Design of Physical Experiments via Collision-Free Latent Space Optimization

Fengxue Zhang, Yair Altas, Louise Fan, Kaustubh Vinchure, Brian Nord, Yuxin Chen
University of Chicago
{zhangfx,atlas99,siqi,vinchure,nord,chenyuxin}@uchicago.edu

Abstract

Learning and optimizing a blackbox function is a common task in Physics experimental design. 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 a *collision* in the latent space: two points with significantly different observations get too close in the learned latent space. Collisions could be regarded as additional noise introduced by the 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. CoFLO takes in pairs of data points and penalizes those too close in the latent space compared to their distances in labels. Our empirical results demonstrate the effectiveness of CoFLO on several synthetic and real-world Bayesian optimization tasks, including a case study for the design of cosmological surveys.

1 Introduction

Bayesian optimization is a classical sequential optimization method widely used in automated experimental design in both science and engineering tasks, such as sequential design of physical experiments and hyper-parameter tuning. Many of these applications involve evaluating an expensive blackbox function; hence the number of function evaluations should be minimized. A common way to model the blackbox function is via Gaussian Processes (GPs). The computational challenge for learning with GPs requires optimizing specific kernels that are used to model the covariance structures of GPs. Notably, the optimization task depends on the dimensionality of the feature space, and it is often prohibitively expensive to train a GP under the high-dimensional regime. In such scenarios, dimensionality reduction algorithms are needed to accelerate the learning process.

Recently, it has become popular to investigate GPs in the context of latent space models. In general, a neural network is trained to learn a simpler latent representation with reduced dimension and has the structure information already embedded for the Gaussian process. As an example, deep kernel learning [1] simultaneously learns a (low-dimensional) data representation and a scalable kernel via an end-to-end trainable deep neural network. In [2], the authors propose to periodically retrain the generative model like generative adversarial network (GAN) [3] or Variational Autoencoders (VAE) [4] which learns the representation to improve the latent space for Bayesian optimization. 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.

Such a combination of neural networks and Gaussian processes could improve the scalability and extensibility of classical Bayesian optimization, but it also poses new challenges for the optimization

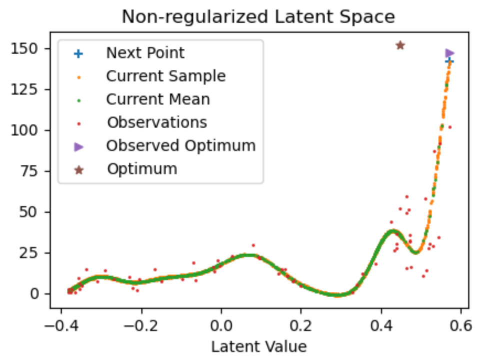


Figure 1: The collision effect in the 1-d latent space learned by non-regularized latent space algorithm on the 6-d Feynman III.9.52 dataset in [5], where BO is misguided to the sub-optimum. Here x-axis values are the latent space positions, while y-axis denotes the objective value.

task, because it does not specifically address the problem of *collisions*: two points with significant different observations can collide 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.

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 neural network. Although Bayesian optimization can 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 increase the regret bound for classical Bayesian optimization algorithms [6].

2 Method

In this section, we formally state the Bayesian optimization problem, and propose *Collision-Free Latent Space Optimization* (CoFLO), an algorithmic framework designed to mitigate the collision effect. We aim to sequentially optimize the function $f : \mathcal{D} \to \mathbb{R}$, where $\mathcal{D} \subset \mathbb{R}^d$ denotes the input domain. At round t, we pick a point $x_t \in \mathcal{D}$ and observe a noisy label of the function value: $y_t = f(x_t) + \epsilon_t$ where $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ is *i.i.d.* Gaussian noise. Our goal is to maximize the sum of rewards $\sum_{t=1}^{T} f(x_t)$ or equivalently minimize the regret $\sum_t^T r_t = \sum_t^T (\max_{x \in \mathcal{D}} f(x) - f(x_t))$.

