_{\mathrm{tes}}}} p(\boldsymbol{f}_{\mathrm{tes}}, \boldsymbol{v}_t) \mathrm{d}\boldsymbol{f}_{\mathrm{tes}}},$$

where, since $\boldsymbol{f}_{\mathrm{tes}}$ is not truncated, i.e., $\boldsymbol{f}_{\mathrm{tes}} \in \mathbb{R}^{n_{\mathrm{tes}}}$, the region of integration is $\mathbb{R}^{n_{\mathrm{tes}}}$. Therefore, we obtain

$$\begin{aligned} p(\boldsymbol{f}_{\mathrm{tes}} \mid \boldsymbol{v}_t, \boldsymbol{v}_t < \mathbf{0}) &= \frac{p(\boldsymbol{f}_{\mathrm{tes}}, \boldsymbol{v}_t)}{p(\boldsymbol{v}_t)} \\ &= p(\boldsymbol{f}_{\mathrm{tes}} \mid \boldsymbol{v}_t). \end{aligned}$$

Consequently, the conditional distribution $p(\boldsymbol{f}_{\mathrm{tes}} \mid \boldsymbol{v}_t, \boldsymbol{v}_t < \mathbf{0})$ is MVN:

$$\boldsymbol{f}_{\mathrm{tes}} \mid \boldsymbol{v}_t, \boldsymbol{v}_t < \mathbf{0} \sim \mathcal{N}(\mathbf{\Sigma}_{\mathrm{tes},\boldsymbol{v}}^\top \mathbf{\Sigma}_{\boldsymbol{v},\boldsymbol{v}}^{-1} \boldsymbol{v}_t, \mathbf{\Sigma}_{\mathrm{tes},\mathrm{tes}} - \mathbf{\Sigma}_{\mathrm{tes},\boldsymbol{v}}^\top \mathbf{\Sigma}_{\boldsymbol{v},\boldsymbol{v}}^{-1} \mathbf{\Sigma}_{\mathrm{tes},\boldsymbol{v}}),$$

which is equivalent to the distribution $p(\boldsymbol{f}_{\mathrm{tes}} \mid \boldsymbol{v}_t)$. Hence, since above derivation can be applied to any $\boldsymbol{X}_{\mathrm{tes}}$, we can see that $f \mid \boldsymbol{v}_t$ is a GP.

By setting $\boldsymbol{x}_{\mathrm{tes}}$ as $\boldsymbol{x}_t^{(1)}$ and adding $\sigma^2_{\mathrm{noise}}$ to the variance, we can obtain the posterior $p(y_{\boldsymbol{x}_t^{(1)}} \mid \boldsymbol{v}_{t-1})$. Conditioning $y_{\boldsymbol{x}_t^{(1)}}$ can be performed as same as the derivation of conditional MVN [24].

B Difference between hallucination and kriging believer

Kriging believer (KB) [27] is a well-known heuristic for parallel BO. KB conditions on the ongoing function evaluation by the posterior mean, and then the next batch point is selected using this GP conditioned by the posterior mean. Some variants, called constant liar, use some predefined constant instead of the posterior mean. Furthermore, some studies averaged the resulting acquisition function value by the sample of the posterior normal distribution [e.g., 29]. These studies aim to guarantee the diversity of the batch points via penalization by conditioning.

On the other hand, one of the important aims of HB is to reduce the skew GP to the standard GP. For this purpose, we conditioned the latent truncated variable $\boldsymbol{v}_t$, which is not related to parallel BO. Furthermore, in preferential BO, if we conditioned the constant including the posterior mean, preferential BO methods cannot consider the skewness. Thus, the conditioning by the constant results in poor performance as with Gaussian approximation. On the other hand, although averaging by the samples from the posterior is promising, it requires huge computational time for MCMC with respect to skew GP. Hence, we employed the conditioning by the hallucination of $\boldsymbol{v}_t$. The conditioning of $y_{\boldsymbol{x}_t^{(1)}}$ is used for the same purpose of KB, i.e., the penalization.

C Gibbs sampling for truncated MVN

For the sampling from truncated MVN, Benavoli et al. [4, 5] used linear elliptical slice sampling (LinESS) [13], the variant of elliptical slice sampling [22]. On the other hand, many sampling methods for truncated MVN have been proposed and are based on Gibbs sampling [1, 6, 14, 18, 20, 25]. In this work, we employed standard Gibbs sampling combining the efficient rejection sampling for univariate truncated normal [20], which uses several proposal distributions depending on the truncation. We empirically observed that the Gibbs sampling works better than LinESS in terms of both the autocorrelation and computational time, which are shown in Appendix E.1.

