Exploring and Exploiting Model Uncertainty in Bayesian Optimization

Zishi Zhang[1*], **Tao Ren**[1*], **Yijie Peng**[1,2†]
[1] Guanghua School of Management, Peking University
[2] Xiangjiang Laboratory
{zishizhang,rtkenny}@stu.pku.edu.cn, pengyijie@pku.edu.cn

Abstract

In this work, we consider the problem of Bayesian Optimization (BO) under *reward model uncertainty*—that is, when the underlying distribution type of the reward is unknown and potentially intractable to specify. This challenge is particularly evident in many modern applications, where the reward distribution is highly ill-behaved, often non-stationary, multi-modal, or heavy-tailed. In such settings, classical Gaussian Process (GP)-based BO methods often fail due to their strong modeling assumptions. To address this challenge, we propose a novel surrogate model, the infinity-Gaussian Process (∞-GP), which represents a sequential spatial Dirichlet Process mixture with a GP baseline. The ∞-GP quantifies both value uncertainty and model uncertainty, enabling more flexible modeling of complex reward structures. Combined with Thompson Sampling, the ∞-GP facilitates principled exploration and exploitation in the distributional space of reward models. Theoretically, we prove that the ∞-GP surrogate model can approximate a broad class of reward distributions by effectively exploring the distribution space, achieving near-minimax-optimal posterior contraction rates. Empirically, our method outperforms state-of-the-art approaches in various challenging scenarios, including highly non-stationary and heavy-tailed reward settings where classical GP-based BO often fails.

1 Introduction

Bayesian Optimization (BO) [Frazier, 2018] is a powerful framework for optimizing expensive-to-evaluate black-box functions and has found broad applications in hyperparameter tuning for machine learning models [Snoek et al., 2012], robotics [Berkenkamp et al., 2023], biology [Amin et al., 2024], and reinforcement learning [Brochu et al., 2010]. BO typically proceeds by fitting a surrogate model to the observed reward collected so far, and then using an acquisition policy to guide the selection of the next location to query. The surrogate model plays a central role in both learning and exploration.

In almost all BO applications, Gaussian Processes (GPs) are used as the default surrogate model. This practice assumes that the deterministic objective function $\mu^*(x)$ and the noise $\epsilon^*(x)$ satisfy strong structural conditions, essentially requiring the true objective to "resemble" a GP. Specifically, $\mu^*(x)$ is often assumed to be a sample path drawn from GP [Russo and Van Roy, 2014, Wang et al., 2025] or lie in a reproducing kernel Hilbert space (RKHS) with known bounded norm [Srinivas et al., 2010, 2012, Chowdhury and Gopalan, 2017] (the connection between GP regression and RKHS regression is detailed in Kanagawa et al. [2018]), and $\epsilon^*(x)$ is assumed to be independent and (sub-)Gaussian, or even noiseless [Chen and Lam, 2023]. These assumptions are critical for theoretical guarantees,

[*] These authors contributed equally to this work.
[†] Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

but are rarely verified in practice. GP-based BO can fail when these assumptions are violated. For instance, GP surrogates struggle to model non-stationary rewards, which are commonly observed in applying BO in machine learning hyperparameter tuning [Snoek et al., 2014], and are inadequate for handling heavy-tailed noise, which frequently arises in financial and real-world BO tasks [Chowdhury and Gopalan, 2019, Cakmak et al., 2020]. Moreover, prompt optimization for large language models (LLMs) is another representative application [Sabbatella et al., 2024] where the reward model is of unknown and complex form. The reward typically involves multiple stages—such as combining with inputs, querying a black-box LLM, and interpreting outputs—each introducing variation and instability. The resulting reward landscape is highly unstable and non-stationary [Deng et al., 2022], as even slight variations in prompt formulation can lead to dramatically different outputs.

In this paper, we consider BO under *reward model uncertainty*—that is, the decision-maker is uncertain not only about the parameters of the reward distribution, but also about its underlying distributional type. While classical GP-based BO can effectively quantify *value uncertainty* under a fixed model, it fails to capture *model uncertainty*. Our goal is to develop a surrogate modeling and decision-making framework that remains robust under *reward model uncertainty*. To this end, we propose a novel surrogate model, the ∞-*Gaussian Process* (∞-GP), which represents an infinite mixture of Gaussian processes implemented via a sequential spatial Dirichlet process prior with a GP baseline. When combined with Thompson Sampling, this model automatically facilitates a principled exploration–exploitation trade-off in the distributional space, and can effectively converge toward a wide range of reward models.

Related Work. To address non-stationarity in BO, prior works have proposed various modifications, such as warping the input space of standard GP kernels [Snoek et al., 2014], replacing the surrogate model with neural networks [Li et al., 2023], or employing non-stationary kernels [Higdon et al., 2022, Seeger, 2004]. To handle heavy-tailed noise, a substantial body of work has been developed in the context of multi-armed bandits, i.e., problems with a finite decision space [Bubeck et al., 2013, Medina and Yang, 2016]. However, for BO with a continuous decision space, the available methods are much more limited. One notable exception is the work of Chowdhury and Gopalan [2019], who propose a truncation-based modification to the GP-UCB algorithm to handle heavy-tailed noises. Misspecified BO [Wynne et al., 2021, Bogunovic and Krause, 2021] addresses limited forms of model mismatch (e.g., incorrect smoothness) within fixed model classes such as GPs or RKHS functions, rather than accounting for distributional uncertainty over the entire distributional space. Mixture-of-experts GP models [Rasmussen and Ghahramani, 2001, Meeds and Osindero, 2005] also construct infinite GP mixtures using Dirichlet processes. However, they rely on a global DP to partition the input space, assigning each input to a single GP expert. In contrast, our model employs a spatial DP prior, allowing each location to mix over infinitely many GP surfaces. This leads to a fundamentally different formulation that offers a richer, nonparametric representation of model uncertainty and enables convergence over a broad class of reward distributions. Bariletto and Ho [2024] also considers combining nonparametric Bayesian models, such as DP, with BO. However, their focus is on distributionally robust optimization, and the integration is carried out fundamentally differently from ours.

Our Contributions The main contributions of this work are as follows:

- **A novel ∞-GP surrogate model.** We propose a new surrogate model for BO, the ∞-Gaussian Process (∞-GP), which quantifies both *model uncertainty* and *value uncertainty*. Combined with Thompson Sampling, it enables principled and efficient exploration and exploitation in the distributional space of the reward.

- **Theoretical guarantees.** We establish posterior convergence guarantees for the ∞-GP model, showing that it can approximate a broad class of reward distributions at a near-optimal minimax rate.

- **Strong empirical performance.** We demonstrate the effectiveness of our method on both synthetic benchmarks and real-world tasks where standard GP-based BO fails—particularly in scenarios involving heavy-tailed noise and non-stationary reward structures.

2 Preliminaries

Bayesian Optimization (BO). We consider the problem of maximizing an expensive-to-evaluate black-box function $\mu^*(x)$ over a compact domain $\mathcal{X} \subset \mathbb{R}^d$. At each iteration, the decision maker selects a point $x_i \in \mathcal{X}$ and observes a noisy reward $y(x_i) = \mu^*(x_i) + \epsilon^*(x_i)$, where $\epsilon^*(x)$ denotes stochastic noise. Since querying the function is costly, the goal is to identify the maximizer of $\mu^*(x)$ using as few evaluations as possible. BO tackles this problem by maintaining a probabilistic model—called a *surrogate model*—over the unknown objective. This surrogate is updated after each evaluation and serves as a cheap proxy for the reward. Based on the surrogate, an *acquisition policy* is used to determine the next query location. The policy aims to balance *exploration* (sampling uncertain regions) and *exploitation* (focusing on areas with high predicted values), thereby efficiently searching for the global optimum. Common acquisition strategies include Expected Improvement (EI), Upper Confidence Bound (UCB), Knowledge Gradient (KG), and Thompson Sampling (TS), which draws a random sample from the posterior and selects the maximizer.

Gaussian Process Surrogate Modeling and Kriging. Standard BO methods adopt a Gaussian Process (GP) as the surrogate model. The observed reward is modeled as $y(x_i) = m(x_i) + \xi(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \tau^2)$, where $m(x) = \beta^\top x$ denotes the deterministic mean trend, and $\xi \sim \mathcal{GP}(0, \sigma^2 \rho_{\boldsymbol{\phi}})$ is a zero-mean stationary GP over $\mathcal{X}$ with squared exponential kernel $\rho_{\boldsymbol{\phi}}(x, x') = \exp\left(-\sum_{k=1}^d \phi_k (x^{(k)} - x'^{(k)})^2\right)$, $\boldsymbol{\phi} = (\phi_1, \cdots, \phi_d)$, $x = (x^{(1)}, \cdots, x^{(d)})$. The combined term $m(x) + \xi(x)$ acts as a probabilistic surrogate for the true reward function $\mu^*(x)$. A key advantage of GP models lies in their closed-form posterior inference: given realized historical data $\xi(x_{1:n}) = \{\xi(x_1), \cdots, \xi(x_n)\}$ at evaluated locations $x_{1:n} = (x_1, \cdots, x_n)$, the posterior distribution of the entire function $\xi(\cdot)$ remains a GP. This is known as ***Kriging*** (or GP regression):

$$[\xi(\cdot) \mid \xi(x_{1:n})] \sim \mathcal{GP}(\mu_n(\cdot), \sigma_n^2(\cdot)), \tag{1}$$

$$\mu_n(\cdot) = \Sigma_0(\cdot, x_{1:n}) \Sigma_0^{-1}(x_{1:n}, x_{1:n}) \xi(x_{1:n}), \tag{2}$$

$$\sigma_n^2(\cdot) = \sigma^2 - \Sigma_0(\cdot, x_{1:n}) \Sigma_0^{-1}(x_{1:n}, x_{1:n}) \Sigma_0(x_{1:n}, \cdot). \tag{3}$$

Here, the kernel is $\Sigma_0(x, x') := \sigma^2 \rho_{\boldsymbol{\phi}}(x, x')$. The matrix $\Sigma_0(x_{1:n}, x_{1:n}) \in \mathbb{R}^{n \times n}$ denotes the covariance matrix with entries $[\Sigma_0]_{ij} = \Sigma_0(x_i, x_j)$, and $\Sigma_0(x, x_{1:n}) \in \mathbb{R}^{1 \times n}$ is the row vector of covariances between x and each observed input x_i.

Value Uncertainty vs. Model Uncertainty Another key strength of GP in BO is their ability to quantify uncertainty through the posterior variance. This enables principled exploration–exploitation (E&E) trade-offs, as in the UCB strategy, which directly uses the variance to guide exploration. However, the uncertainty quantified by a GP is inherently limited to *value uncertainty*—that is, uncertainty about the response value at a given location under a fixed reward distribution model. This overlooks a more fundamental source of uncertainty: whether the GP itself is an appropriate surrogate model for the underlying reward. We refer to this as *model uncertainty*, which captures the decision maker's uncertainty belief about the type or complexity of the true reward distribution. Consequently, standard GP-based BO requires strong assumptions for convergence, and GPs fail to capture complex reward models, such as those that are **non-stationary** or contaminated with **heavy-tailed noises**.