2.1 Latent Space Optimization

Latent Space Optimization (LSO) is a two-staged data-driven approach for Bayesian global optimization: First, it learns a latent space mapping $g : \mathcal{X} \to \mathcal{Z}$ to convert the input space $\mathcal{X}$ to the latent space $\mathcal{Z}$. Then, it constructs an objective mapping $h : \mathcal{Z} \to \mathbb{R}$ such that $f(x) \approx h(g(x))$. Different from [2], we consider an end-to-end approach in which the latent space mapping g, together with a base kernel k defined on $\mathcal{Z}$ could be regarded as a *deep kernel* [1], denoted by $k_{\text{nn}}(x, x') = k(g(x), g(x'))$. Thus, the actual input space for BO is the latent space $\mathcal{Z}$ and the objective function is h. With acquisition function $\alpha_{\text{nn}}(x) := \alpha(g(x))$, it is unnecessary to compute an inverse mapping g^{-1} as discussed in [2], as BO could directly select $x_t = \arg\max_{x_t \in \mathcal{X}} \alpha_{\text{nn}}(x)\ \forall t \leq T$ and evaluate f. In the meantime, BO can leverage the latent space mapping g, usually represented by a neural network, to effectively learn and optimize the target function h on (a lower-dimension) input space $\mathcal{Z}$.

Note that by construction, the mapping $g : \mathcal{D} \to \mathcal{Z}$ (e.g., learned via a neural network) offers no guarantee to avoid collision in the learned latent space $\mathcal{Z}$. In the noise-free case, a *collision* in $\mathcal{Z}$ corresponds to a pair of points $x_i, x_j \in \mathcal{D}$, where $g(x_i) = g(x_j)$, $|f(x_i) - f(x_j)| > 0$. For a noisy observation $y = f(x) + \epsilon$, we define *collision* as follows: for $\rho > 0$, $\exists x_i, x_j \in D$, $|g(x_i) - g(x_j)| < \rho|y_i - y_j|$.

Algorithm 1 Collision-Regularized Latent Space Optimization (CoFLO)

```
1: Input: Penalty parameter $\lambda$ (cf. Equation 1), retrain interval $\tilde{T}$, importance weight parameter
   $\gamma$(cf. Equation 2), regularization weight $\rho$ (cf. Equation 3), neural network $M_0$, base kernel $K_0$,
   prior mean $\mu_0$, total time steps $T$;
2: for $t = 1$ to $T$ do
3:     $x_t \leftarrow \arg\max_{x \in D} \alpha(M_t(x))$                          ▷ maximize acquisition function
4:     $y_t \leftarrow$ evaluation on $x_t$                                        ▷ update observation
5:     if $t \equiv 0 \pmod{\tilde{T}}$ then
6:         $M_{t+1}, K_{t+1} \leftarrow$ 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 3                                                   ▷ periodical retrain
7: Output: $\max_t y_t$
```

2.2 Collision-Free Latent Space Optimization

The major challenge in eliminating the *collision effect* in the latent space is that, unlike the traditional regression problem, we cannot quantify it on a single point's observation. Instead, we can only characterize the severity of collisions on pairs of data points and the corresponding observations. Therefore, we define the *collision penalty* on pairs of data points and their corresponding predictions, and further introduce a *pair loss* for the model to explicitly mitigate the collision effect. We call this novel regularized LSO algorithm *Collision-Free latent space optimization* (CoFLO). CoFLO concurrently feeds pairwise inputs into the same network and then computes the pair loss function. We demonstrate this process in Figure 2, and provide a concrete instantiation of the algorithm in Appendix A.1. The pseudocode of CoFLO is summarized in Algorithm 1.

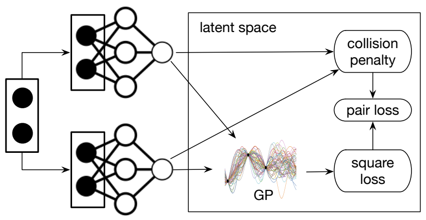


Figure 2: Schematic of CoFLO

As illustrated in Figure 2, we feed pairs of data points into the neural network and obtain their latent space representations. Apart from maximizing the GP's likelihood, we concurrently calculate the amount of collision on each pair, and penalize only if the value is positive. For $x_i, x_j \in \mathcal{X}$, $y_i = f(x_i) + \epsilon$, y_i are the corresponding observations, and $z_i = g(x_i)$, $z_j = g(x_j)$ are the corresponding latent space representations. We define the *collision penalty* as

