LEARNING COLLISION-FREE LATENT SPACE FOR BAYESIAN OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning and optimizing a blackbox function is a common task in Bayesian optimization. In real-world scenarios (e.g., tuning hyper-parameters for deep learning models, synthesizing a protein sequence, etc.), these functions tend to be expensive to evaluate and often rely on high-dimensional inputs. While classical Bayesian optimization algorithms struggle in handling the scale and complexity of modern experimental design tasks, recent works attempt to get around this issue by applying neural networks ahead of the Gaussian process to learn a (low-dimensional) latent representation. We show that such learned representation often leads to *collision* in the latent space: two points with significant different observations get too close in the learned latent space. Collisions could be regarded as additional noise introduced by the traditional neural network, leading to degraded optimization performance. To address this issue, we propose *Collision-Free Latent Space Optimization* (CoFLO), which employs a novel regularizer to reduce the collision in the learned latent space and encourage the mapping from the latent space to objective value to be Lipschitz continuous. CoFLO takes in pairs of data points and penalizes those too close in the latent space compared to their target space distance. We provide a rigorous theoretical analysis of the regret of the proposed algorithm. Our empirical results further demonstrate the effectiveness of CoFLO on several synthetic and real-world Bayesian optimization tasks, including a case study for computational cosmic experimental design.

1 INTRODUCTION

Bayesian optimization is a classical sequential optimization method and is widely used in various fields, including scientific experimental design, automated hyper-parameter tuning, etc. Many of theses applications involve evaluating an expensive blackbox function; therefore the number of queries should be minimized. A fundamental problem in Bayesian optimization is the exploration-exploitation dilemma: we have to sequentially make decisions with uncertain outcomes, with a potentially large (even infinite) number of available decisions; at each step, we have to trade estimating the function (exploration) and choosing decisions maximizing the estimated function (exploitation).

A common way to model the unknown function is via Gaussian Processes (GPs). GPs have been extensively studied under the bandit optimization setting, and has proven to be an effective approach for addressing a broad class of black-box function optimization problems. A computational challenge for learning with GPs concerns with optimizing specific kernels used to model the covariance structures of GPs. As such optimization task depends on the dimension of feature space, for high dimensional input, it is often prohibitively expensive to train a Gaussian process model. Meanwhile, GPs is not intrinsically designed to deal with structured input that has a strong correlations among different dimensions, e.g., the graphs and time sequences. Therefore, dimensionality reduction algorithms are needed to speed up the learning process.

Recently, it has become popular to investigate GPs in the context of latent space models. As an example, deep kernel learning (Wilson et al., 2016) simultaneously learns a (low-dimensional) data representation and a scalable kernel via an end-to-end trainable deep neural network. In general, the neural network is trained to learn a simpler latent representation with reduced dimension and has the structure information already embedded for the Gaussian process. Such a combination of neural

network and Gaussian process could improve the scalability and extensibility of classical Bayesian optimization, but it could also pose new challenges for the optimization task (Tripp et al., 2020). One critical challenge brought by introducing the neural network is that the latent representation is prone to *collisions*: two points with significant different observations can get too close in the latent space. The collision effect is especially obvious when information is lost in dimension reduction, and/or when the training data is limited in size in Bayesian optimization.

As illustrated in figure 1, data points with drastically different observations are transformed to close positions in the latent space via the neural network. Such collisions could be regarded as additional noise introduced by the traditional neural network.. Although Bayesian optimization could handle mild noisy observations, the collision in latent space could deteriorate the performance of Bayesian optimization as it is challenging to model the collision, and theoretically, the additional noise will enlarge the regret bound for classical Bayesian optimization algorithms (Srinivas et al., 2010).

Overview of main results To mitigate the collision effect, we propose a novel regularization scheme which can be applied as a simple plugin amendment for the latent space-based Bayesian optimization models. The proposed algorithm, namely Collision-Free Latent Space Optimization (CoFLO), uses the regularized neural network to learn a latent space and to conduct periodic retraining during the optimization.

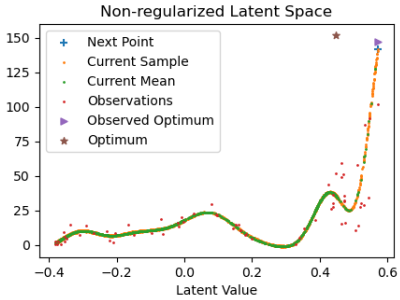


Figure 1: The 1-dimensional latent space shown here demonstrates the collision's negative influence in optimization tasks. Since the data points around the optimum severely collided, BO is misguided to the sub-optimum.

Concretely, CoFLO integrates a novel *pairwise collision penalty* defined on both the target space and latent space, to largely mitigate the negative influence of the latent space collision, as well as to boost the performance of Bayesian optimization. When the regularizer is uniformally applied to the latent space, CoFLO is designed to mitigate the collision effect in the whole latent space. However, in global optimization tasks, a reasonable collision regularizer should prioritize dealing with collisions close to the possible optimum regions. Therefore, we further propose a optimization-oriented regularization scheme, where we assign a higher weight for the collision penalty of a pair of points closer to the optimum region in the latent space. This dynamically-weighted CoFLO is designed to dynamically assess the importance of a collision during optimization. Comparing to the uniform collision penalty over the latent space, the dynamic weighting mechanism has demonstrated drastical improvement over the state-of-the-art latent space-based Bayesian optimization models.