3 ∞-GP Surrogate Modeling

In this section, we develop a novel surrogate model called the ∞-Gaussian-Process (∞-GP), formally referred to as the *sequential spatial Dirichlet Process mixture with a Gaussian Process baseline*. The observed reward is modeled as

$$y(x_i) = m(x_i) + \xi(x_i) + \epsilon_i, \ \epsilon_i \sim N(0, \tau^2), \quad \xi(x_i) \overset{\text{ind}}{\sim} G_{x_i}, \tag{4}$$

where the deterministic mean term $m(x_i)$ is the same as classic GP model and the stochastic process $\xi \triangleq \{\xi(x) \in \mathbb{R} : x \in \mathcal{X}\}$ follows distribution $\{G_x : x \in \mathcal{X}\}$. To capture **model uncertainty** and explore the space of reward distributions, we do not fix the distribution $\{G_x : x \in \mathcal{X}\}$ as a GP. Instead, we assume that **the distribution $\{G_x : x \in \mathcal{X}\}$ is itself random** and follows a sequential Spatial Dirichlet Process (SDP) prior, which places a prior over the space of distributions:

$$\{G_x : x \in \mathcal{X}\} \sim \text{SDP}(\nu G_0). \tag{5}$$

Therefore, each reward $y(x_i)$ is generated via the following hierarchical process: (i) the nature draws a stochastic process $\{G_x : x \in \mathcal{X}\}$ from the distribution space under a SDP prior. (ii) in the i-th iteration, after determining the location x_i to evaluate, a sample $\xi(x_i)$ is generated from distribution G_{x_i}.

The SDP(νG_0) is characterized by two key quantities: a scalar precision parameter $\nu > 0$, and a baseline distribution G_0. In this work, we specify a stationary Gaussian Process $\mathcal{GP}(0, \sigma^2\rho_{\boldsymbol{\phi}}(\cdot,\cdot))$ over $\mathcal{X}$ as the baseline distribution. Specifically, a distribution (or stochastic process) arising from SDP(νG_0) is almost surely discrete and admits the representation

$$G_x = \sum_{l=1}^{\infty} w_l \delta_{\xi^{(l)}(x)}, \forall x \in \mathcal{X}, \tag{6}$$

where each surface $\xi^{(l)} \triangleq \{\xi^{(l)}(x) : x \in \mathcal{X}\}$ is a sample path independently drawn from the baseline $\mathcal{GP}(0, \sigma^2\rho_{\boldsymbol{\phi}}(\cdot,\cdot))$ and the weights $\{w_l\}_{l=1}^{\infty}$ admits the traditional "stick breaking" construction of Dirichlet process, i.e.,$w_1 = V_1,\ \ w_l = V_l \prod_r^{l-1}(1 - V_r),\ \ V_r \overset{iid}{\sim} Beta(1, \nu)$. Notably, our approach directly models the distribution of the observed reward $y(x)$, rather than $\mu^*(x)$. In this context, the additive noise term $\epsilon_i \sim N(0, \tau^2)$ in model (4) is not intended to impose a Gaussian assumption on the true observation noise. Instead, it is solely introduced for modeling purposes—to mix with the discrete-distributed $\xi(x)$ and produce a continuous reward distribution.

As defined in (6), each realized distribution $\{G_x : x \in \mathcal{X}\}$ from the SDP is a discrete measure over an infinite collection of surfaces $\{\xi^{(l)}\}_{l=1}^{\infty}$. At any location x_i, there is a positive probability that $\xi(x_i) = \xi^{(l)}(x_i)$ for some l, meaning that a draw $\xi(x_i) \sim G_{x_i}$ can be realized on any of the surfaces in the collection. To identify which surface x_i lies on, we introduce a latent variable $\boldsymbol{z}_{1:n} = \{z_i\}_{i=1}^n$, such that $z_i = l$ if $\xi(x_i) = \xi^{(l)}(x_i)$ and thus $\xi^{(z_i)}$ is the surface on which observation x_i is realized. Note that for any surface $\xi^{(j)}$ on which the observation at x_i is not realized, the corresponding value $\xi^{(j)}(x_i)$ remains latent and must also be inferred. As illustrated in Figure 1, solid dots (e.g., $\xi^{(2)}(x_5)$) indicate that the observation at x_5 is realized on surface $\xi^{(2)}$, while hollow dots (e.g., $\xi^{(1)}(x_3)$) indicate that the observation at x_3 is not realized on surface $\xi^{(1)}$. Since multiple observations may be realized on the same surface, the number of realized surfaces, denoted by K_n, typically grows slowly with n. Let $\boldsymbol{\xi}_{1:n} = (\xi^{(1)}, \ldots, \xi^{(K_n)})$ denote the collection of realized surfaces up to iteration n, and let n_j denote the number of locations within $\{x_1, \cdots, x_n\}$ that are realized on surface $\xi^{(j)}$, i.e., $n_j = \#\{i : z_i = j\}$, $j = 1, \cdots, K_n$.

It is important to note that the SDP was initially introduced in the spatial statistics and geostatistics literature [Gelfand et al., 2005, Quintana et al., 2022]. Our ∞-GP model incorporates two key modifications to adapt SDP for the sequential nature of BO. First, instead of assuming a fixed set of locations, we allow observations to be collected sequentially at arbitrary locations. Second, unlike the original SDP, where observations across all locations in each replication are drawn from the same surface, our model permits each observation to be realized on a potentially different surface, enabling greater flexibility.

Model Uncertainty Quantification It is important to understand the two quantities that characterize the SDP: the baseline distribution G_0 and the concentration parameter ν. For any measurable set B, if $G \sim$ SDP(νG_0), then $\mathbb{E}[G(B)] = G_0(B)$, meaning that G_0 serves as the mean measure. The parameter ν can be interpreted as the "variance" in the distribution space, controlling the variability of G around G_0: $\mathrm{Var}(G(B)) = \frac{G_0(B)(1-G_0(B))}{1+\nu}$. In practice, updating the posterior of ν with data allows dynamic quantification of *model uncertainty*: **the higher the ν, the more confident we are that the true reward distribution follows** G_0. In contrast, a small ν implies significant model uncertainty, in which case the model relies primarily on empirical distributions (detailed in the next section). Notably, although the baseline G_0 is a stationary GP, a sample path drawn from the ∞-GP can be **non-stationary** and exhibit different smoothness. This highlights the enhanced modeling flexibility of the ∞-GP surrogate.

Full Bayesian Treatment of Hyperparameters Following the seminal work of Snoek et al. [2012], we adopt a fully Bayesian approach for the hyperparameters $\Theta = \{\beta, \nu, \tau, \sigma^2, \phi\} = \{\Theta^{(1)}, \Theta^{(2)}\}$, where the first-layer parameters are $\Theta^{(1)} = \{\beta, \tau\}$ and the second-layer parameters are $\Theta^{(2)} =$

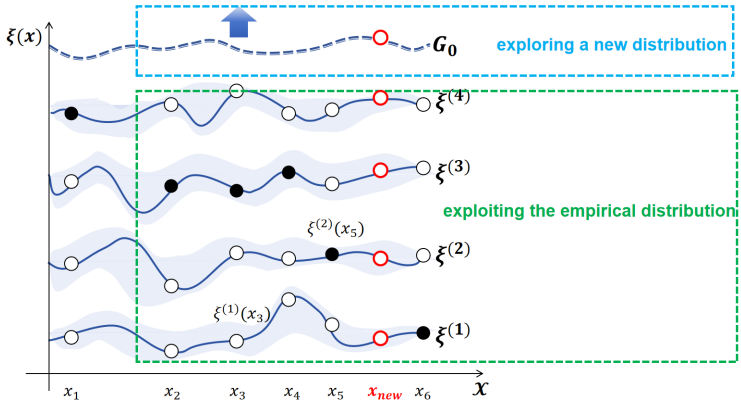

Figure 1: Illustration of the proposed ∞-GP model as a mixture over infinitely many GP surfaces. Each observation may be realized on any surface—solid dots indicate assignments to the corresponding surface, while hollow dots represent latent observations not realized on that surface. The red dot denotes a new input location x_{new}, which may either be assigned to an existing surface or initiate a new one drawn from the base measure.

$\{\nu, \sigma^2, \phi\}$. Instead of relying on frequentist estimators such as MLE, we place prior distributions on all components of Θ and update them via posterior inference. Details of the prior specifications are deferred to the Appendix.

4 ∞-GP Thompson Sampling: E&E in Distributional Space

Although the surrogate modeling in the previous section is performed on the noisy observed reward $y(x)$, the goal in BO is to maximize the expected reward $\mu^*(x) := \mathbb{E}[y(x)]$. The corresponding Thompson Sampling (TS) acquisition policy proceeds by drawing a sample from the posterior distribution of $\mu^*(x)$ given data $\mathcal{H}_n = \{(x_1, y(x_1)), \cdots, (x_n, y(x_n))\}$, and selecting its maximizer as the next query point, i.e.

$$x_{n+1} = \arg\max_{x \in \mathcal{X}} \hat{\mu}^*(x), \quad \hat{\mu}^* \sim p(\mu^* \mid \mathcal{H}_n). \tag{7}$$

In our hierarchical Bayesian model, at the $(n+1)$-th step, for any unobserved location $x_{n+1} \in \mathcal{X}$, the posterior distribution $\mu^*(x_{n+1}) \mid \mathcal{H}_n$ is implicitly induced by the joint posterior over latent components $\Theta' := \{\xi^{(z_{n+1})}(x_{n+1}), \Theta^{(1)}\}$, where $\xi^{(z_{n+1})}$ denotes the surface on which x_{n+1} is realized and $\Theta^{(1)} = \{\beta, \tau^2\}$. Specifically, the posterior distribution of $y(x_{n+1})$ is given by (see Appendix for proof):

$$\begin{aligned} f(y(x_{n+1}) \mid \mathcal{H}_n) &= \int f(y(x_{n+1}) \mid \Theta') \cdot f(\Theta' \mid \mathcal{H}_n) d\Theta' \\ &= \int\int \mathcal{N}(x_{n+1}^\top \beta + \xi^{(z_{n+1})}(x_{n+1}), \tau^2) \underbrace{f(\xi^{(z_{n+1})}(x_{n+1}) \mid \Theta^{(2)}, \boldsymbol{\xi}_{1:n}, \boldsymbol{z}_{1:n}) f(\Theta, \boldsymbol{\xi}_{1:n}, \boldsymbol{z}_{1:n} \mid \mathcal{H}_n)}_{f(\Theta' \mid \mathcal{H}_n)} \\ &\quad d\xi^{(z_{n+1})}(x_{n+1}) d\{\Theta, \boldsymbol{\xi}_{1:n}, \boldsymbol{z}_{1:n}\}, \end{aligned} \tag{8}$$

where $f(\cdot)$ denotes the probability density. According to this decomposition, given Θ', the distribution of $y(x_{n+1})$ is Gaussian: $y(x_{n+1})|\Theta' \sim \mathcal{N}(x_{n+1}^\top \beta + \xi^{(z_{n+1})}(x_{n+1}), \tau^2)$. Therefore, the (conditional) expected reward is $\mu^*(x_{n+1})|\Theta' = x_{n+1}^\top \beta + \xi^{(z_{n+1})}(x_{n+1})$.