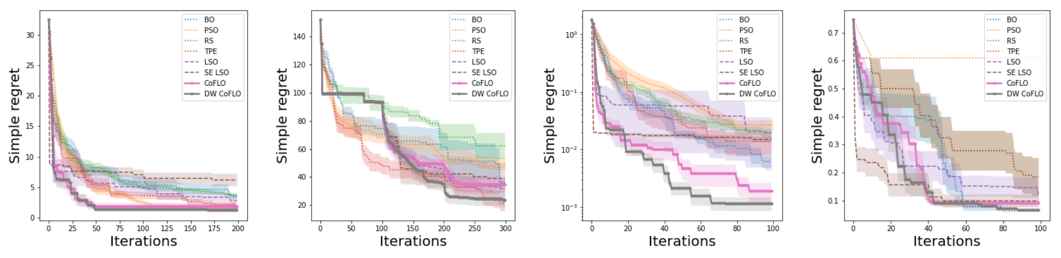
$$p_{ij} = \max(\lambda|y_i - y_j| - |z_i - z_j|, 0) \tag{1}$$

where λ is a penalty parameter that controls the smoothness of the target function $h : \mathcal{Z} \to \mathbb{R}$. Note that it is challenging to universally eliminate the collision effect by minimizing the collision penalty and the GP's regression loss—this is particularly true with a limited amount of training data. Fortunately, for global optimization tasks it is often unnecessary to optimize learned representations for sub-optimal regions. Therefore, we can dedicate more training resources to improve the learned latent space by focusing on the (potentially) near-optimal regions. Following this insight, we propose a weighted collision penalty function, which uses the objective values for each pair as importance weight at each iteration. Formally, for any pair $((x_j, z_j, y_j), (x_i, z_i, y_i))$ in a batch of observation pairs $D_t = \{((x_m, z_m, y_m), (x_n, z_n, y_n))\}_{m,n}$, we define the *importance-weighted penalty function* as

$$\tilde{p}_{ij} = p_{ij} w_{ij} \qquad \text{with} \qquad w_{ij} = \frac{e^{\gamma(y_i + y_j)}}{\sum_{(m,n)\in D_t} e^{\gamma(y_m + y_n)}}. \tag{2}$$

Here the importance weight γ is used to control the aggressiveness of the weighting strategy. Combining the above penalty function and the regression loss of GP, we define the *pair loss* function L as

$$L_{\rho,\lambda,\gamma}(M_t, K_t, D_t) = \frac{1}{||D_t||^2} \sum_{i,j \in D_t} \left((GP_{K_t}(M_t(x_i)) - y_i)^2 + (GP_{K_t}(M_t(x_j)) - y_j)^2 + \rho \tilde{p}_{ij} \right), \tag{3}$$



(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 (see Appendix A.2 for experimental details). The shaded 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.

Here, $GP_{Kt}(M_t(x_i))$ denotes the Gaussian process's posterior mean on x_i with kernel K_t and neural network M_t at timestep t. ρ denotes the regularization weight; as we demonstrate in Section 3, in practice we often choose ρ to keep the penalty at a order close to the regression loss.

3 Experiments

We consider four baselines in our experiments. The random selection algorithm (RS) is a natural baseline to demonstrate the complexity of the optimization task. Three popular optimization algorithms, namely particle swarm optimization (PSO) [7], Tree-structured Parzen Estimator Approach (TPE) [8], and standard Bayesian optimization (BO) [9] which uses Gaussian process as the statistical model and the upper confidence bound (UCB) as its acquisition function, are tuned in each task. Another baseline we consider is the Sample-Efficient LSO (SE LSO) algorithm proposed in [2], which learns latent space representation of inputs via a variational autoencoder. We also compare the non-regularized latent space optimization (LSO), Collision-Free Latent Space Optimization (CoFLO) and the Dynamically-Weighted CoFLO (DW CoFLO) proposed in this paper. The performance for each task is measured on 10,000 pre-collected data points.

3.1 Experimental Results

We now evaluate CoFLO on two synthetic datasets and two real-world datasets.

2D-Rastrigin The Rastrigin function is a non-convex function first proposed in 1974 [10] and has been widely used as a benchmark for Gaussian process regression: $f(x) = 10d + \sum_{i=1}^{d} x_i^2 - 10\cos(2\pi x_i)$ (where $d = 2$ denotes the dimensionality). Here, we take $-f(x)$ as the objective value to make the optimization tasks a maximization task.