We summarize our the key results below:

- We propose a novel regularization scheme, as a simple plugin amendment for latent space-based Bayesian optimization models that could penalize the collision in latent space.
- We propose an optimization-aware dynamic weighting mechanism for adjusting the collision penalty to further improve the effectiveness of regularization for the Bayesian optimization task.
- We provide theoretical analysis for the performance of Bayesian optimization on regularized latent space.
- We conducted an extensive empirical study on four synthetic and real-world datasets, including a real-world case study for cosmic experimental design, and demonstrate strong empirical performance for our algorithm.

2 Related Work

Bayesian optimization has demonstrated promisming performance in various cost-sensitive, constrained optimization tasks (Shahriari et al., 2016). However, due to its intrinsic computational limitation in the high-dimensional regime, its applicability has been restricted to relatively simple taskss. In this section, we provide a short survey on several methods proposed to overcome the dimensionality problem in both the optimization and regression literature.

Deep kernel learning Deep kernel learning (DKL) (Wilson et al., 2016) combines the power of the Gaussian process and that of neural network by introducing a deep neural network g to learn the mapping $g : X \to Z$ and use Z as the input of the Gaussian process. The neural network g and a spectral mixture base kernel k forms a scalable expressive closed-form covariance kernels for Gaussian processes $k_{DK}(x_i, x_j) \to k(g(x_i), g(x_j))$. Despite of encouraging results in numerous regression tasks, it remains unclear whether DKL is readily applicable to Bayesian optimization. The first problem is the expensive access to training data, since Bayesian optimization's common setting does not match with the DKL regression setting. Second problem lies in the difference between regression task and sequential optimization task objective. While DKL focuses on improving the general regression performance, it does not specifically address the problem caused by collisions, which—as we later demonstrate in section 3.3—could be harmful for sequential decision making.

Representation learning and latent space optimization Aiming at improving the scalability and extensibility of the Gaussian process, various methods are proposed to reduce the dimensionality of the original input. Djolonga et al. (2013) assume only a subset of input dimensions various, and the kernel is smooth(of a small norm in Reproducible Kernel Hilbert Space). Using low-rank matrix recovery techniques, they construct the underlying subspace of the unknown function. Huang et al. (2015) use Autoencoder to learn a low-dimension in an unsupervised way to increase GP's scalability in regression tasks. Snoek et al. (2015) further propose to learn a pretrained encoder neural network before BO. Lu et al. (2018) learn a Variational Auto-encoder iteratively during sequential optimization to embed the structure of the input. The possible issue with the latent space learning for Bayesian optimization is that a pre-trained neural network might not learn adequate information around the more promising region of input space, and the latent space could be outdated without a continuous updating with the latest acquired observation. In Tripp et al. (2020), they propose to use a periodical weighted retrain to learn a better latent space and to minimize iterations needed for LSO. They claim that by assigning a higher weight to more promising data points in the original input space, the model could focus more on learning high-value regions and allow a substantial extrapolation in the latent space to accelerate the optimization. However, such an framework doesn't explicitly deal with collisions in the latent space, which can degrade any latent space optimization algorithm.

3 Problem Statement

In this section, we introduce necessary notations and formally state the problem. We target at the problem of sequentially optimizing the function $f : D \to \mathbb{R}$. In each round t, we pick a point $x_t \in D \subseteq \mathbb{R}^d$ and evaluate the function value there with additive noise: $y_t = f(x_t) + \epsilon_t$ with $\epsilon_t \sim N(0, \sigma^2)$ i.i.d. Gaussian noise. Our goal is to maximize the sum of rewards $\sum_{t=1}^{T} f(x_t)$ or equally minimize the regret $\sum_t^T r_t = \sum_t^T f(x^*) - f(x_t)$. Here $x^* = \max_{x \in D} f(x)$.

3.1 Bayesian Optimization

Bayesian optimization methods are typically using Gaussian process (GPs) as the statistic model for the unknown objective function. The major advantage of using GPs is that GPs present a computationally tractable solution to depict a sophisticated and consistent view across the space of all possible function (Rasmussen and Williams, 2005), which allows closed-form posterior distribution in the function space. In the beginning, BO methods assign a prior to the objective black-box function. Then BO iteratively uses the observations to calculate a posterior distribution in the function space and maximize an auxiliary function measuring each point's contribution to finding the optimum, to select the next point for evaluation.