Thompson Sampling Implementation. Therefore, the Thompson Sampling acquisition policy, as defined in Eq. (7), is implemented as follows:

(i) Draw a posterior sample $\hat{\Theta}' \sim f(\Theta' \mid \mathcal{H}_n)$ and (ii) Select the next location to evaluate:

$$x_{n+1} = \arg\max_{x \in \mathcal{X}} \hat{\mu}^*(x)|\hat{\Theta}' = \arg\max_{x \in \mathcal{X}} x^\top \hat{\beta} + \hat{\xi}^{(z_{n+1})}(x), \tag{9}$$

where the quantities with a hat (e.g., $\hat{\nu}, \hat{\beta}, \hat{K}_n, \{\hat{n}_j\}_{j=1}^{\hat{K}_n}$) denote posterior samples. The core challenge lies in drawing a posterior sample from $f(\Theta' \mid \mathcal{H}_n)$, particularly in sampling a path $\{\hat{\xi}^{(z_{n+1})}(x) : x \in \mathcal{X}\}$ from the posterior distribution. As shown in Eq. (8), we can achieve this by (i) first sampling a $\hat{\Theta}, \hat{\boldsymbol{\xi}}_{1:n}, \hat{\boldsymbol{z}}_{1:n}$ from the joint posterior $f(\Theta, \boldsymbol{\xi}_{1:n}, \boldsymbol{z}_{1:n} \mid \mathcal{H}_n)$ using a Gibbs sampler (see Appendix for details). (ii) Next, given $\hat{\Theta}, \hat{\boldsymbol{\xi}}_{1:n}, \hat{\boldsymbol{z}}_{1:n}$, we sample a path from the conditional posterior distribution $\{\xi^{(z_{n+1})}(x), x \in \mathcal{X}\} \mid \hat{\Theta}^{(2)}, \hat{\boldsymbol{\xi}}_{1:n}, \hat{\boldsymbol{z}}_{1:n}$. According to the Chinese restaurant process (CRP), a constructive representation of the Dirichlet process, a new upcoming data point either lies on one of the previously realized surfaces $\{\xi^{(j)}(\cdot)\}_{j=1}^{\hat{K}_n}$, or initiates a new surface drawn from the baseline distribution G_0 (as shown in the red dots in Figure 1). Based on this, we derive the following conditional distribution (see Appendix for proof):

$$f\big(\xi^{(z_{n+1})}(\cdot) \mid \hat{\Theta}^{(2)}, \hat{\boldsymbol{\xi}}_{1:n}, \hat{\boldsymbol{z}}_{1:n}\big) \sim \underbrace{P_n^{(0)} G_0(\cdot)}_{\text{Exploration}} + \underbrace{\sum_{j=1}^{\hat{K}_n} P_n^{(j)} \overbrace{\mathcal{GP}(\hat{\mu}_n^{(j)}(\cdot), \hat{\sigma}_n^2(\cdot))}^{\text{Kriging on } \xi^{(j)}},}_{\text{Exploitation}} \tag{10}$$

where $P_n^{(0)} = \frac{\hat{\nu}}{\hat{\nu}+n}$ and $P_n^{(j)} = \frac{\hat{n}_j}{\hat{\nu}+n}$. Each term $\mathcal{GP}(\hat{\mu}_n^{(j)}(\cdot), \hat{\sigma}_n^2(\cdot))$ corresponds to performing Kriging on an existing surface $\xi^{(j)}(\cdot)$, using past values $\xi^{(j)}(x_{1:n}) = (\xi^{(j)}(x_1), \cdots, \xi^{(j)}(x_n))$ realized on that surface. This follows the same calculation as in Eq. (1), with $\xi(x_{1:n})$ replaced by $\xi^{(j)}(x_{1:n})$. Efficient sampling from GP posterior $\mathcal{GP}(\hat{\mu}_n^{(j)}(\cdot), \hat{\sigma}_n^2(\cdot))$, as required here, has been widely explored in recent literature; see Wilson et al. [2020], Lin et al. [2023], Zhou [2025]. The pseudo code of our proposed ∞-GP-TS is shown in Algorithm 1.

Exploration and Exploitation in Distributional Space. As illustrated in Figure 1, Eq. (10) can be viewed as a structured trade-off between exploration and exploitation in the distributional space:

- **Exploitation:** With probability $P_n^{(j)} = \frac{\hat{n}_j}{\hat{\nu}+n}$, the new input x_{n+1} is assigned to an existing surface $\xi^{(j)}(\cdot)$. In this case, the model performs Kriging inference using observations previously associated with that surface. This corresponds to exploiting empirical knowledge, resulting in conservative updates confined to the empirical distribution. Notably, the assignment probability is proportional to the number of observations $\hat{n}_j$ already associated with surface j. The more observations are assigned to a surface, the more likely it will be reused.
- **Exploration:** With probability $P_n^{(0)} = \frac{\hat{\nu}}{\hat{\nu}+n}$, a new surface is instantiated from the base GP prior G_0. Since no observations have been made on this surface, it represents functional-level exploration—expanding the surrogate model by admitting novel structures beyond those already observed.

As the number of observations n increases, the exploration weight $P_n^{(0)}$ naturally diminishes, gradually **shifting focus from exploration toward exploitation** of well-supported surfaces. This hierarchical Bayesian mechanism thus equips Thompson Sampling with a principled way to adaptively balance exploration of new functional hypotheses and exploitation of learned structures, effectively broadening the surrogate model's support within the distribution space.

Algorithm 1 Thompson Sampling with ∞-GP Surrogate Model (∞-GP-TS)

Input: $\mathcal{H}_0$, the total number of iterations N.
for $n = 1$ to N **do**
Step 1: Sample $\{\hat{\Theta}, \hat{\boldsymbol{\xi}}_{1:n}\}$ from the posterior $f(\Theta, \boldsymbol{\xi}_{1:n} | \mathcal{H}_n)$ using Gibbs sampling (see the Appendix for details).
Step 2: Sample a surface $\xi^{(z_{n+1})}$ from (10).
Step 3: Evaluate the reward at the next candidate location $x_{n+1} = \arg\max_{x \in \mathcal{X}} x^\top \hat{\beta} + \hat{\xi}^{(z_{n+1})}(x)$. Observe the response $y(x_{n+1})$ and update $\mathcal{H}_{n+1} = \mathcal{H}_n \cup (x_{n+1}, y_{n+1})$.
Step 4: Continue to the next iteration $n + 1$.
end for

5 Theoretical Analysis: Efficient Exploration in the Distributional Space

In this section, we show that by exploring the space of reward distributions, the proposed ∞-GP model can approximate a wide range of true reward models and achieve a near-minimax-optimal rate. Since the noisy reward is a stochastic process over $\mathcal{X}$, it is standard to characterize it in terms of its finite-dimensional distributions. Specifically, for any finite set of input locations $\boldsymbol{x}_{1:D} = \{x_1, \cdots, x_D\} \subset \mathcal{X}$ with $D \in \mathbb{N}^+$, we denote the corresponding D-dimensional joint density of the true reward values by $f^*(\cdot \mid \boldsymbol{x}_{1:D})$. Let $\boldsymbol{k} = (k_1, \ldots, k_D) \in \mathbb{N}_0^D$ be a multi-index. Define the corresponding mixed partial derivative operator of a function $f(y_1, \cdots, y_D)$ as $f^{(\boldsymbol{k})} := \partial^{k_1+\cdots+k_D} f / \partial y_1^{k_1} \cdots \partial y_D^{k_D}$. For any $\alpha > 0$, $\lambda_0 \geq 0$, and any non-negative function $L : \mathbb{R}^D \to \mathbb{R}_{\geq 0}$, we define the locally Hölder class $\mathcal{C}^{\alpha,L,\lambda_0}(\mathbb{R}^D)$ as the set of all functions $f : \mathbb{R}^D \to \mathbb{R}$ that satisfy:(i) $f^{(\boldsymbol{k})}$ is finite for all multi-indices $\boldsymbol{k}$ such that $|\boldsymbol{k}| := \sum_{i=1}^D k_i \leq \lfloor \alpha \rfloor$; (ii) for every $x, y \in \mathbb{R}^D$, and $\boldsymbol{k}$ such that $|\boldsymbol{k}| = \lfloor \alpha \rfloor$, $\left| f^{(\boldsymbol{k})}(x+y) - f^{(\boldsymbol{k})}(x) \right| \leq L(x)\, e^{\lambda_0 \|y\|^2} \|y\|^{\alpha - \lfloor \alpha \rfloor}$.

Assumption 1 *The true density $f^*(\cdot|\boldsymbol{x}_{1:D}) \in \mathcal{C}^{\alpha,L,\lambda_0}(\mathbb{R}^D)$ for some $\alpha > 0$, $\lambda_0 \geq 0$ and a non-negative function L on $\mathbb{R}^D$ and satisfy $\mathbb{E}_{Y \sim f^*(\cdot|\boldsymbol{x}_{1:D})} \left(\frac{|f^{*(\boldsymbol{k})}(Y|x)|}{f^*(Y|x)} \right)^{\frac{2\alpha+\eta}{\sum_i^D k_i}} < \infty$ for any $\boldsymbol{k} \in \mathbb{N}_0^D$ satisfying $\sum_{i=1}^D k_i \leq \lfloor \alpha \rfloor$ and $\mathbb{E}_{Y \sim f^*(\cdot|\boldsymbol{x}_{1:D})} \left(\frac{L(Y)}{f^*(Y|\boldsymbol{x}_{1:D})} \right)^{\frac{2\alpha+\eta}{\alpha}} < \infty$ for some $\eta > 0$. Additionally, there are positive constant a_1, a_2, a_3, γ such that*

$$f^*(y|\boldsymbol{x}_{1:D}) \leq a_1 \exp(-a_2 \|y\|^\gamma), \ \|y\| > a_3. \tag{11}$$

Assumption 1 imposes mild regularity conditions on the true reward distribution. Apart from the tail condition in (11), the remaining smoothness assumptions are relatively weak. This assumption is satisfied by a broad class of distributions, including Weibull (with shape $k > 1$), Gaussian, Laplace, Gamma, and exponential distributions, as well as their finite mixtures of the form $\sum_{l=1}^L w_l(x)\, \psi_l(\theta_l)$, where each ψ_l belongs to one of the aforementioned distribution families.