The neural network is pretrained on 100 data points. As illustrated by Figure 3a , CoFLO, and DW CoFLO could quickly reach the (near-) optimal region, while the baselines generally suffer larger simple regret even after an excessive number of iterations.

Atomic Transition Probabilities Growing datasets have motivated the pursuit of data-only analysis in physics. The dataset of 100 equations from the *Feynman Lecture on Physics* for the symbolic regression tasks in physics [5] could play the role as a test set for such algorithms in physics. Equation 9.52 from the Feynman Lecture on physics gives probability for atomic transitions in the Ammonia maser $\rho_\gamma = \frac{p_d E_f t}{h/2\pi} \frac{sin((\omega-\omega_0)t/2)^2}{((\omega-\omega_0)t/2)^2}$. The equation has six variables as inputs and are reported to require at least 10^3 data for the regression task. The neural network is randomly initialized.

As illustrated by Figure 3b, in the first 100 iterations, CoFLO, and DW CoFLO behave similarly with random selection. After the first re-training step 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 Our next task is to perform maximum likelihood inference on a real-world dataset, with the goal to infer 3 cosmological parameters from observational data: 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 [11].

The neural network is pretrained on 100 data points. As illustrated by Figure 3c, the SE LSO has a faster drop in simple regret in the beginning; later it remained relatively stable and eventually ends at a similar level with LSO. These results demonstrate the efficiency of SE LSO when finding sub-optimal inputs. However, without collision reduction, SE LSO could not outperform LSO in the long run. Meanwhile, CoFLO and DW CoFLO make steady improvement as they approach the optimal input. Among these two, DW CoFLO slightly outperforms CoFLO.

Redshift Distribution Carefully accounting for all the requirements and features of cosmological experiments is increasingly important as the experiments grow in scale and complexity. SPOKES (SPectrOscopic KEn Simulation) is an end-to-end framework that can simulate all the major operations and critical decisions of a cosmological sky survey [12]. In this task, we use SPOKES to generate galaxies within a specified window of distances from Earth (redshifts) and then minimize the Hausdorff distance between the desired redshift window and the SPOKEs-generated simulations of cosmological surveys.

In our experiments, the neural network is pre-trained on 200 data points. As illustrated by Figure 3d, the simple regret of SE LSO drops faster at the initial phase. However, when it gets close to the (near-) optimal region where simple regret is approximately 0.15, it is caught up by both CoFLO and DW CoFLO, and eventually gets slightly outperformed. Such a result indicates that the collision effect could have more severe impact when the algorithm gets close to the optimal region. Notice that BO eventually outperformed the non-regularized LSO, indicating that without mitigation of the collision effect, the learned representation could worsen the performance at the later stage when the algorithm gets close to the optimal. In conclusion, the mitigation of collision in CoFLO is effective in further improving the performance of LSO, when collision plays a more significant role in the near-optimal areas.

3.2 Discussion

In general, our experimental results consistently demonstrate the robustness of our methods against collisions in the learned latent space. Our method outperforms all baselines; when compared to the SE LSO, the DW CoFLO 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 SE LSO might fail due to the collision problem.

4 Conclusion and Discussions

We have proposed a novel regularization scheme for latent space-based Bayesian optimization. Our algorithm—namely CoFLO—addresses the collision problem induced by dimensionality reduction, and improves the performance for latent space-based optimization algorithms. The regularization is proved to be effective in mitigating the collision problem in learned latent space, and therefore can boost the performance of the Bayesian optimization in the latent space. We demonstrate strong empirical results for CoFLO on several synthetic and real-world datasets, and show that CoFLO is capable of dealing with high-dimensional input that could be highly valuable for real-world experiment design tasks such as cosmological survey scheduling.

There are several challenges around the latent space learning in CoFLO that we hope to address in future studies. One challenge is the choice of the latent space, as was investigated in [13] under a slightly different context. In this work, we set the dimensionality of the latent spaces to 1 for simplicity and visualization. Despite the empirical improvement on various datasets, learning a latent space with higher dimension could still be worth trying if it eases the training difficulty of the neural network. Another challenge lies in the interpretability in the latent space. While interpretability of neural networks is of growing popularity in the Machine Learning community, the latent space learned for global optimization tasks could be of special interest as a better understanding of the latent space could help further boost the performance of LSO.