Formally, in Bayesian optimization we assume that f follows a $\mathcal{GP}(m(x), k(x, x'))$, where $m(x)$ is the mean function, $k(x, x')$ is the kernel or covariance function. Through out the paper, we use squared exponential kernel, $k_{SE}(x, x') = \sigma_{SE}^2 \exp\left(-\frac{(x-x')}{2l}\right)$, where length scale l determines the length of the "wiggles" and the output variance σ_{SE}^2 determines the average distance of the function away from its mean. At iteration T, given the historically selected points $A_T = \{x_1, ..., x_t\}$ and the corresponding noisy evaluations $\mathbf{y}_T = [y_1, ...y_T]$. The posterior over f is a GP again, with mean

$\mu_T(x)$, covariance $k_T(x, x')$, and variance $\sigma_T^2(x)$.

$$\mu_T(x) = k_T(x)^T(K_T + \sigma^2 I)^{-1}\mathbf{y}_T$$
$$k_T(x, x') = k(x, x') - k_T(x)T(K_T + \sigma^2 I)^{-1}k_T(x')$$
$$\sigma_T^2(x) = k_T(x, x)$$

where $k_T(x) = [k(x_1, x), ...k(x_T, x)]^T$ and K_T is the positive definite kernel matrix $[k(x, x')]_{x,x' \in A_T}$. After having the posterior, the acquisition function $\alpha : \mathcal{D} \to \mathbb{R}$ use it in order to select the next point to evaluate the function at. Various acquisition functions have been proposed in the literature, with the most popular ones: Upper Confidence Bound (UCB) (Srinivas et al., 2010) and Thompson sampling (TS) (THOMPSON, 1933). UCB use the upper confidence bound and has rigorous sublinear regret bound: $\alpha_{UCB}(x) = \mu_t(x) + \beta^{1/2}\sigma_t(x)$ Here $\beta(x)$ is the confidence coeffecient. Thompson sampling usually outperform UCB yet doesn't come with theoretical analysis. It samples a function $\tilde{f}_t$ from the GP posterior $\tilde{f}_t \sim \mathcal{GP}(\mu_t(x), k_t(x, x'))$, then use the sample as the acquisition function: $\alpha_{TS}(x) = \tilde{f}_t(x)$

Remark. *Regret is commonly used as performance metric for BO methods. In this work we focus on the simple regret* $r_T^* = f(x^*) - \max_{t<t} f(x_t)$ *and cumulative regret* $R(T) = \sum_t^T r_t$. *Here* $r_t = f(x^*) - f(x_t)$.

3.2 Latent Space Optimization

Latent Space Optimization (LSO) by BO is a method for solving optimization problems. A generative model $g : D \to Z$ transforms the original input space D to the latent space Z. And there exists an objective model $h : Z \to \mathbb{R}$ such that $f(x) \approx h(g(x)),\ \forall z \in Z$. In LSO, the generative model g and base kernel k could be regarded as a integrated kernel $k_{nn}(x, x') = k(g(x), g(x'))$. Thus the actual input for classical BO is the latent space Z and the objective function is h. With the integrated acquisition $\alpha_{nn}(x) = \alpha(g(x))$, there is no need to find a inverse mapping g^{-1} as discussed in Tripp et al. (2020) since BO could directly select $x_t = \arg\max_{x_t \in D} \alpha_{\text{nn}}(x)\ \forall t \leq T$ and evaluate f. While BO could still leverage the power of g, usually a neural network, to actually work on a lower-dimension representation.

3.3 The Collision Effect of LSO

The mapping $g : D \to Z$ is learned by the neural network, which offers no guarantee to avoid the unwanted collision in the learned latent space Z. For a noise-free evaluation, the simple collision in Z would be $\exists x_i, x_j \in D$, when $g(x_i) = g(x_j)$, $|f(x_i) - f(x_j)| > 0$. Such collision could be regarded as an unknown noise on the observations introduced by neural network g.

For a noisy observation $y = f(x) + \epsilon$, the collision could be defined as for a $\rho > 0$, $\exists x_i, x_j \in D$, $|g(x_i) - g(x_j)| < \rho|y_i - y_j|$. When the distance for a pair in the latent space is too close comparing to its observation difference, such that an additional noise with unknown model and parameter is added to the original ϵ noise.

In general, the collision could degrade the performance of LSO. Since it is of unknown model and parameter, it's challenging to tackle such collision in LSO even if we regard it as additional noise and increase the noise variance in the Gaussian process. Thus it's necessary to mitigate the collisions' negative effect by directly reducing it in the representation learning phase.

4 Algorithm

In this section, we introduce Collision-regularized Latent Space Optimization (CoFLO), an algorithmic framework designed to mitigate the collision effect.

4.1 Collision-regularized Latent Space Optimization

The major challenge in solving the collision in latent space is that unlike the traditional regression problem, we cannot quantify it on a single point's observation. We can only quantify and solve the

collision with a pair of data points and the corresponding observations. Therefore we define the collision penalty base on the pair of representation and further a pair loss function for training to reduce collision. Based on the loss, we propose the novel regularization and optimization algorithm. The proposed algorithm first uses the pair-wise input and concurrently feeds them into the same network and then calculates the pair loss function. We demonstrate this process in figure 2.

Given the embedding as input, the neural network could be pre-trained on a relative task like DKL by Wilson et al. (2016) or untrained. To mitigate the neural network's collision problem, the latent space optimization framework will update the weights and parameters periodically with regularization.

