HyperBO+: Pre-training a universal hierarchical Gaussian process prior for Bayesian optimization

Zhou Fan
Harvard University
zfan@g.harvard.edu

Xinran Han
Harvard University
xinranhan@g.harvard.edu

Zi Wang
Google Research, Brain Team
wangzi@google.com

Abstract

We present HyperBO+: a framework of pre-training a hierarchical Gaussian process that enables the same prior to work universally for Bayesian optimization on functions with different domains. We propose a two-step pre-training method and demonstrate its empirical success on challenging black-box function optimization problems with varied input dimensions and search spaces.

1 Introduction

While Bayesian optimization (BO) with Gaussian process (GP) priors has been shown to be effective for expensive black-box function optimization, it is often difficult to hand-specify a good Bayesian prior [6, 3, 7]. In this work, we propose HyperBO+ that addresses the issue of prior specification by pre-training a hierarchical GP on collections of function observations partitioned by relevance. Notably, we only need to pre-train the hierarchical GP once and we can use it universally for BO on functions with different domains.

Our work is inspired by HyperBO [5], which shows pre-training helps learning a better GP prior and improving BO performance on the same domain over different functions. Similar ideas [7, 3, 6, 4] were also proposed for transfer learning and meta learning in Bayesian optimization but they can only transfer the knowledge for functions with the same domain.

Recently, Chen et al. [1] proposed OptFormer, a transformer based framework of transfer learning for hyperparameter tuning on universal search spaces, a goal shared by our work. However, due to its core idea of proposing hyperparameters in an end-to-end fashion, OptFormer must be trained on millions of BO trajectories, requires giant transformer models, demands expensive TPU hardware and works only with continuous domains. On the contrary, our model does not have any constrains on how the data is collected, uses compact GP models with a small number of parameters, trains with much shorter time on CPUs and works with both continuous and discrete domains.

Our contributions include (1) a new pre-training method for hierarchical GPs on functions with different domains; (2) HyperBO+, a noval transfer learning BO method that generalizes to universal search spaces; and (3) showing the empirical success of HyperBO+ on challenging BO tasks.

2 Problem Formulation

Our goal is to optimize unseen black-box functions by pre-training on existing data from functions in multiple search spaces, which can have different numbers of dimensions for their respective domains.

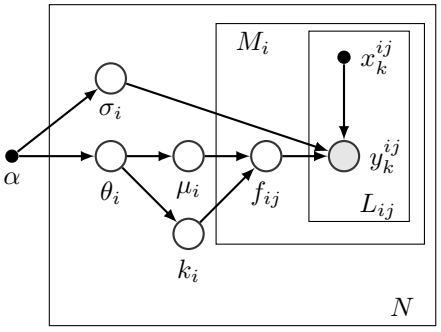
2022 NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems.

We use the term *super-dataset* to denote all datapoints collected across multiple search spaces, while a *dataset* includes the data from a single search space, and a *sub-dataset* means the collection of datapoints from a single function within a search space.

More formally, we define a super-dataset as $D = \{D_i\}_{i=1}^N$. Each dataset D_i consists of observations on a collection of black-box functions $F_i = \{f_{ij} : \mathcal{X}_i \to \mathbb{R}\}_{j=1}^{M_i}$ where functions in F_i share the same compact search space $\mathcal{X}_i \in \mathbb{R}^{d_i}$. Let $D_i = \{D_{ij}\}_{j=1}^{M_i}$, where each sub-dataset $D_{ij} = \{(x_k^{ij}, y_k^{ij})\}_{k=1}^{L_{ij}}$. L_{ij} is the number of observations on function f_{ij} perturbed by *i.i.d.* additive Gaussian noise, i.e. $y_k^{ij} \sim \mathcal{N}(f_{ij}(x_k^{ij}), \sigma_i^2)$. For each $i = 1, ..., N$, we assume all functions in F_i are *i.i.d.* function samples from the same GP: $\mathcal{GP}_i = \mathcal{GP}(\mu_i, k_i)$.

For each function set F_i and its corresponding $\mathcal{GP}_i$ with mean function $\mu_i : \mathcal{X}_i \to \mathbb{R}$ and kernel $k_i : \mathcal{X}_i \times \mathcal{X}_i \to \mathbb{R}$, we denote the parameters of μ_i, k_i by θ_i. Our major assumption is that the GP parameters and noise variances $\{(\theta_i, \sigma_i)\}_{i=1}^N$ are *i.i.d.* samples from a distribution, as in $(\theta_i, \sigma_i) \sim p((\theta, \sigma); \alpha)$, which is parameterized by α. The distribution $p((\theta, \sigma); \alpha)$ is a *universal prior* for all search spaces. Fig. 1 illustrates the graphical model of our hierarchical GP.