Gibbs sampling is often used for the sampling of truncated MVN [1, 20, 14, 6]. Let us consider the sampling from truncated MVN $p(\boldsymbol{v} | \boldsymbol{v} < \mathbf{0})$, where original $\boldsymbol{v} \in \mathbb{R}^n$ follows MVN below:

$$\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{\Sigma}),$$

where $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is an arbitrary covariance matrix. Let v_j and $\boldsymbol{v}_{-j}$ be j-th element of $\boldsymbol{v}$ and the vector consisting of the elements except for v_j, respectively. In Gibbs sampling, we repeat the

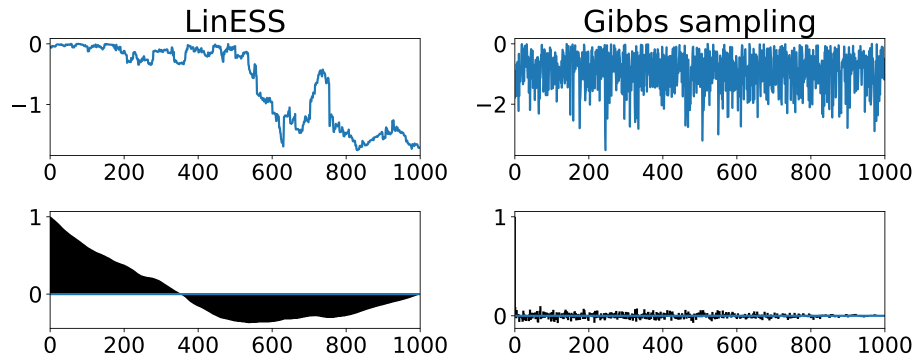


Figure 2: The trace plot (top) and autocorrelation plot (bottom), in which the Hartmann6 function is used with 50 uniformly random duels.

sampling $v_j \mid \boldsymbol{v}_{-j}, v_j < 0$, which follows uni-variate truncated normal distribution. The conditional distribution $v_j \mid \boldsymbol{v}_{-j} \sim \mathcal{N}(\mu_j, \sigma_j^2)$, where μ_j and σ_j^2 are computed efficiently by computing $\boldsymbol{\Sigma}^{-1}$ once [Section 5.4.2 in 24]. Algorithm 2 shows the procedure of Gibbs sampling, in which $[\cdot]_j$ and $[\cdot]_{jj}$ imply the j-th element of the vector and (j, j)-th element of the matrix, respectively. For the sampling from uni-variate truncated normal, we employed the efficient rejection sampling [Section 2.1 in 20], which uses several proposal distributions depending on the truncation.

Algorithm 2 Gibbs sampling for truncated MVN

Require: $\boldsymbol{v}^{(0)}$, $\boldsymbol{\Sigma}$
  Compute $\boldsymbol{\Sigma}^{-1}$
  for $i = 1, \dots$ **do**
    $\boldsymbol{v}^{(i)} \leftarrow \boldsymbol{v}^{(i-1)}$
    for $j = 1, \dots, n$ **do**
      $\mu_j^{(i)} \leftarrow [\Sigma^{-1}\boldsymbol{v}^{(i)}]_j / [\Sigma^{-1}]_{jj}$
      Set $v_j^{(i)}$ by the sampling from $\mathcal{N}(\mu_j^{(i)}, \sigma_j^2 = 1/[\Sigma^{-1}]_{jj})$ with truncation above at $\mathbf{0}$
    end for
  end for

D Experimental settings

All the details of benchmark functions are shown in `https://www.sfu.ca/~ssurjano/optimization.html`. In each function, we used the RBF kernel with automatic relevance determination [24], whose hyper-parameters are chosen using the usual marginal likelihood maximization of the GP regression model with a direct observation y beforehand. Although [4] originally employed LinESS for the sampling from truncated MVN, we employed Gibbs sampling in DuelTS, DuelUCB, and EIIG as with our proposed methods, for a fair comparison. For the parameters for Gibbs sampling, burn-in is 1000, thinning is 10, and MC sample size for DuelUCB and EIIG is 1000. Other settings for existing methods, such as the percentage for UCB, are set as with the suggestion from the paper. We set the recommendation point $\tilde{\boldsymbol{x}}_t$ as $\boldsymbol{x}_{t,w}$ in DuelTS, DuelUCB, EIIG, HB-EI, and HB-UCB following [4]. In other methods, $\tilde{\boldsymbol{x}}_t$ is set by the maximization of the posterior mean computed by Laplace approximation.

E Additional experiments

In this section, we provide additional experimental results.

Table 1: The computational time of MCMC for Figure 2. The mean and standard error with 20 random trials are shown.