Algorithm 1 Collision-Regularized Latent Space Optimization (CoFLO)

1: **Input**: Regularization parameters ρ, penalty parameter λ, retrain interval $\tilde{T}$, weighting parameter γ, neural network M_0, base kernel K_0, prior mean μ_0, total time steps T;
2: **for** $t = 1$ *to* T **do**
3: $\quad x_t \leftarrow \arg\max_{x \in D} \alpha(M_t(x))$ ▷ maximize acquisition function
4: $\quad y_t \leftarrow$ evaluation on x_t ▷ Update observation
5: $\quad$ **if** $t \equiv 0 \pmod{\tilde{T}}$ **then**
6: $\quad\quad M_{t+1}, K_{t+1} \leftarrow$ retrain M_t and K_t with the pair loss function $L_{\rho,\lambda,\gamma}(M_t, K_t, D_t)$ as defined in equation 1 ▷ Periodical retrain
7: $\quad$ **end if**
8: **end for**
9: **Output**: $\max_t y_t$

4.2 Collision Penalty

Using the pair-wise input, we quantify the collision with the definition proposed in Section 3. We feed pairs of data points into the neural network and get the latent space representation. Apart from maximizing the GP's likelihood, we concurrently calculate the collision on each pair and penalize only if the value is positive.

$$p_{i,j} = \max(\lambda|y_i - y_j| - |z_i - z_j|, 0)$$

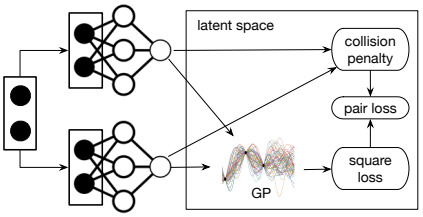


Figure 2: CoFLO schematic

4.3 Dynamic Weight

With limited historical data, it is hard to eliminate the collision by minimizing the collision penalty and the GP's regression loss. The good thing is, in the optimization tasks, it is unnecessary to learn equally good representation for low-value regions and we can improve the learned latent space by focusing on the potential high-value region. Thus we propose the collision penalty's dynamic weight, which uses the objective values for each pair in each iteration. Formally, for any pair $((x_j, z_j, y_j), (x_i, z_i, y_i))$ in a batch of observation pairs $D_t = \{((x_m, z_m, y_m), (x_n, z_n, y_n))\}_{m,n}$, the weight is defined as $w_{ij} = \frac{e^{\gamma(y_i + y_j)}}{\sum_{(m,n) \in D_t} e^{\gamma(y_m + y_n)}}$.

Then the weighted penalty for each pair is defined as $\tilde{p}_{ij} = p_{ij} w_{ij}$. Combining the collision penalty and regression loss of GP, the pair loss function L for the latent space learning is defined as

$$L_{\rho,\lambda,\gamma}(M_t, K_t, D_t) = \frac{1}{||D_t||^2} \sum_{i \in D_t, j \in D_t} (GP_{K_t}(M_t(x_i)) - y_i)^2 + (GP_{K_t}(M_t(x_j)) - y_j)^2 + \rho \tilde{p}_{ij} \tag{1}$$

5 Theoretical Analysis

In this section, we offer a theoretical analysis for the regret bound of CoFLO when applying the regularization to learn a collision-free latent space. We first link the proposed collision penalty to Lipschitz-continuity, and then use the continuity to provide a reduced regret bound.

5.1 LIPSCHITZ CONTINUITY

The collision penalty encourages the Lipschitz-continuity for h. Formally, the proposed regularization promotse to learn a latent space where $\forall x_i, x_j \in D$, $z_i = g(x_i), z_j = g(x_j) \in Z$,

$$|g(x_i) - g(x_j)| \leq \lambda |f(x_i) - f(x_j)|$$

This is exactly the Lipschitz-continuity for h. Unlike typical smoothness in BO, a function can be non-smooth and still Lipschitz continuous. In recent work, Ahmed et al. (2019) leverage the Lipschitz-continuity of the objective function to propose improved acquisition functions based on the common acquisition functions, and proves the improvement in regret bound both theoretically and empirically. Here we offer a proof for the regret bound based on the results in Srinivas et al. (2010) and elaborate the improvement in regret bound.

5.2 COLLISION-FREE

Theorem 1. *Let $Z \subset [0, r]^d$ be compact and convex, $d \in N, r > 0, \lambda \geq 0$. Suppose that the objective function h on latent space is a sample path from the GP and is Lipschitz continuous with Lipschitz constant at most λ. Pick $\delta \in (0, 1)$, and define $\beta_t = 2\log\left(\pi^2 t^2/6\delta\right) + 2d\log\left(\lambda r d t^2\right)$. Running the GP-UCB with β_t for a sample h of a GP with mean function zero and covariance function $k(x, x')$, we obtain a regret bound of $O^*(\sqrt{dT\gamma_T})$ with high probability. Precisely, with $C_1 = 8/\log\left(1 + \sigma^{-2}\right)$. We have $\mathbb{P}\left[R_T \leq \sqrt{C_1 T \beta_T \gamma_T} + 2\right] \geq 1 - \delta$. Here γ_T is the maximum information gain after T iterations, defined as $\gamma_T := \max_{A \subset Z, |A| = T} I(y_A; h_A)$.*

Comparing to Theorem 2 in Srinivas et al. (2010) which offers the regret bound with the sub-gaussianity assumption on the objective functions derivative, the second part of the regret bound doesn't rely on δ. The coefficients are also slightly smaller as the deterministic bound for f's derivative avoids using union bound.