Figure 1: Graphical model for a hierarchical Gaussian process.



For simplicity, we use a length-scale based kernel, e.g. Matern kernel, and assume that each dimension of the GP parameters and the noise variance are all independent. We further assume that all length-scale parameters share the same prior distribution. Thus, we can estimate the distribution $p(\cdot; \alpha)$ using the super-dataset and this learned universal prior can be used to guide BO on new black-box functions, either from seen search spaces or from unseen search spaces.

Evaluation metrics. For an unseen blackbox "testing function" f, we run BO on the function for T steps and accumulate the observations $\{x_t, y_t\}$. We evaluate the optimization performance of a given method on an individual testing function using the *simple regret* $r_T = \max_{x \in \mathcal{X}} f(x) - \max_{t \in [T]} y_t$. We use the *mean of simple regret* on all testing functions as the overall evaluation metric.
Additionally, we also report the *negative log-likelihood (NLL)* of the compared models (either a hierarchical GP or a vanilla GP) on the training and testing datasets. For a vanilla GP parameterized by (θ, σ), the NLL on a dataset D is defined as $\mathrm{NLL}(\theta, \sigma) = -\log p(D|\theta, \sigma)$. For a hierarchical GP parameterized by α, the NLL is defined as follows and is computed via sampling in practice.

$$\mathrm{NLL}(\alpha) = -\log p(D|\alpha) = -\log \left(\int_{(\theta,\sigma)} p(D|\theta, \sigma) p((\theta, \sigma); \alpha) \mathrm{d}(\theta, \sigma) \right) \tag{1}$$

$$\approx -\log \left(\frac{1}{Q} \sum_{q=1}^{Q} p(D|\theta_q, \sigma_q) p((\theta_q, \sigma_q); \alpha) \right) \tag{2}$$

where $\{(\theta_q, \sigma_q)\}_{q=1}^Q$ are *i.i.d.* samples from $p((\theta, \sigma); \alpha)$.

3 Methodology

The HyperBO+ framework we propose consists of mainly two phases: (1) Training: estimate the universal prior α from the super-dataset $D = \{D_i\}_{i=1}^N$ with a two-step approach. (2) Optimization: running BO with the hierarchical GP parameterized by the learned α on testing functions.

3.1 Two-step training

Estimating GP parameters of each search space. For each function collection F_i with domain $\mathcal{X}_i$, we can infer its GP parameters θ_i and noise variance σ_i by minimizing the negative log-likelihood of the dataset as in the original HyperBO [5]. Specifically, we use L-BFGS [2] algorithm to minimize

the NLL loss function

$$\text{NLL}(\theta, \sigma, D_i) = -\log p(D_i|\theta, \sigma) = -\sum_{j=1}^{M_i} \log p(D_{ij}|\theta, \sigma) \tag{3}$$

to get the estimate $(\hat{\theta}_i, \hat{\sigma}_i) = \arg\min_{\theta,\sigma} \text{NLL}(\theta, \sigma, D_i)$.

Estimate the universal prior. Using the estimated $\{(\hat{\theta}_i, \hat{\sigma}_i)\}_{i=1}^N$ from all datasets, we can use the the maximum likelihood estimator (MLE) for the universal prior parameter α as $\hat{\alpha} = \arg\max_\alpha p(\{(\hat{\theta}_i, \hat{\sigma}_i)\}_{i=1}^N; \alpha)$. For instance, we can choose the universal prior to be a normal distribution, i.e.

$$p(\{(\hat{\theta}_i, \hat{\sigma}_i)\}_{i=1}^N; \alpha) = \mathcal{N}(\alpha_0, \mathbf{I}\alpha_1)$$

where $\alpha = [\alpha_0, \alpha_1] \in \mathbb{R}^2$, and directly estimate the mean and variance as the parameter α. For lengthscales, we treat all estimated lengthscales as *i.i.d.* samples from the same distribution.