Theorem 1 (Posterior Convergence of ∞-GP Model) *Let $f^*(y \mid \boldsymbol{x}_{1:D})$ be the true reward distribution satisfying Assumption 1, Π be the ∞-GP prior as defined in Eq.* (4)-(6)*, and let $\Pi_n(\cdot \mid \boldsymbol{x}_{1:D})$ denote the posterior distribution based on n i.i.d. evaluations at locations $\boldsymbol{x}_{1:D}$. Then there exists a sequence $\epsilon_n = n^{-\alpha/(2\alpha+D)} (\log n)^t$ for some constant $t > 0$, such that for any $M > 0$,*

$$\lim_{n \to \infty} \Pi_n \left(\{ f : \|f - f^*\|_1 > M\epsilon_n \} \,|\, \boldsymbol{x}_{1:D} \right) \to 0 \quad \textit{almost surely under } f^*.$$

This result demonstrates that by efficiently exploring the distribution space, the posterior distribution of the ∞-GP model can contract around a broad class of true reward distributions at a near-minimax-optimal rate in L_1 distance. In contrast to classical GP regression, which requires strictly stronger functional class assumptions [Kanagawa et al., 2018, Theorem 5.1] than our method and assumes Gaussian noise, our framework allows much broader modeling flexibility.

6 Empirical Evaluation

We evaluate our method on ten benchmark tasks, including six synthetic functions, three real-world problems, and one LLM prompt optimization task. We particularly focus on BO tasks characterized by **non-stationary**, complex reward landscapes and **heavy-tailed** observation noise. Extensive results and ablations are provided in the Appendix.

Synthetic Benchmarks. We consider three popular synthetic test functions: Ackley, Rosenbrock, and StybTang [Xu et al., 2024]. To introduce more challenging scenarios, we consider non-stationary and heavy-tailed variants of these functions. In the heavy-tailed (HT) cases, all the functions are corrupted by Weibull-distributed noises. In the non-stationary (NS) setting, the base test functions are modulated by a trigonometric-exponential term of the form $f_{\text{NS}}(x) = (1 + \alpha \sin(x) e^x) \cdot f(x)$, which introduces non-stationarity across the domain.

Real-world Benchmarks. We consider three real-world benchmarks.

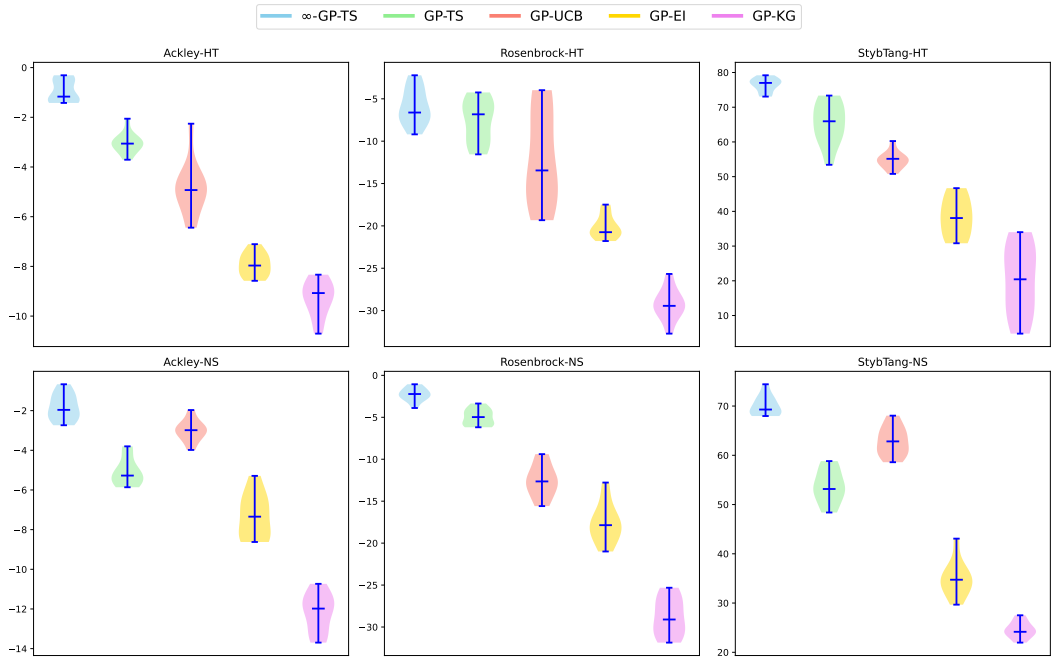

Figure 2: Results of synthetic benchmarks. The suffix HT stands for the function with heavy-tailed noises and NS stands for the non-stationary scenarios.

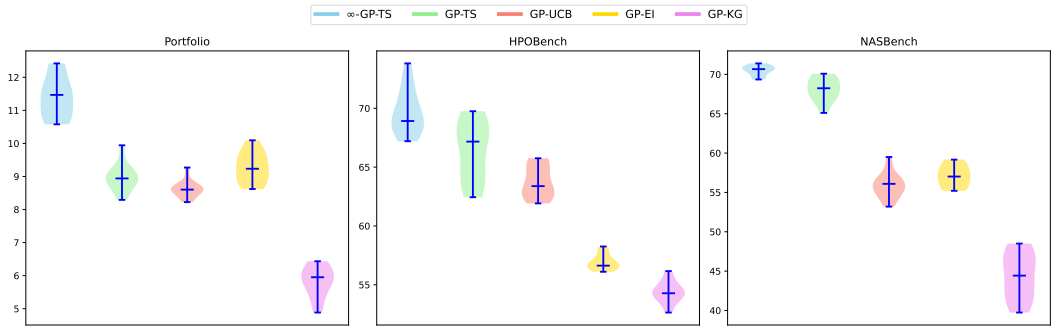

Figure 3: Results of real-world benchmarks.

- **Portfolio** [Cakmak et al., 2020] is a benchmark of tuning the hyperparameters of a trading strategy to maximize returns. The stock data with heavy-tailed nature is generated from a realistic financial simulator, CVXPortfolio.
- **HPOBench** [Eggensperger et al., 2021] provides standard hyperparameter tuning tasks for ML models, and we use the MLP task in the HPOBench in our experiment.
- **NASBench201** [Dong and Yang, 2020] is a neural network search task on CIFAR100. The search space of hyperparameters and model architecture is considered non-stationary. Therefore, we use these tasks for our evaluation of real-world non-stationary cases.

Prompt optimization for language understanding. We conduct prefix prompt optimization for seven language understanding datasets, including sentiment classification (SST-2 [Socher et al., 2013]), SST-5 [Socher et al., 2013], MR [Pang and Lee, 2005], CR [Hu and Liu, 2004]), topic classification (AG's News[Zhang et al., 2015], TREC[Voorhees and Tice, 2000]) and subjectivity classification (Subj [Pang and Lee, 2004]). We use the Alpaca-7B model [Taori et al., 2023] as the frozen language model, and optimize continuous prefix embeddings to guide the model toward better task performance. Due to the high dimensions of the decision variable, we use Uniform Manifold Approximation and Projection (UMAP) [McInnes et al., 2018] to reduce the dimension.

Method	SST-2	CR	MR	SST-5	AG's News	TREC	Subj	Avg.
GP-TS	93.87	91.44	89.85	49.91	72.81	45.29	65.18	72.61
GP-UCB	92.68	90.20	89.98	49.86	71.76	43.27	62.70	71.49
GP-EI	91.82	89.75	87.69	48.26	60.83	42.09	59.70	68.59
GP-KG	90.05	88.38	85.90	42.89	45.98	36.20	49.75	62.74
∞-GP-TS	**96.57**	**92.52**	**91.51**	**52.58**	**75.34**	**46.23**	**68.55**	**74.76**

Table 1: Performance comparison on seven language understanding datasets.

Baselines. We compare our methods with the following baselines based on GP surrogate model. **GP-TS** [Chowdhury and Gopalan, 2017]: Following the TS policy defined in Eq. (7), except that the posterior is constructed under a standard GP model for $\mu^*(x)$. **GP-UCB** [Srinivas et al., 2009]: Selects the location $x_{n+1} = \arg\max_{x\in\mathcal{X}} \mu_n(x) + \beta_n^{1/2}\sigma_n(x)$, where β_n is a time-dependent exploration parameter. **GP-EI** [Ament et al., 2023]: Selects $x_{n+1} = \arg\max_{x\in\mathcal{X}} \mathbb{E}\left[\max\{0, \mu^*(x) - \mu^{\text{best}}\}\right]$, where the expectation is taken under the posterior distribution of $\mu^*(x)$. **GP-KG** [Wu and Frazier, 2016]: Selects the point whose evaluation is expected to most improve the posterior estimate of the maximum value of $\mu^*(x)$. **Non-stationary GP-UCB/TS/EI/KG**: In non-stationary tasks and the Prompt optimization task, all baseline methods adopt input warping techniques in Snoek et al. [2014] to mitigate GP misspecification. **Truncated-GP-UCB**: In heavy-tailed noise tasks, GP-UCB adopts the truncation technique in Chowdhury and Gopalan [2019].

Performance. We report the final optimization outcome of the synthetic and real-world experiment in Figure 2 and 3, respectively (runtime regret results are provided in the Appendix). The result of each task is reported from 10 repetitive runs with different seeds. In the synthetic experiment, ∞-GP consistently outperforms the baselines in heavy-tailed and non-stationary settings. The GP-TS and GP-UCB have similar performance against each other. The GP-EI performs poorly in both scenarios. The GP-KG has the worst results, and the final performance fluctuates heavily.

In the portfolio task, we report the returns of the strategy. Stock prices often exhibit sharp fluctuations, making portfolio optimization a suitable testbed for evaluating performance under heavy-tailed environments. Our method achieves the highest return among others, while the GP-EI has the second-best return. In the HPOBench and the NASBench, the performance gap between our method and baselines is close, but our method has a smaller fluctuation with different seeds, showing better robustness than others.

The language model in the prompt optimization task is Alphaca-7b. Table 1 reports the average final accuracy across three independent runs. The proposed ∞-GP outperforms all GP-based baselines on average, achieving the highest mean accuracy (74.76), and ranking first on every dataset. Compared to GP-TS and GP-UCB, ∞-GP offers notable gains on complex tasks such as SST-5 and Subj, where the reward landscape is highly non-stationary. GP-EI and GP-KG perform significantly worse, consistent with their known sensitivity to noisy or irregular reward signals. These results demonstrate the robustness of ∞-GP in prompt optimization tasks under distributional uncertainty.