Remark. *With the proposed regularization, we could encourage h to be Lipschitz-continuous on the latent space. Ideally, when the collision penalty $p_{i,j}(\lambda)$ is trained to be zero for all the data points in the latent space, we could claim that h is Lipschitz-continuous with Lipschitz constant at most λ. Then applying the theorem result: $\beta_t = 2\log\left(\pi^2 t^2/6\delta\right) + 2d\log\left(\lambda r d t^2\right)$, we could reduce the regret bound by choosing smaller λ. However, in practice, since the observation can be noisy, we need to choose a λ big enough to tolerant the noise and Choosing λ to be unreasonably small can make it impossible to learn a meaningful representation.*

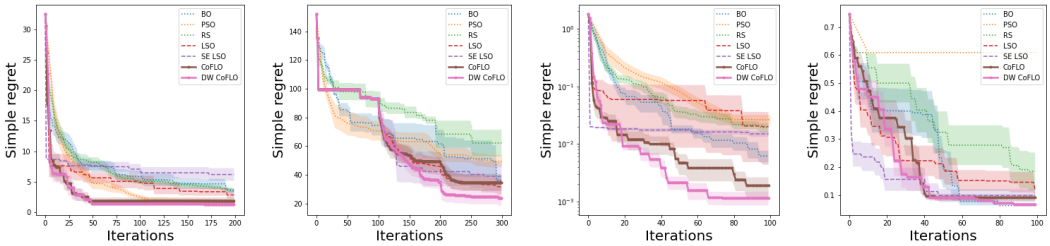
6 EXPERIMENTS

This section empirically evaluates our algorithm on two synthetic test function optimization tasks and two practical optimization problems.

6.1 EXPERIMENTAL SETUP

In the optimization tasks, we use Gaussian processes for all Bayesian optimization. For comparison, we set four baselines in the experiments. The rudimentary random selection algorithm(RS) shows the task complexity. Two popular optimization algorithms, particle swarm optimization(PSO) from Miranda (2018) and standard Bayesian optimization (BO) from Nogueira (2014), are tuned in each task. We implement a sample-efficient LSO(SE LSO) based on the algorithm and ranking weight proposed in Tripp et al. (2020). We also compare the non-regularized latent space optimization(LSO), Collision-Free Latent Space Optimization(CoFLO) and the dynamically-weighted CoFLO (DW CoFLO). The performance for each task is measured on ten thousand pre-collected data points.

One crucial problem in practice is tuning the hyper-parameters. The hyper-parameters for GP are tuned during periodically retrain in the optimization process by minimizing the loss function. The model M is selected to be the most basic architecture for the commonly limited training data due to the expensive access in BO setting. The coefficient ρ is, in general, selected to guarantee a similar order for the collision penalty to GP loss. The λ should be estimated according to



(a) Rastrigin 2D (b) Feynman III.9.52 (c) Supernova (d) SPOKES

Figure 3: Experiment results on four pre-collected datasets. Each experiment is repeated at least ten times.The colored area around the mean curve denotes the $\frac{\hat{\sigma}}{\sqrt{n}}$. Here $\hat{\sigma}$ denotes the empirical standard deviation. n denotes the number of cases repeated in experiments. The hyper-parameters are set as the following. The neural networks used to learn the representation are basically no more than three dense layers. The retrain interval $\tilde{T}$ are set to be 100 iterations for 3c 3a and 3d, 200 for 3b. The Regularization parameters ρ are set to be $1e^5$ for 3c 3a and 3d, $1e^3$ for 3b. The penalty parameter λ are all set to be $1e^{-1}$. The weighting parameter γ are set to be $1e^{-2}$. The prior mean μ_0 are all set to be 0. The squared exponential kernel is used as the GP covariance for all the four experiments.

the first several sampled data and tolerant the additive noise in the evaluation. γ controls the aggressiveness of the weighting. While γ should not be too close to zero (which is equivalent to uniform weight), an extremely high value could make the regularization too biased. Such a severe bias could possibly allow a heavily collided representation in most of the latent space and degrade regularization effectiveness. Here we use the first batch of observed samples to estimate the order of all observations and choose the appropriate γ.