Algorithm 1 HyperBO+ training and optimization with acquisition function $ac(\cdot)$.

function HYPERBO+ (f, D)
  for $D_i \in D$ **do**
    $\hat{\theta}_i, \hat{\sigma}_i \leftarrow \text{TRAIN}(D_i)$
  end for
  $\hat{\alpha} \leftarrow \text{MLE}(\{\hat{\theta}_i, \hat{\sigma}_i\}_{i=1}^N)$
  $D_f \leftarrow \emptyset$
  for $t = 1, \cdots, T$ **do**
    $x_t \leftarrow \arg\max_{x \in \mathfrak{X}} ac_t(x; \hat{\alpha}))$ (Eq. 4)
    $y_t \leftarrow \text{OBSERVE}(f(x_t))$
    $D_f \leftarrow D_f \cup \{(x_t, y_t)\}$
  end for
  return D_f
end function

3.2 Bayesian Optimization

HyperBO+ models functions via a hierarchical GP with pre-trained universal prior parameter $\hat{\alpha}$. At step t of Bayesian optimization when optimizing a testing function f, we compute the acquisition function for HyperBO+ by first computing the posterior distribution of GP parameters and noise variance (θ, σ) themselves, then taking the average of acquisition function values according to (θ, σ)-s sampled from this posterior distribution:

$$ac_t(x; \hat{\alpha}) = \sum_{r=1}^{R} \left[ac_t(x; \theta_r, \sigma_r) p((x_k, y_k)_{k=1}^t | \theta_r, \sigma_r)\right] \tag{4}$$

where $(\theta_1, \sigma_1), \ldots, (\theta_R, \sigma_R)$ are *i.i.d.* samples from the prior distribution $p((\theta, \sigma); \hat{\alpha})$. By injecting the term $p((x_k, y_k)_{k=1}^t | \theta_r, \sigma_r)$ to the acquisition function value of each sample, we convert sampling from the prior distribution to sampling from the posterior distribution $p((\theta, \sigma)|(x_k, y_k)_{k=1}^t; \hat{\alpha})$ according to Bayes' Rule $p((\theta, \sigma)|(x_k, y_k)_{k=1}^t; \hat{\alpha}) = p((x_k, y_k)_{k=1}^t | \theta_r, \sigma_r) p((\theta_r, \sigma_r); \hat{\alpha}) / p((x_k, y_k)_{k=1}^t)$. Here we can ignore the denominator $p((x_k, y_k)_{k=1}^t)$ since the BO selection is invariant under any positive affine transformation of the acquisition function. Note that posterior sampling could be a preferred option than re-weighting with likelihood. We find our re-weighting strategy to be sufficient since the samples from the pre-trained prior are often representative of the samples from the posterior.

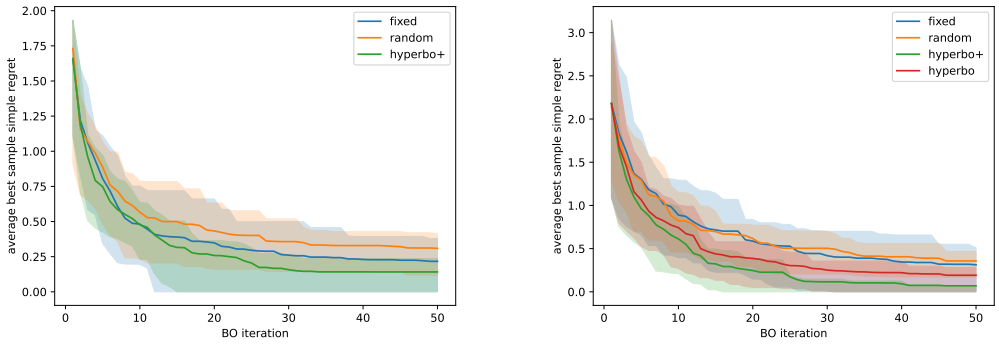
4 Experiments

We generate a synthetic super-dataset with multiple search spaces following the graphical model in Fig. 1. The super-dataset includes 20 datasets (search spaces) with 10 sub-datasets in each dataset. Each sub-dataset includes noisy observations at 300 input locations in its respective search space. The dimensions of search spaces are between 2 and 5.

Experiment Setups. We consider 3 baselines: (1) A fixed hierarchical GP with a misspecified prior. (2) Random sampling for optimization. (3) HyperBO [5]. Note that HyperBO is not applicable to training on some search spaces and testing on others (Setup A).

Setup A: We split the super-dataset into 16 training datasets and 4 testing datasets and report the aggregated BO performance on test datasets. This setup demonstrate the ability of HyperBO+ to transfer learned knowledge to unseen search spaces.

Setup B: We split each dataset into into 8 training sub-datasets and 2 testing sub-datasets and report BO performance on testing sub-datasets from all datasets. Compared with the original HyperBO which fits GP parameter for each dataset individually, HyperBO+ learns a universal prior for hierarchical GP from the collection of training sub-datasets from all datasets.