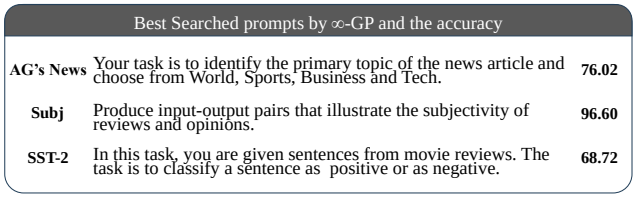

Figure 4: Qualitative examples of best searched prompts

7 Discussion

This work introduces the ∞-GP model and its integration with TS, providing a principled mechanism to automatically balance exploration and exploitation in the space of reward distributions. Empirically, the proposed method demonstrates strong performance in a wide range of challenging settings where classical GP-based BO methods often fail. Nonetheless, we do not systematically evaluate the performance of ∞-GP in high-dimensional BO tasks. That said, as shown in Eq. (10) and Algorithm 1, our approach can be interpreted as a finite mixture of traditional GP-TS. This structure is orthogonal to many recent advances in high-dimensional GP-based BO, and hence, those techniques can be readily incorporated into our framework. We leave the exploration of such extensions to future work.

References

Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Felix Berkenkamp, Andreas Krause, and Angela P Schoellig. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, 112(10):3713–3747, 2023.

Alan Nawzad Amin, Nate Gruver, Yilun Kuang, Lily Li, Hunter Elliott, Calvin McCarter, Aniruddh Raghu, Peyton Greenside, and Andrew Gordon Wilson. Bayesian optimization of antibodies informed by a generative model of evolving sequences. *arXiv preprint arXiv:2412.07763*, 2024.

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 preprint arXiv:1012.2599*, 2010.

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

Jingyi Wang, Haowei Wang, Cosmin G Petra, and Nai-Yuan Chiang. On the convergence of noisy bayesian optimization with expected improvement. *arXiv preprint arXiv:2501.09262*, 2025.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1015–1022. Omnipress, 2010.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE transactions on information theory*, 58(5):3250–3265, 2012.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.

Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

Haoxian Chen and Henry Lam. Pseudo-bayesian optimization. *arXiv preprint arXiv:2310.09766*, 2023.

Jasper Snoek, Kevin Swersky, Rich Zemel, and Ryan Adams. Input warping for bayesian optimization of non-stationary functions. In *International conference on machine learning*, pages 1674–1682. PMLR, 2014.

Sayak Ray Chowdhury and Aditya Gopalan. Bayesian optimization under heavy-tailed payoffs. *Advances in Neural Information Processing Systems*, 32, 2019.

Sait Cakmak, Raul Astudillo Marban, Peter Frazier, and Enlu Zhou. Bayesian optimization of risk measures. *Advances in Neural Information Processing Systems*, 33:20130–20141, 2020.

Antonio Sabbatella, Andrea Ponti, Ilaria Giordani, Antonio Candelieri, and Francesco Archetti. Prompt optimization in large language models. *Mathematics*, 12(6):929, 2024.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P Xing, and Zhiting Hu. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*, 2022.

Yucen Lily Li, Tim GJ Rudner, and Andrew Gordon Wilson. A study of bayesian neural network surrogates for bayesian optimization. *arXiv preprint arXiv:2305.20028*, 2023.

Dave Higdon, Jenise Swall, and John Kern. Non-stationary spatial modeling. *arXiv preprint arXiv:2212.08043*, 2022.

Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.

Sébastien Bubeck, Nicolo Cesa-Bianchi, and Gábor Lugosi. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717, 2013.

Andres Munoz Medina and Scott Yang. No-regret algorithms for heavy-tailed linear bandits. In *International Conference on Machine Learning*, pages 1642–1650. PMLR, 2016.

George Wynne, François-Xavier Briol, and Mark Girolami. Convergence guarantees for gaussian process means with misspecified likelihoods and smoothness. *Journal of Machine Learning Research*, 22(123):1–40, 2021.

Ilija Bogunovic and Andreas Krause. Misspecified gaussian process bandit optimization. *Advances in neural information processing systems*, 34:3004–3015, 2021.

Carl Rasmussen and Zoubin Ghahramani. Infinite mixtures of gaussian process experts. *Advances in neural information processing systems*, 14, 2001.

Edward Meeds and Simon Osindero. An alternative infinite mixture of gaussian process experts. *Advances in neural information processing systems*, 18, 2005.

Nicola Bariletto and Nhat Ho. Bayesian nonparametrics meets data-driven distributionally robust optimization. *arXiv preprint arXiv:2401.15771*, 2024.

Alan E Gelfand, Athanasios Kottas, and Steven N MacEachern. Bayesian nonparametric spatial modeling with dirichlet process mixing. *Journal of the American Statistical Association*, 100(471): 1021–1035, 2005.

Fernando A Quintana, Peter Müller, Alejandro Jara, and Steven N MacEachern. The dependent dirichlet process and related models. *Statistical Science*, 37(1):24–41, 2022.

James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Efficiently sampling functions from gaussian process posteriors. In *International Conference on Machine Learning*, pages 10292–10302. PMLR, 2020.

Jihao Andreas Lin, Javier Antorán, Shreyas Padhy, David Janz, José Miguel Hernández-Lobato, and Alexander Terenin. Sampling from gaussian process posteriors using stochastic gradient descent. *Advances in Neural Information Processing Systems*, 36:36886–36912, 2023.

Zhihao Zhou. Accelerating posterior sampling for scalable gaussian process model. *arXiv preprint arXiv:2505.02000*, 2025.

Zhitong Xu, Haitao Wang, Jeff M Phillips, and Shandian Zhe. Standard gaussian process is all you need for high-dimensional bayesian optimization. In *The Thirteenth International Conference on Learning Representations*, 2024.

Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, René Sass, Aaron Klein, Noor Awad, Marius Lindauer, and Frank Hutter. Hpobench: A collection of reproducible multi-fidelity benchmark problems for hpo. *arXiv preprint arXiv:2109.06716*, 2021.

Xuanyi Dong and Yi Yang. Nas-bench-201: Extending the scope of reproducible neural architecture search. *arXiv preprint arXiv:2001.00326*, 2020.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*, 2005.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Ellen M Voorhees and Dawn M Tice. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207, 2000.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*, 2004.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for bayesian optimization. *Advances in Neural Information Processing Systems*, 36:20577–20612, 2023.

Jian Wu and Peter Frazier. The parallel knowledge gradient method for batch bayesian optimization. *Advances in neural information processing systems*, 29, 2016.

Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

David Blackwell and James B MacQueen. Ferguson distributions via pólya urn schemes. *The annals of statistics*, 1(2):353–355, 1973.

Subhashis Ghosal and Aad van der Vaart. Posterior convergence rates of dirichlet mixtures at smooth densities. *The Annals of Statistics*, 35(2):697–723, 2007.

Weining Shen, Surya T Tokdar, and Subhashis Ghosal. Adaptive bayesian multivariate density estimation with dirichlet mixtures. *Biometrika*, 100(3):623–640, 2013.

A Proofs

A.1 Proof of Eq. (8) and Eq. (10)

We adopt the bracket notation system used in Gelfand and Smith [1990], where $[Y \mid X]$ denotes the conditional density of Y given X, and $[Y]$ denotes the marginal density of Y.

The predictive reward distribution for any unexplored solution x_{n+1} is given by

$$
\begin{aligned}
[y(x_{n+1})|\mathcal{H}_n] &= \int \underbrace{[y(x_{n+1})|\Theta^{(1)}, \xi^{(z_{n+1})}(x_{n+1})]}_{A\text{: Gaussian}} \underbrace{[\Theta^{(1)}, \xi^{(z_{n+1})}(x_{n+1})|\mathcal{H}_n]}_{B+C} \\
&= \int\int \underbrace{[y(x_{n+1})|\Theta^{(1)}, \xi^{(z_{n+1})}(x_{n+1})]}_{A\text{: Gaussian}} \underbrace{[\xi^{(z_{n+1})}(x_{n+1})|\boldsymbol{\xi}_{1:n}(x_{1:n}), \boldsymbol{z}_{1:n}, \Theta^{(2)}]}_{B} \\
&\quad \underbrace{[\Theta, \boldsymbol{\xi_{1:n}}(x_{1:n}), \boldsymbol{z}_{1:n}|\mathcal{H}_n]}_{C}
\end{aligned} \tag{12}
$$

Notice that, when performing computations involving stochastic processes over $\mathcal{X}$, in practice, we are working with their finite-dimensional distributions, i.e., the joint distribution over a finite subset of $\mathcal{X}$. Therefore, the term $\boldsymbol{\xi}_{1:n}$, becomes $\boldsymbol{\xi}_{1:n}(x_{1:n}) = \{\xi^{(j)}(x_i)\}_{i=1,\cdots,n\ j=1,\cdots,K_n}$, which represents the $K_n \times n$ table of values on the realized surfaces corresponding to the observed locations $x_{1:n}$.

In the first line of Eq. (12), conditioned on the new surface $\xi^{(z_{n+1})}(x_{n+1})$ and first-layer hyper parameter $\Theta^{(1)} = (\beta, \tau^2)$, the distribution $[y(x_{n+1})|\Theta^{(1)}, \xi^{(z_{n+1})}(x_{n+1})]$ is $\mathcal{N}(x_{n+1}^\top \beta + \xi^{(z_{n+1})}(x_{n+1}), \tau^2)$, according to Eq. (4). The remaining Term "B+C" is $[\Theta'|\mathcal{H}_n]$. In the second line of Eq. (12), the posterior distribution $[\Theta^{(1)}, \xi^{(z_{n+1})}|\mathcal{H}_n]$, i.e. the Term "B+C", is decomposed into Term B and Term C. Term C represents the posterior distribution of the hyperparameters and realized surfaces, which requires inference through Gibbs sampling. The proof of Eq. (8) is now complete.