6.2 Dataset and Results

2D-Rastrigin The Rastrigin function is a non-convex function used as a performance test problem for optimization algorithms. It was first proposed in 1974 by RASTRIGIN (1974) and used as a benchmark problem in Gaussian process regression algorithms in Cully et al. (2018). Formally, the 2D Rastrigin function is

$$f(x) = 10d + \sum_{i=1}^{d} x_i^2 - 10cos(2\pi x_i),\ d = 2$$

Here for convenience of comparison, we take the $-f(x)$ as the objective value to make the optimization tasks a maximization task. The neural network is pretrained on one hundred data points. As illustrated by figure 3a , CoFLO and DW CoFLO could quickly reach the sub-optimal, while the baselines generally suffer a bigger simple regret even after more iterations.

Feynman III.9.52 Equation Growing datasets have motivated pure data-backed analysis in physics. The dataset of 100 equations from the *Feynman Lecture on Physics* for the symbolic regression tasks in physics (Udrescu and Tegmark, 2020) could play the role as a test set for data-back analysis algorithms in physics. The III.9.52 we choose to test the optimization algorithms is

$$\rho_\gamma = \frac{p_d E_f t}{h/2\pi} \frac{sin((\omega - \omega_0)t/2)^2}{((\omega - \omega_0)t/2)^2}$$

The equations have six variables as inputs and are reported to require at least 10^3 data for the regression task. The neural network is randomly initialized at the beginning. As illustrated by figure 3b, in the first 100 iterations, CoFLO and DW CoFLO behaves similarly to random selection. After the first training at iteration 100, CoFLO and DW CoFLO approach the optimum at a much faster pace compared to the baselines; among them, DW CoFLO shows a faster reduction in simple regret.

Supernova The first real-world experiment is to perform maximum likelihood inference on 3 cosmological parameters, the Hubble constant $H_0 \in (60, 80)$, the dark matter fraction $\Omega_M \in (0, 1)$ and the dark energy fraction $\Omega_A \in (0, 1)$. The likelihood is given by the Roberson-Walker metric, which requires a one-dimensional numerical integration for each point in the dataset from Davis et al. (2007). The neural network is pretrained on one hundred data points. As illustrated by figure 3c, the simple regret SE LSO has a faster drop at the beginning, while later remained relatively stable and eventually ends at a similar level to LSO. These results demonstrate the efficiency of SE LSO when finding sub-optimal. However, without collision reduction, SE LSO couldn't outperform the LSO in the long run, where both reach their limitation. And the CoFLO and DW CoFLO demonstrate its robustness when close to the optimal as both constantly approach the optimal. Among them, DW CoFLO slightly outperform CoFLO.

Redshift Distribution The challenges in designing and optimizing cosmological experiments grow commensurately with their scale and complexity. Careful accounting of all the requirements and features of these experiments becomes increasingly necessary to achieve the goals of a given cosmic survey. SPOKES (SPectrOscopic KEn Simulation) is an end-to-end framework that can simulate all the operations and key decisions of a cosmic survey (Nord et al., 2016). It can be used for the design, optimization, and forecasting of any cosmic experiment. For example, some cosmic survey campaigns endeavor to observe populations of galaxies that exist at a specific range of redshifts (distances) from us. In this work, we use SPOKES to generate galaxies within a specified window of distances from Earth. We then minimize the Hausdorff distance between the desired redshift distribution and the simulation of specific cosmological surveys generated by SPOKES.

In experiments, the neural network is pretrained on two hundred data points. As illustrated by figure 3d, the simple regret SE LSO drops faster at the beginning. However, when it gets close to the sub-optimal where simple regret is approximately 0.15, it is caught up by both CoFLO and DW CoFLO and eventually get slightly outperformed. Such a result indicates that the collision problem could have more impact when the algorithm gets close to the optimal region. Notice that the rudimentary BO eventually outperformed the non-regularized LSO, indicate that without mitigation of collision, the learned representation could worsen the performance in the later stage when the algorithm gets close to the optimal. In conclusion, the mitigation of collision like CoFLO is necessary to further improve the later performance of LSO, when collision matters more in the near-optimal areas.

6.3 Discussion

In general, the four experiments' results demonstrate the robustness of our methods against collisions in the learned latent space. Our method outperforms all baselines. When compared to the sample-efficient LSO, the dynamically-weighted LSO performs better in most cases and shows a steady capability to reach the optimum by explicitly mitigating the collision in the latent space. In contrast, the Sample-efficient LSO might fail due to the collision problem.

7 Conclusion

We've proposed a novel regularization scheme on the latent space learning for Bayesian optimization. Applying the regularization on low-dimensional representation learning for Gaussian processes, the optimization algorithm termed Collision-free Latent Space Optimization (CoFLO) addresses the collision problem brought by dimensionality reduction, and improve the optimization performance for the Latent Space Optimization technique. The regularization is proved to be effective in mitigating the collision problem in the neural network learned latent space and therefore boosting the performance of the Bayesian optimization in the latent space. We demonstrate strong performance for CoFLO both theoretically and empirically on several synthetic and real-world physics datasets, and show that CoFLO is capable of dealing with high-dimensional input that could be highly valuable for applications in experiment design, such as cosmological survey scheduling, and could also be applied to other fields such as robotics application and computational biology.