Acknowledgment

The authors thank James Bowden, Jialin Song, Thomas Desautels, and Yisong Yue for the helpful discussions. The project was supported by the Center for Data and Computing (CDAC) at the University of Chicago, via a JTFI AI + Science Grant.

Broader Impact

Our work contributes to several important subareas in the interaction between AI and the physical sciences, including active learning, Bayesian optimization and optimal experimental design in physics and cosmology/astrophysics. Our latent space-based Bayesian optimization framework not only suggests that one can combine the best of both worlds of deep learning and Bayesian optimization (i.e., the representation power of neural network and the expressiveness of GPs), it also sheds light on devising novel regularization schemes for more general classes of data-driven decision making problems.

References

[1] 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.

[2] 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.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc., 2014.

[4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

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

[6] 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.

[7] Lester James V. Miranda. PySwarms, a research-toolkit for Particle Swarm Optimization in Python. *Journal of Open Source Software*, 3, 2018.

[8] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24, pages 2546–2554. Curran Associates, Inc., 2011.

[9] Fernando Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python, 2014.

[10] L. A. RASTRIGIN. Systems of extremal control. *Nauka*, 1974.

[11] 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.

[12] 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.

[13] Wesley J Maddox, Gregory Benton, and Andrew Gordon Wilson. Rethinking parameter counting in deep models: Effective dimensionality revisited. *arXiv preprint arXiv:2003.02139*, 2020.

[14] Geoffrey E. Hinton, Oriol Vinyals, and J. Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

A Supplemental Materials on Algorithmic Details

A.1 Algorithmic Details on Neural Network Architecture

As the main goal of our paper was to showcase the performance of a novel collision-free regularizer, we picked our network architectures to be basic multi-layer dense neural network:

For SPOKES, we used a 5-layer dense neural network. Its hidden layers consist of 16 neurons with the Leaky Relu activation function, 8 neurons with a Sigmoid activation function, 4 neurons with a Sigmoid activation function, and 2 neurons with a Sigmoid activation function respectively. Each hidden layer also applies a dropout rate of 0.2. The output layer applies a Leaky Relu activation function.

For SuperNova, Feynman, and Rastrigin 2D, we used a 4-layer dense neural network. Its hidden layers consist of 8 neurons with a Sigmoid activation function, 4 neurons with a Leaky Relu activation function, and 2 neurons with a Leaky Relu activation function respectively. Each hidden layer applies a dropout rate of 0.2. The output layer also applies a Leaky Relu activation function.

The neural networks are trained using ADAM with a learning rate of $1e^{-2}$.

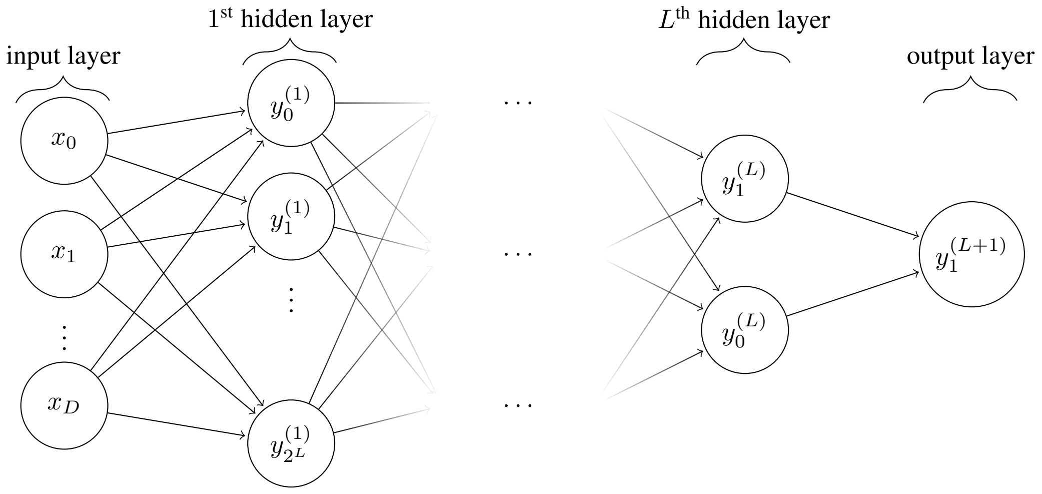


Figure 4: Network graph of a $(L+1)$-layer dense network with D input units and 1 output units. In our experiments L is set to be 4 for Rastrigin 2D, Feynman II.9.52, Supernova, and 5 for SPOKES.