We next proceed to establish Eq. (10). The Term B represents the prediction of the new surface $\xi^{(z_{n+1})}$ based on the previously realized surfaces $\boldsymbol{\xi}_{1:n} = (\xi^{(1)}, \ldots, \xi^{(K_n)})$ and their associated assignment labels $\boldsymbol{z}_{1:n}$. Conditioned on $\boldsymbol{\xi}_{1:n} = (\xi^{(1)}, \ldots, \xi^{(K_n)})$ and $\boldsymbol{z}_{1:n}$, and marginalizing out the random measure $\{G_x : x \in \mathcal{X}\}$, the distribution of the latent surface index z_{n+1} follows the urn scheme of the Dirichlet process [Blackwell and MacQueen, 1973]. Specifically, the new point either reuses one of the existing surfaces (solid lines in Figure 1) or initiates a new surface drawn from the base measure $G_0 = \mathcal{GP}(0, \sigma^2 \rho_{\boldsymbol{\phi}}(\cdot, \cdot))$ (dashed lines in Figure 1). Formally:

$$
\begin{aligned}
&\underbrace{[\xi^{(z_{n+1})} \mid \boldsymbol{\xi}_{1:n}, \boldsymbol{z}_{1:n}, \Theta^{(2)}]}_{\text{Urn scheme}} \\
&= \int [\xi^{(z_{n+1})} \mid \boldsymbol{\xi}_{1:n}, \boldsymbol{z}_{1:n}, \Theta^{(2)}, G]\,[G] \\
&\sim \frac{\nu}{\nu+n} G_0 + \sum_{j=1}^{K_n} \frac{n_j}{\nu+n} \delta_{\xi^{(j)}},
\end{aligned} \tag{13}
$$

where n_j denotes the number of data points currently assigned to surface $\xi^{(j)}$. The more points are assigned to a surface, the more likely it is to be reused. Then, after determining which surface $\xi^{(z_{n+1})}$ the new location x_{n+1} will be realized on, we perform Kriging on that surface to predict $\xi^{(z_{n+1})}(x_{n+1})$ using previously realized values $\boldsymbol{\xi}_{1:n}(x_{1:n})$. Therefore, the Term B can be decomposed as

$$
\begin{aligned}
&[\xi^{(z_{n+1})}(x_{n+1})|\boldsymbol{\xi}_{1:n}(x_{1:n}), \boldsymbol{z}_{1:n}, \Theta^{(2)}] \\
&= \int \underbrace{[\xi^{(z_{n+1})}(x_{n+1})|\boldsymbol{\xi}_{1:n}(x_{n+1}), \boldsymbol{z}_{1:n}, \Theta^{(2)}]}_{\text{Step (a): Urn scheme (13)}} \underbrace{[\boldsymbol{\xi}_{1:n}(x_{n+1})|\boldsymbol{\xi}_{1:n}(x_{1:n}), \Theta^{(2)}]}_{\text{Step (b): kriging}},
\end{aligned} \tag{14}
$$

where the urn scheme term follows (13) by evaluating $\xi^{(z_{n+1})}$ and $\boldsymbol{\xi}_{1:n}$ at location x_{n+1}. The Kriging term is given by (recall that $\boldsymbol{\xi}_{1:n} = \{\xi^{(j)}\}_{j=1}^{K_n}$)

$$
\underbrace{[\xi^{(j)}(x_{n+1})|\boldsymbol{\xi}_{1:n}(x_{1:n}), \Theta^{(2)}]}_{\text{kriging}} \sim \mathcal{GP}(\mu_n^{(j)}(x_{n+1}), \sigma_n^2(x_{n+1})), \tag{15}
$$

where $\mu_n^{(j)}(x_{n+1}) = \Sigma_0(x_{n+1}, x_{1:n})\Sigma_0(x_{1:n}, x_{1:n})^{-1}\xi^{(j)}(x_{1:n})$, and $\sigma_n^2(x_{n+1})$ is given by Eq. (3). Therefore, combing Eq. (14) and Eq. 15, we have

$$f\big(\xi^{(z_{n+1})}(\cdot) \mid \hat{\Theta}^{(2)}, \hat{\boldsymbol{\xi}}_{1:n}, \hat{\boldsymbol{z}}_{1:n}\big) \sim \underbrace{P_n^{(0)} G_0(\cdot)}_{\text{Exploration}} + \underbrace{\sum_{j=1}^{\hat{K}_n} P_n^{(j)} \overbrace{\mathcal{GP}(\hat{\mu}_n^{(j)}(\cdot), \hat{\sigma}_n^2(\cdot))}^{\text{Kriging on } \xi^{(j)}},}_{\text{Exploitation}} \tag{16}$$

which concludes the proof.

A.2 Proof of Theorem 1

For any finite set of input locations $\boldsymbol{x}_{1:D} = \{x_1, \cdots, x_D\} \subset \mathcal{X}$ with $D \in \mathbb{N}^+$, we consider the corresponding D-dimensional joint density of the true reward values by $f^*(\cdot \mid \boldsymbol{x}_{1:D})$.

Notice that, focusing on fixed $x_{1:D}$, the ∞-GP model can be viewed as a DP location mixture of normals prior Π, which is the distribution of a random distribution $f_{DP_G,\Sigma} = \int \phi_\Sigma(x - z) DP_G(dz)$. ϕ_Σ is the normal density with mean zero and covariance Σ. DP_G is a Dirichlet process with baseline distribution G, which is a D-dimension Gaussian distribution, with squared exponential kernel $\rho_{\boldsymbol{\phi}}(x, x') = \exp\left(-\sum_{k=1}^d \phi_k (x^{(k)} - x'^{(k)})^2\right)$ (i.e. the kernel of baseline of ∞-GP) and zero mean. In our model, we assume the covariance matrix $\Sigma \in \mathbb{R}^{D\times D}$ to be diagonal with independent inverse-gamma priors on each variance component:

$$\Sigma = \begin{bmatrix} \tau_1^2 & 0 & \cdots & 0 \\ 0 & \tau_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tau_D^2 \end{bmatrix},$$

with $\tau_i^2 \sim$ Inv-Gamma(a_τ, b_τ) independently for $i = 1, \ldots, D$.

Let $|G| = G(\mathbb{R}^D)$ and $\bar{G} = G/|G|$. Let eig_j be the j-th eigenvalue. We first present the following lemma.

A.2.1 Lemma 1

Lemma 1 *There exist positive constants a_1-a_8.*

$$1 - \bar{G}([-x, x]^D) \le a_1 \exp(-a_2 x^{a_3}), \tag{17}$$

and

$$\mathbb{P}(\Sigma : \textit{eig}_D(\Sigma^{-1}) \ge x) \le a_4 \exp(-a_5 x^{a_6}). \tag{18}$$

for sufficiently large $x > 0$, and

$$\mathbb{P}(\Sigma : \textit{eig}_1(\Sigma^{-1}) < x) \le a_7 X^{a_8} \tag{19}$$

for sufficiently small $x > 0$. Moreover, there exist a_9 such that for any $0 < s_1 \le \cdots \le s_D$ and $t \in (0, 1)$,

$$\mathbb{P}(\Sigma : s_j < \textit{eig}_j(\Sigma^{-1}) < s_j(1 + t), j = 1, \cdots, D) \ge a_9 s_1^{a_{10}} t^{a_{11}} \exp(-a_{12} s_d). \tag{20}$$

Proof: Equation (17) follows from the exponential tail behavior of the Gaussian base measure G. To prove Equation (18), recall that $\Sigma = \text{diag}(\tau_1^2, \ldots, \tau_D^2)$ and hence $\text{eig}_D(\Sigma^{-1}) = \max_i \left\{\frac{1}{\tau_i^2}\right\}$. Since $\tau_i^2 \sim$ Inv-Gamma(a_τ, b_τ), we use the left-tail bound for inverse-gamma random variables:

$$\mathbb{P}\left(\tau_i^2 \le \tfrac{1}{x}\right) \le C_1 \exp(-C_2 x), \quad \text{as } x \to \infty.$$

Applying a union bound over $i = 1, \ldots, D$, we obtain:

$$\mathbb{P}\left(\text{eig}_D(\Sigma^{-1}) \ge x\right) \le D \cdot C_1 \exp(-C_2 x) = a_4 \exp(-a_5 x^{a_6}).$$

For Equation (19), we use the right-tail of the inverse-gamma distribution:

$$\mathbb{P}\left(\tau_i^2 \geq \tfrac{1}{x}\right) = \mathbb{P}\left(\frac{1}{\tau_i^2} \leq x\right) \leq C_3 x^{a_\tau}, \quad \text{as } x \to 0^+.$$

Therefore,

$$\mathbb{P}\left(\text{eig}_1(\Sigma^{-1}) < x\right) = \mathbb{P}\left(\frac{1}{\tau_i^2} < x \text{ for some } i\right) \leq D \cdot C_3 x^{a_\tau} = a_7 x^{a_8}.$$

Finally, for the lower bound in Equation (20), note that due to the independence of τ_i^2, the eigenvalues of Σ^{-1} are also independent. Thus, for each j, we consider the event:

$$\text{eig}_j(\Sigma^{-1}) \in [s_j, s_j(1+t)],$$

which translates to $\tau_j^2 \in \left[\frac{1}{s_j(1+t)}, \frac{1}{s_j}\right]$.

The inverse-gamma distribution has a continuous, strictly positive density on compact intervals bounded away from zero. Therefore, for each j,

$$\mathbb{P}\left(\frac{1}{\tau_j^2} \in [s_j, s_j(1+t)]\right) \geq c_j s_j^{-b} t,$$

for some constants $c_j, b > 0$. Multiplying across $j = 1, \ldots, D$, we get:

$$\mathbb{P}\left(\text{eig}_j(\Sigma^{-1}) \in [s_j, s_j(1+t)]\ \forall j\right) \geq a_9 s_1^{a_{10}} t^{a_{11}} \exp(-a_{12} s_D),$$

where we used the fact that inverse-gamma densities decay exponentially on the left (i.e., for small τ_j^2), which leads to the $\exp(-a_{12} s_D)$ term.

A.2.2 Lemma 2

Lemma 2 *[Ghosal and van der Vaart [2007]]*

$$\lim_{n\to\infty} \Pi_n\left(\{f : \|f - f^*\|_1 > M\epsilon_n\} \,|\, \boldsymbol{x}_{1:D}\right) \to 0 \quad \textit{almost surely under } f^*$$

whenever there exist positive constants c_1, c_2, c_3, c_4, a sequence of positive numbers $(\tilde{\epsilon}_n)_{n\geq 1}$ with $\tilde{\epsilon}_n \leq \epsilon_n$ and $\lim_{n\to\infty} n\tilde{\epsilon}_n^2 = \infty$, and a sequence of compact subsets $(\mathcal{F}_n)_{n\geq 1}$ of probability densities satisfying

$$\log N(\epsilon_n, \mathcal{F}_n, \rho) \leq c_1 n\epsilon_n^2, \tag{21}$$

$$\Pi(\mathcal{F}_n^c) \leq c_3 e^{-(c_2+4)n\tilde{\epsilon}_n^2}, \tag{22}$$

$$\Pi\{f : \mathcal{K}(f^*, \tilde{\epsilon}_n) \leq \tilde{\epsilon}_n^2, \} \geq c_4 e^{-c_2 n\tilde{\epsilon}_n^2}. \tag{23}$$

then the posterior contracts at the rate ϵ_n around f_0 in the metric ρ.

Here, $\mathcal{K}(f^*, \epsilon)$ is the Kullback–Leibler ball around f^* of size $\tilde{\epsilon}_n$, defined as

$$\mathcal{K}(f^*, \epsilon) = \left\{f : \int f^* \log\left(\frac{f^*}{f}\right) < \epsilon^2,\ \int f^* \log^2\left(\frac{f^*}{f}\right) < \epsilon^2\right\}.$$

$N(\epsilon_n, \mathcal{F}_n, \rho)$ is the ϵ_n-covering number of $\mathcal{F}_n$ under metric ρ.