	LinESS	Gibbs sampling
Computational time (sec)	1.70 ± 0.04	0.75 ± 0.04

E.1 Comparison between Gibbs sampling and LinESS

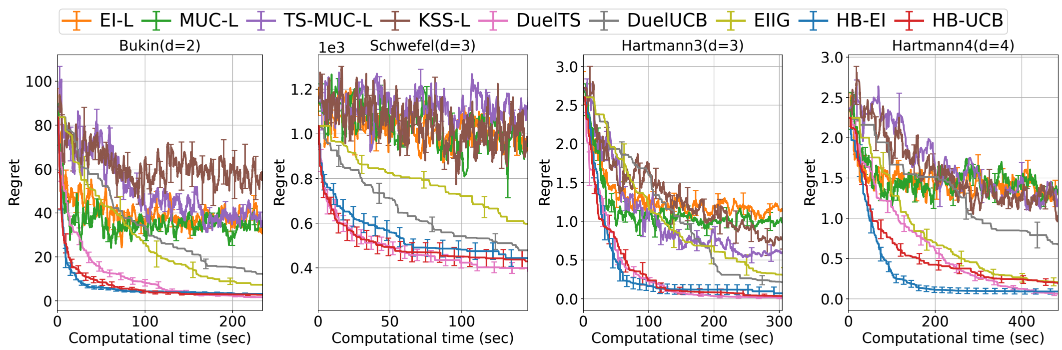
We compared the Gibbs sampling and LinESS for truncated MVN. For LinESS, we used the implementation by [4, 5] (`https://github.com/benavoli/SkewGP` with BSD-3-Clause license).

For comparison, we generate 50 uniformly random duels using the Hartmann6 function. Then, we perform the sampling of $\boldsymbol{v}_t \mid \boldsymbol{v}_t < 0$. Figure 2 shows the trace and autocorrelation plot in 1000 iterations for the first element of $\boldsymbol{v}_t$. We can observe that Gibbs sampling has lower autocorrelation than LinESS. This tendency can be confirmed in other elements. Furthermore, Table 1 shows the mean and standard error of the computational time for 20 random trials. If we consider that the burn-in is set as 1000, this time can be seen as a computational time to generate one sample. We confirmed that Gibbs sampling is 2x faster than LinESS.

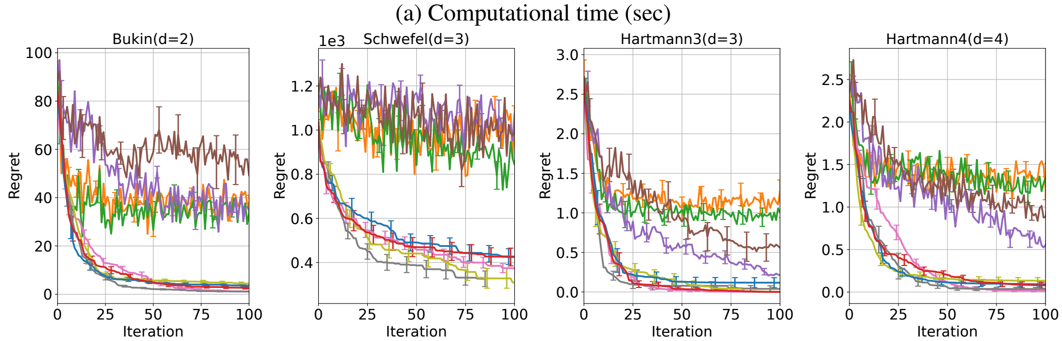
Note that LinESS was originally proposed for truncated MVN with many linear truncations [13]. LinESS is expected to be effective when the number of truncations is much larger than the dimension of MVN since LinESS is a rejection-free sampling method in contrast to Gibbs sampling. Inversely, if the number of truncations is huge, a rejection sampling-based Gibbs sampling suffers a low acceptance rate. On the other hand, in this case for $\boldsymbol{v}_t \mid \boldsymbol{v}_t < \boldsymbol{0}$, both dimensions are t. We conjecture that this is the reason why Gibbs sampling is efficient compared to LinESS in our experiments.

E.2 Experimental results for other benchmark functions

Figure 3 shows the results for Bukin, Schwefel, Hartmann3, and Hartmann4. In these plots, we can still confirm that the proposed methods, HB-EI and HB-UCB, show superior performance in terms of both computational time and iteration.



(a) Computational time (sec)



(b) Iteration

Figure 3: Additional performance comparison of the proposed method (HB-EI and HB-UCB) with the state-of-the-art preferential BO methods. The horizontal axis represents (a) the computational time (sec) and (b) the number of iterations. The vertical axis represents the regret, which is the smaller, the better it is. Suffix “-L” in the method name indicates using Laplace approximation.