A.2 Parameter Choices

One crucial problem in practice is tuning the hyper-parameters. The hyper-parameters for GP are tuned for periodically retraining in the optimization process, by minimizing the loss function on a validation set. For all our tasks, we choose a simplistic neural network architecture M, due to limited and expensive access to labeled data under the 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 the first several sampled data and tolerant the additive noise in the evaluation. γ controls the aggressiveness of the importance weight. While γ should not be too close to zero (which is equivalent to uniform weight), an extremely high value could make the regularization overly biased. Such a severe bias could possibly allow a heavily collided representation in most of the latent space and degrade regularization effectiveness. The value choice is similar to the inverse of the temperature parameter of softmax in deep learning [14]. Here we use the first batch of observed samples to estimate the order of all observations and choose the appropriate γ.

The retrain intervals $\tilde{T}$ are set to be 100 iterations for Figure 3c Figure 3a and Figure 3d, 200 for 3b. The Regularization parameters ρ are set to be $1e^5$ for Figure 3c Figure 3a and Figure 3d, $1e^3$ for Figure 3b. The penalty parameters λ are all set to be $1e^{-2}$. The weighting parameters γ are set to be $1e^{-2}$. The prior means μ_0 are all set to be 0. The squared exponential kernel is used to model the GP covariance for all the four experiments.

B Visualization of The Negative Influence of Collision in Latent Space

In Figure 5, we provide additional empirical results demonstrating the collision effect for the example provided in Figure 1. Here, we measure the collision effect via two different metrics as rigorously stated in the figure. We can observe the distribution of the collisions in the latent space, which indicates the severity of collision in the near-optimum region.

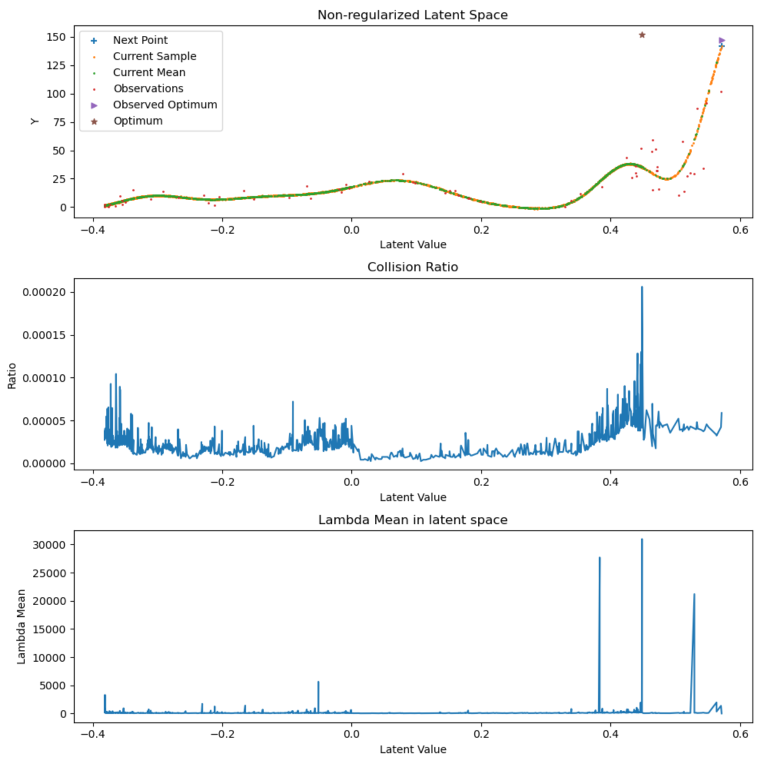


Figure 5: Illustrate the collision and quantified measurement of the collision. Here we propose two quantity measurement of the collision. For the second graph the y axis is the ratio of exceeding $|y_1 - y_2| > |z_1 - z_2| * L$. And for the third graph, the y axis of the third column is the mean of $\lambda = |y_1 - y_2| / |z_1 - z_2|$.