A.2.3 Proof of Theorem 1

Now we prove Theorem 1 by verifying conditions (21), (22) and (23), respectively. We construct a sieve

$$\mathcal{Q}(a, M, J, \epsilon) \triangleq \left\{ f_{DP_G,\Sigma} : \begin{array}{l} DP_G = \sum_{j=1}^{\infty} w_j \delta_{\xi_j(x_{1:D})},\ \xi_j(x_{1:D}) \in [-a, a]^D \text{ for } j < J; \\ \sum_{j>J} w_j < \epsilon, \sigma_0^2 \leq \text{eig}_j(\Sigma) < \sigma_0^2\left(1 + \frac{\sigma^2}{D}\right)^M,\ \forall j = 1, \ldots, D \end{array} \right\}.$$

Then, according to [Shen et al., 2013, Theorem 5] and Lemma 1, conditions (21) and (22) hold for $\epsilon_n = n^{-\gamma'}(\log n)^t$, $\tilde{\epsilon}_n = n^{-\gamma'}(\log n)^{t_0}$, $J = \lfloor \frac{n\epsilon_n^2}{\log n} \rfloor$, $M = a^{a_2}$, when fixing $\gamma' \in (0, 1/2)$ and $t > t_0 \geq \frac{D+1}{2}$.

To verify condition (23), we follow the prior thickness argument in Shen et al. [2013, Theorem 4]. Under Assumption 1, the true density f^* belongs to a locally Hölder smooth class with exponential tail decay. Theorem 4 guarantees that there exists a discrete mixture density $p_{F,\Sigma}$ such that both the Kullback–Leibler divergence and its second moment between f^* and $p_{F,\Sigma}$ are bounded by $C\tilde{\epsilon}_n^2$ for some constant $C > 0$. Moreover, the Dirichlet process prior assigns exponentially non-negligible mass to such mixtures via a partitioning and stick-breaking construction. The prior on the covariance matrix Σ, as discussed in Lemma 1, also places sufficient mass on the eigenvalue shell corresponding to the mixture approximation. Combining these results, let $\tilde{\epsilon}_n^2 = n^{-\alpha/(2\alpha+D)}(\log n)^t$, for any $t \geq \frac{D(1+1/\gamma+1/\alpha)+1}{2+D+\alpha}$, according to Shen et al. [2013, Theorem 4], we obtain

$$\Pi\left\{f : \mathrm{KL}(f^*, f) \leq \tilde{\epsilon}_n^2,\ \mathrm{Var}_{f^*} \log \frac{f^*}{f} \leq \tilde{\epsilon}_n^2\right\} \geq c_4 e^{-c_2 n \tilde{\epsilon}_n^2},$$

for some constants $c_2, c_4 > 0$, thus verifying condition (23). Finally, according to Lemma (2), the proof is concluded.

B Gibbs Sampling Algorithm

B.1 Prior Specification

We adopt a fully-Bayesian treatment for hyperparameters $\Theta = \{\beta, \nu, \tau, \sigma^2, \phi\} = \{\Theta^{(1)}, \Theta^{(2)}\}$, where the fist-layer parameters $\Theta^{(1)} = \{\beta, \tau\}$ and the second-layer parameters $\Theta^{(2)} = \{\nu, \sigma^2, \phi\}$. Specifically, the priors on hyperparameters Θ are given by

$$\beta, \tau^2 \sim N_p(\beta_0, \Sigma_\beta) \times IGamma(a_\tau, b_\tau), \tag{24}$$

$$\sigma^2 \sim IGamma(a_\sigma, b_\sigma), \tag{25}$$

$$\boldsymbol{\phi} \sim U([0, b_\phi]^d), \tag{26}$$

$$\nu \sim Gamma(a_\nu, b_\nu), \tag{27}$$

where β has a Gaussian prior, τ^2 and σ^2 have inverse Gamma prior, ϕ has a uniform prior on $(0, b_\phi]$ and ν has a Gamma prior.

Selection of Hyperparameters in the Prior we set $a_\tau = a_\sigma = 2$ and thus the mean of the prior is b_τ and b_σ. Therefore, prior information about the variance (for example, a rough estimation of the variance and the mean) can be incorporated.

ϕ: ϕ has a uniform prior on $(0, b_\phi]$. If the distance between two locations x_1 and x_2, $\|x_1 - x_2\| > \frac{3}{\phi}$, their correlation will decrease to less than 0.05. Therefore, we set $\frac{3}{b_\phi} = d \max\|x_1 - x_2\|$ with a small d (e.g. 0.01).

β_0 and Σ_β: we set $\beta_0 = [1, \cdots, 1] \in \mathbb{R}^d$ and set $\Sigma_\beta = I_{d\times d}$.

B.2 Gibbs Sampling

To perform posterior inference under our fully-Bayesian model, we employ the Gibbs sampling algorithm. Gibbs sampling is a Markov Chain Monte Carlo (MCMC) method designed to sample from a complex joint posterior distribution by iteratively sampling from each of its full conditional distributions. This approach is particularly well-suited for our hierarchical ∞-GP model, where the full conditionals of most parameters have closed-form expressions or can be easily computed. To ensure computational tractability, we adopt a truncation approximation to the stick-breaking representation of the Dirichlet process. Specifically, we fix the number of mixture components to a finite value L.

Algorithm 2 Gibbs Sampling for $\{\Theta, \boldsymbol{\xi_{1:n}}(x_{1:n})\}$

Input: Initialization value of $\{\Theta, \boldsymbol{\xi_{1:n}}(x_{1:n})\}$, data $\mathcal{H}_n$, MCMC iteration number B, truncation number L

Output: Samples of $[\Theta, \boldsymbol{\xi_{1:n}}(x_{1:n})|\mathcal{H}_n]$, including $\boldsymbol{\xi_{1:n}}(x_{1:n})$, $(w_1, \ldots, w_L)$, $(z_1, \ldots, z_n)$, and $\Theta = \{\beta, \nu, \tau, \sigma^2, \phi\}$.

for each iteration $b = 1, \ldots, B$ **do**

Step 1: Sampling $\boldsymbol{\xi_{1:n}}(x_{1:n})$:

for each $l = 1, \ldots, L$ **do**

Compute $\mathcal{I}^l(\boldsymbol{z}_{1:n}) = \text{diag}(\mathbb{I}_{\{z_1=l\}}, \ldots, \mathbb{I}_{\{z_n=l\}})$

Sample $\xi_l(x_{1:n})$ from the distribution $\xi_l(x_{1:n}) \sim \mathcal{N}\left(\frac{1}{\tau^2}\Lambda^l\mathcal{I}^l\left(y(x_{1:n}) - x_{1:n}^\top\beta\right), \Lambda^l\right)$, where $\Lambda^l = \left(\Sigma_0^{-1} + \tau^{-2}\mathcal{I}^l\right)^{-1}$ and $\Sigma_0 = \sigma^2 H(\phi)$

end for

Step 2: Sampling Weights $(w_1, \ldots, w_L)$**:**

for each $l = 1, \ldots, L$ **do**

Draw $V_l \sim \text{Beta}(1 + M_l, \nu + \sum_{j=l+1}^{L} M_j)$, where $M_j = \#\{i : z_i = j\}$

Compute weights by $w_1 = V_1,\ w_l = (1 - V_1)(1 - V_2)\ldots(1 - V_{l-1})V_l,\ w_L = 1 - \sum_{l=1}^{L-1} w_l$

end for

Step 3: Sampling Latent Labels $(z_1, \ldots, z_n)$**:**

for each observation x_i **do**

Sample z_i from $\{1, \ldots, L\}$ with probabilities proportional to

$$P(z_i = j) \propto w_j \exp\left\{-\frac{1}{2\tau^2}\left(y(x_i) - x_i^\top\beta - \xi_j(x_i)\right)^2\right\}.$$

end for

Step 4: Sampling Hyper-Parameters $\Theta = \{\beta, \nu, \tau, \sigma^2, \phi\}$**:**

(4a) Sampling ν**:** $\nu \sim \text{Gamma}(a_\nu + n - 1, b_\nu - \log(w_L))$

(4b) Sampling β**:**

$$\beta \sim \mathcal{N}(\beta_0^n, \Sigma_\beta^n)$$

where $\Sigma_\beta^n = \left(\Sigma_\beta^{-1} + \tau^{-2}x_{1:n}^\top x_{1:n}\right)^{-1}$, $\beta_0^n = \Sigma_\beta^n\left(\Sigma_\beta^{-1}\beta_0 + \tau^{-2}X_{1:n}^\top\bar{y}(x_{1:n})\right)$ and $\bar{y}(x_i) = y(x_i) - \xi^{(z_i)}(x_i)$.

(4c) Sampling τ^2**:**

$$\tau^2 \sim \text{InvGamma}(a_\tau^n, b_\tau^n)$$

where $a_\tau^n = a_\tau + \frac{n}{2}, \quad b_\tau^n = b_\tau + \frac{1}{2}\sum_{i=1}^{n}\left(\bar{y}(x_i) - x_i^\top\beta\right)^2$.

(4d) Sampling σ^2**:**

$$\sigma^2 \sim \text{IGamma}(a_\sigma^n, b_\sigma^n)$$

where $a_\sigma^n = a_\sigma + \frac{nL}{2}, \quad b_\sigma^n = b_\sigma + \frac{1}{2}\sum_{l=1}^{L}\left(\xi_l(x_{1:n})\right)^\top\rho_{\boldsymbol{\phi}}^{-1}(x_{1:n}, x_{1:n})\xi_l(x_{1:n})$.

(4e) Sampling ϕ**:** ϕ is sampled from a grid of values $\{\phi_1, \phi_2, \ldots, \phi_M\}$ with probabilities proportional to

$$P(\phi_m) \propto \frac{1}{[\det(\rho_{\phi_m}(x_{1:n}, x_{1:n}))]^{L/2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{l=1}^{L}\left(\xi_l(x_{1:n})\right)^\top\rho_{\phi_m}^{-1}(x_{1:n}, x_{1:n})\xi_l(x_{1:n})\right).$$

end for

C Experimental Details, More Experiment Results and Ablations

C.1 More Experiment Results

In this subsection, we provide additional plots that visualize the evolution of instant regret over time for all benchmark tasks, where instant regret is defined as the gap between the reward obtained and the global optimum. These plots offer a clearer comparison of the performance dynamics across different algorithms.