REFERENCES

M. O. Ahmed, Sharan Vaswani, and M. Schmidt. Combining bayesian optimization and lipschitz optimization. *Machine Learning*, 109:79–102, 2019.

A. Cully, K. Chatzilygeroudis, F. Allocati, and J.-B. Mouret. Limbo: A Flexible High-performance Library for Gaussian Processes modeling and Data-Efficient Optimization. *The Journal of Open Source Software*, 3(26):545, 2018. doi: 10.21105/joss.00545.

T. M. Davis, E. Mortsell, J. Sollerman, A. C. Becker, S. Blondin, P. Challis, A. Clocchiatti, A. V. Filippenko, R. J. Foley, P. M. Garnavich, S. Jha, K. Krisciunas, R. P. Kirshner, B. Leibundgut, W. Li, T. Matheson, G. Miknaitis, G. Pignata, A. Rest, A. G. Riess, B. P. Schmidt, R. C. Smith, J. Spyromilio, C. W. Stubbs, N. B. Suntzeff, J. L. Tonry, W. M. Wood-Vasey, and A. Zenteno. Scrutinizing exotic cosmological models using ESSENCE supernova data combined with other cosmological probes. *The Astrophysical Journal*, 666(2):716–725, sep 2007. doi: 10.1086/519988. URL https://doi.org/10.1086%2F519988.

Josip Djolonga, Andreas Krause, and V. Cevher. High-dimensional gaussian process bandits. In *NIPS*, 2013.

Wenbing Huang, Deli Zhao, Fuchun Sun, Huaping Liu, and Edward Chang. Scalable gaussian process regression using deep neural networks. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 3576 – 3582. AAAI Press, 2015. ISBN 9781577357384.

Xiaoyu Lu, Javier Gonzalez, Zhenwen Dai, and Neil Lawrence. Structured variationally auto-encoded optimization. volume 80 of *Proceedings of Machine Learning Research*, pages 3267–3275, Stockholm International Fairs, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/lu18c.html.

Lester James V. Miranda. PySwarms, a research-toolkit for Particle Swarm Optimization in Python. *Journal of Open Source Software*, 3, 2018. doi: 10.21105/joss.00433. URL https://doi.org/10.21105/joss.00433.

Fernando Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python, 2014. URL https://github.com/fmfn/BayesianOptimization.

B. Nord, A. Amara, A. Réfrégier, La. Gamper, Lu. Gamper, B. Hambrecht, C. Chang, J.E. Forero-Romero, S. Serrano, C. Cunha, O. Coles, A. Nicola, M. Busha, A. Bauer, W. Saunders, S. Jouvel, D. Kirk, and R. Wechsler. Spokes: An end-to-end simulation facility for spectroscopic cosmological surveys. *Astronomy and Computing*, 15:1 – 15, 2016. ISSN 2213-1337. doi: https://doi.org/10.1016/j.ascom.2016.02.001. URL http://www.sciencedirect.com/science/article/pii/S2213133716300026.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

L. A. RASTRIGIN. Systems of extremal control. *Nauka*, 1974. URL https://ci.nii.ac.jp/naid/10018403158/en/.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Md. Mostofa Ali Patwary, Prabhat Prabhat, and Ryan P. Adams. Scalable bayesian optimization using deep neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2171 – 2180. JMLR.org, 2015.

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 International Conference on Machine Learning*, ICML'10, pages 1015–1022, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

WILLIAM R THOMPSON. ON THE LIKELIHOOD THAT ONE UNKNOWN PROBABILITY EXCEEDS ANOTHER IN VIEW OF THE EVIDENCE OF TWO SAMPLES. *Biometrika*, 25(3-4):285–294, 12 1933. ISSN 0006-3444. doi: 10.1093/biomet/25.3-4.285. URL https://doi.org/10.1093/biomet/25.3-4.285.

Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *ArXiv*, abs/2006.09191, 2020.

Silviu-Marian Udrescu and Max Tegmark. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6, 2020.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. volume 51 of *Proceedings of Machine Learning Research*, pages 370–378, Cadiz, Spain, 09–11 May 2016. PMLR. URL http://proceedings.mlr.press/v51/wilson16.html.

A Regret Bound for a Lipschitz-continuous Objective function

In this section, we provide the detailed proof for theorem 1. We first modify Lemma 5.7 and Lemma 5.8 in Srinivas et al. (2010) since we are assuming the deterministic Lipschitz-continuity for h. Use the same analysis tool Z_t defined as a set of discretization $Z_t \subset Z$ where Z_t will be used at time t in the analysis.

We choose a discretization Z_t of size $(\tau_t)^d$. so that $\forall z \in Z$,

$$||z - [z]_t||_1 \leq rd/\tau_t \tag{2}$$

where $[z]_t$ denotes the closest point in Z_t to z.

Lemma 1. *Pick $\delta \in (0,1)$ and set $\beta = 2log(\pi_t/\delta) + 2dlog(Lrdt^2)$, where $\sum_{t\geq 1} \pi_t^{-1} = 1,\ \pi_t > 0$. Let $\tau_t = Lrdt^2$. Hence then*

$$|h(z^*) - \mu_{t-1}([z^*]_t)| \leq \beta_t^{1/2}\sigma_{t-1}([z^*]_t) + 1/t^2 \quad \forall t \geq 1$$

holds with probability $\geq 1-\delta$. Here $z^* := g(x^*)$.