C.1.1 Synthetic Benchmarks

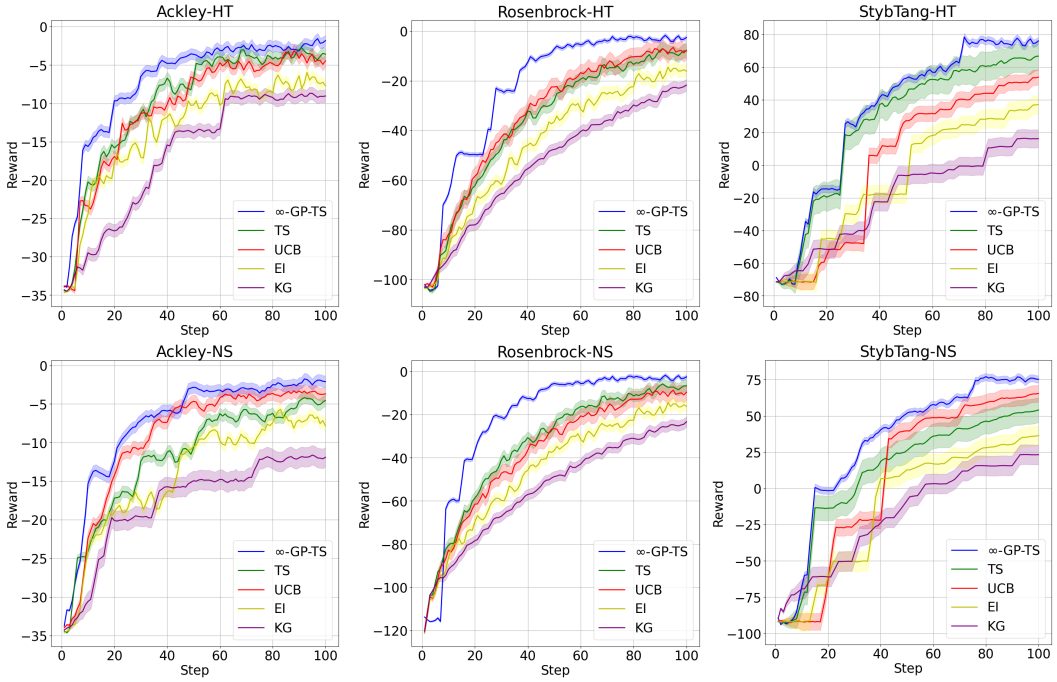

Figure 5: Cumulative regret in synthetic benchmarks

Figure 5 shows the evolution of instant regret over time across various synthetic benchmark functions. The compared methods—TS, UCB, EI, and KG—are all implemented using standard Gaussian Process surrogate models. Overall, we observe that ∞-GP-TS achieves the best performance in settings with nonstationary or heavy-tailed reward structures, significantly outperforming classical GP-based approaches. In the early iterations (e.g., the first 10 steps), the performance of ∞-GP-TS is initially lower due to its stronger exploration behavior. However, it quickly adapts and achieves lower regret as the optimization progresses, demonstrating its ability to efficiently learn complex reward landscapes.

C.1.2 Real-world Benchmark

Figure 6 shows the runtime performance (instant regret) on three real-world benchmarks: Portfolio, HPOBench, and NASBench. Across all tasks, NBO consistently outperforms other methods, achieving faster convergence and lower regret. In the Portfolio task, NBO quickly surpasses all baselines; in HPOBench and NASBench, it maintains a clear lead throughout the optimization process. These results highlight NBO's strong generalization ability and its effectiveness in complex, real-world settings.

C.2 More Experiment Details

In this subsection, we present more experiment details, including the synthetic benchmarks, real-world benchmarks and the prompt optimization problem.

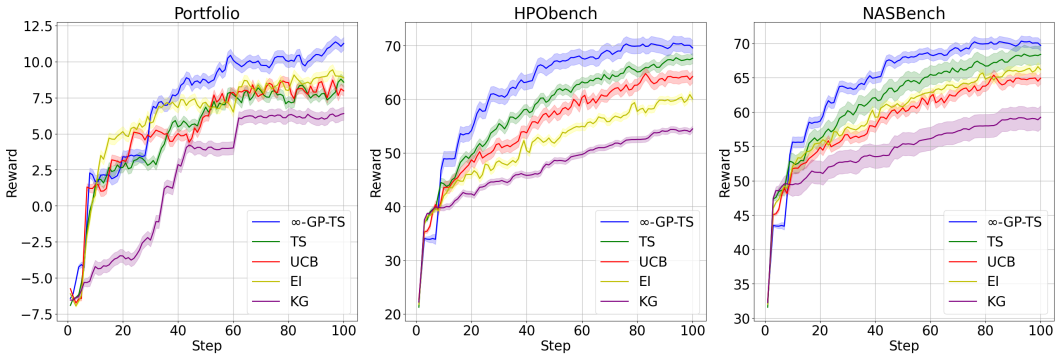

Figure 6: Cumulative regret in real-world benchmarks

C.2.1 Benchmark Function Definitions

We consider three popular synthetic test functions commonly used in global optimization benchmarks: the Ackley function, the Rosenbrock function, and the Styblinski–Tang (StybTang) function. These functions are defined over a d-dimensional input vector $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$.

Ackley Function.

$$f_{\text{Ackley}}(x) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1),$$

with default parameters $a = 20$, $b = 0.2$, and $c = 2\pi$.

Rosenbrock Function.

$$f_{\text{Rosenbrock}}(x) = \sum_{i=1}^{d-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right].$$

Styblinski–Tang Function.

$$f_{\text{ST}}(x) = \frac{1}{2}\sum_{i=1}^{d}\left(x_i^4 - 16x_i^2 + 5x_i\right).$$

To introduce more challenging and realistic scenarios, we further consider the following two variants of these base functions: **Heavy-tailed (HT) variant:** We add random noise to the function outputs, where the noise is sampled from a Weibull distribution. **Non-stationary (NS) variant:** We modulate each base function by a spatially-varying multiplicative factor:

$$f_{\text{NS}}(x) = (1 + \alpha \sin(x)e^x) \cdot f(x),$$

where $\alpha > 0$ controls the magnitude of non-stationarity. This formulation introduces location-dependent amplitude variations, breaking the stationarity assumption often made in kernel-based methods.

C.2.2 Real-world Benchmark Details

Portfolio [Cakmak et al., 2020] We tune three hyperparameters of a trading strategy: risk aversion $[0.1, 1000]$, trade aversion $[5, 8]$, and holding cost multiplier $[0.1, 100]$. The environment includes two random variables: bid-ask spread $\sim \mathcal{U}[10^{-4}, 10^{-2}]$ and borrowing cost $\sim \mathcal{U}[10^{-4}, 10^{-3}]$. The objective is to maximize the $\text{VaR}_{0.8}$ of the average daily return over four years. To reduce cost, we build a GP surrogate model using 3000 evaluations selected via Sobol sampling, as in Cakmak et al. [2020].

HPOBench [Eggensperger et al., 2021] provides standard hyperparameter tuning tasks for ML models, and we use the MLP task in the HPOBench in our experiment.

NASBench201 [Dong and Yang, 2020] is a neural network search task on CIFAR-100. Each architecture is represented by a 4-node directed acyclic graph with operations selected from a fixed set of 5 choices (e.g., conv, skip, zero). The total search space includes 15,625 architectures. Each architecture has been trained multiple times with fixed hyperparameters and its validation/test accuracy and training cost are pre-recorded. For our experiments, we query the test accuracy of a selected architecture based on its index as the ground-truth objective.

C.2.3 Prompt Optimization Experiment Details

We conduct prefix prompt optimization experiments on seven language understanding datasets. All experiments are performed using the Alpaca-7B model [Taori et al., 2023], which remains frozen throughout. The decision variable is a continuous prefix embedding prepended to each input; its quality is evaluated based on downstream task performance (e.g., accuracy). Due to the high dimensionality of the prompt embedding space, we apply Uniform Manifold Approximation and Projection (UMAP) [McInnes et al., 2018] to project the original space into a lower-dimensional latent space.

- **SST-2** and **SST-5** are sentiment classification tasks based on the Stanford Sentiment Treebank [Socher et al., 2013]. SST-2 is a binary classification task where each sentence is labeled as either positive or negative. SST-5 is a more challenging 5-class variant with labels: very negative, negative, neutral, positive, and very positive. Both tasks are derived from movie review sentences and are standard benchmarks for evaluating natural language understanding.

- **MR** [Pang and Lee, 2005] is a sentiment classification dataset composed of full movie reviews with hidden star ratings. The authors constructed both three-class (negative, neutral, positive) and four-class versions of the dataset by removing explicit rating indicators and grouping documents by rating level.

- **CR** [Hu and Liu, 2004] is a sentiment classification dataset constructed from customer reviews of various electronic products collected from Amazon and CNET. Each review is labeled as either positive or negative at the sentence level.

- **AG's News** [Zhang et al., 2015] is a topic classification dataset consisting of news articles categorized into 4 classes: World, Sports, Business, and Science/Technology. Each sample includes the article's title and a short description. The dataset contains 120,000 training samples and 7,600 test samples, and is commonly used to benchmark text classification models. We treat this as a 4-class supervised learning task using the full text (title + description) as input.

- **TREC** [Voorhees and Tice, 2000] is a question classification dataset designed for evaluating systems that categorize natural language questions into broad types. The standard setup includes 6 coarse-grained classes (e.g., Person, Location, Numeric) and 50 fine-grained subclasses. We use the 6-way classification task with 5,452 training and 500 test examples.

- **Subj** [Pang and Lee, 2004] is a sentence-level subjectivity classification dataset consisting of 10,000 sentences: 5,000 subjective snippets from movie review sites and 5,000 objective sentences from plot summaries on IMDb. Each sentence is labeled as either subjective or objective.

	GP-TS	GP-UCB	GP-EI	GP-KG	∞-GP-TS
Time(second)	2773	2649	3017	3388	2813

Table 2: Computation time (s) for five GP-based algorithms.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: the abstract and introduction clearly state the contributions and scope of this paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See section 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: See Assumption 1 and Theorem 1.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in Appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The information needed to reproduce the experiment is provided in Section 6 and the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All datasets used are publicly available. Code, configuration files, and instructions for reproducing experiments will be released publicly upon publication.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 6 and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in Appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See Section 6, Figure 2 and Figure 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Section 6 and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This work is a theoretical study focused on the development and analysis of BO algorithms. Therefore, we believe this research does not raise direct societal impact concerns.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper does not release any pretrained models, scraped datasets, or generative components that pose a risk of misuse or dual-use. The work is theoretical and does not involve the development or deployment of models or data with potential for harm. Therefore, safeguards for responsible release are not applicable.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All third-party assets used in this work, including publicly available datasets and code libraries, are properly credited in the main paper

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release a new codebase to support the implementation of our proposed method. The code is fully documented and includes usage instructions, dependencies, and example scripts to reproduce all main results. An anonymized version of the code is included in the supplementary materials for review.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: the paper does not involve crowdsourcing

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve human subjects, crowdsourcing, or any form of user study.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This research does not use LLM as part of its core methods, experiments, or contributions. LLMs were not involved in any component that affects the originality.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.