Proof. Using the Lipschitz-continuity and equation 2, we have that

$$\forall z \in Z, |h(z) - h([z]_t)| \leq Lrd/\tau_t$$

By choosing $\tau_t = Lrdt^2$, we have $|Z_t| = (Lrdt^2)^d$ and

$$\forall z \in Z, |h(z) - h([z]_t)| \leq 1/t^2$$

Then using Lemma 5.6 in Srinivas et al. (2010), we reach the expected result. □

Base on Lemma 5.5 in Srinivas et al. (2010) and Lemma 1, we could have the following result directly.

Lemma 2. *Pick $\delta \in (0,1)$ and set $\beta = 2log(2\pi_t/\delta) + 2dlog(Lrdt^2)$, where $\sum_{t\geq 1} \pi_t^{-1} = 1,\ \pi_t > 0$. Then with probability $\geq 1-\delta$, for all $t \in N$, the regret is bounded as follows:*

$$r_t \leq 2\beta_t^{1/2}\sigma_{t-1}(z_t) + 1/t^2$$

Proof. Using the union bound of $\delta/2$ in both Lemma 5.5 in Srinivas et al. (2010) and Lemma 1, we have that with probability $1-\delta$:

$$\begin{aligned} r_t &= h(z^*) - h(z_t) \\ &\leq \beta_t^{1/2}\sigma_{t-1}(z_t) + 1/t^2 + \mu_{t-1}(z_t) - h(z_t) \\ &\leq 2\beta_t^{1/2}\sigma_{t-1}(z_t) + 1/t^2 \end{aligned}$$

which complete the proof. □

Now we are ready to use the Lemma5.4 in Srinivas et al. (2010) and Lemma 2 to complete the proof of Theorem 1.

Proof. Using Lemma5.4 in Srinivas et al. (2010), we have that with probability $\geq 1-\delta$:

$$\sum_{t=1}^{T} 4\beta_t \sigma_{t-1}^2(x_t) \leq C_1 \beta_T \gamma_T \quad \forall T \geq 1$$

By Cauchy-Schwarz:

$$\sum_{t=1}^{T} 2\beta_t^{1/2} \sigma_{t-1}(x_t) \leq \sqrt{C_1 \beta_T \gamma_T} \quad \forall T \geq 1$$

Finally, substitute π_t with $\pi^2 t^2/6$ (since $\sum 1/t^2 = \pi^2/6$). Theorem 1 follows. □

B VISUALIZATION OF THE NEGATIVE INFLUENCE OF COLLISION IN LATENT SPACE

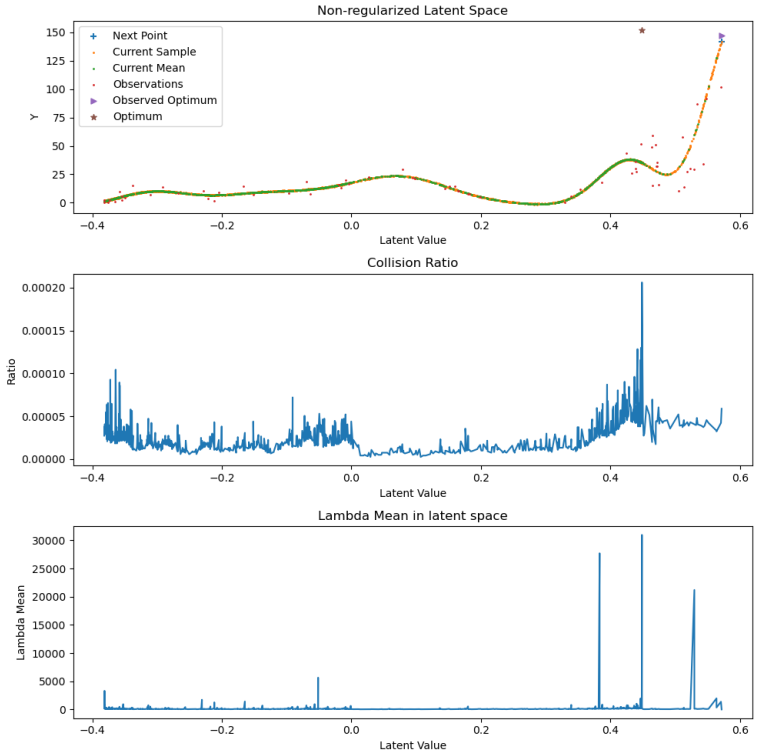


Figure 4: Illustrate the collision and quantified measurement of the collision. Here we propose two quantity measurement of the collision. For the second graph the y axis of the is the ratio of exceeding $|y_1 - y_2| > |z_1 - z_2| * L$. And for the third graph, the y axis of the third column is the mean of $\lambda = |y_1 - y_2|/|z_1 - z_2|$.