(a) Setup A

(b) Setup B

Figure 2: Average simple regret of each method across test sub-datasets during BO with GP-UCB acquisition function ('fixed' stands for the fixed misspecified hierarchical GP). The highlighted areas show the 25th and 75th percentile of simple regret for each method.

Analyses on BO results. The regret curves of compared methods during BO in Setup A are shown in Fig. 2a. We can see that HyperBO+ outperforms the fixed misspecified hierarchical GP baseline and the random baseline, demonstrating its capability of learning knowledge about the universal prior from training datasets and generalizing to new search spaces that are unseen during training.

Fig. 2b shows the regret curves for Setup B. Besides the same trend we have seen in Setup A, HyperBO+ also outperforms HyperBO, which is rather surprising given that HyperBO specifically pre-trains a GP that targets a single search space. In contrast, HyperBO+ learns one universal prior parameterized by α on training sub-datasets from all search spaces and would need to automatically capture the characteristics of any testing function through posterior inference of GP parameters based on function observations. One possible reason for this is that HyperBO overfits on the training sub-datasets which damages its generalization performance on the testing sub-datasets from the same search space, but further experiments are needed to verify this conjecture.

The results shown here are under one of a couple of groundtruth prior distributions we have tried to generate the synthetic super-dataset. Results for more prior distributions and more acquisition function types, as well as discussion on those additional results are included in the appendix.

Evaluations of NLLs. We also evaluated the NLLs of the compared GP-based methods on training and testing data in both Setup A and B. For both training and testing data, NLLs are computed on every sub-dataset and the average is reported. In Setup A, the misspecified hierarchical GP gets the NLLs of 210.78 and 81.64, and HyperBO+ gets 64.23 and -69.20, on training and testing data respectively. In Setup B, the misspecified hierarchical GP gets the NLLs of 187.54 and 188.27, HyperBO+ gets 38.17 and 38.63, and the original HyperBO gets 25.40 and 29.47, on training and testing data respectively.

It can be observed that HyperBO+ achieves much lower NLLs than the misspecified hierarchical GP on both training and testing data in both setups, which demonstrates the effectiveness of its training. The original HyperBO gets even lower NLLs than HyperBO+ in Setup B, which makes sense as HyperBO specifically pre-trains a GP that targets a single search space.

5 Conclusions

In this paper, we propose HyperBO+, a method that learns a universal prior of hierarchical GP from observations collected on different black-box functions across different search spaces. We show empirically on synthetic datasets that HyperBO+ outperforms traditional methods and shows capability of generalizing to new functions from either seen search spaces or unseen search spaces. It is worth noting that the synthetic dataset is constructed under our modeling assumptions, which provides an ideal condition for HyperBO+ to learn the desired universal prior. We plan to test the performance of HyperBO+ on datasets from realistic black-box optimization tasks in future work.

Acknowledgments and Disclosure of Funding

We thank Jasper Snoek and Eytan Bakshy for helpful conversations and feedback. Our work also benefited from Microsoft Azure credits provided by the Harvard Data Science Initiative.

References

[1] Yutian Chen, Xingyou Song, Chansoo Lee, Zi Wang, Qiuyi Zhang, David Dohan, Kazuya Kawakami, Greg Kochanski, Arnaud Doucet, Marc'aurelio Ranzato, et al. Towards learning universal hyperparameter optimizers with transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[2] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

[3] Valerio Perrone, Rodolphe Jenatton, Matthias Seeger, and Cédric Archambeau. Scalable hyperparameter transfer learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6846–6856, 2018.

[4] Michael Volpp, Lukas P Fröhlich, Kirsten Fischer, Andreas Doerr, Stefan Falkner, Frank Hutter, and Christian Daniel. Meta-learning acquisition functions for transfer learning in Bayesian optimization. In *International Conference on Learning Representations (ICLR)*, 2020.

[5] Zi Wang, George E Dahl, Kevin Swersky, Chansoo Lee, Zelda Mariet, Zachary Nado, Justin Gilmer, Jasper Snoek, and Zoubin Ghahramani. Pre-trained Gaussian processes for Bayesian optimization. *arXiv preprint arXiv:2109.08215*, 2022.

[6] Zi Wang, Beomjoon Kim, and Leslie Pack Kaelbling. Regret bounds for meta Bayesian optimization with an unknown Gaussian process prior. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[7] Martin Wistuba and Josif Grabocka. Few-shot Bayesian optimization with deep kernel surrogates. In *International Conference on Learning Representations (ICLR)*, 2021.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix A.1.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In Figure 2, we report the variance of regret-iteration plots with respect to running BO on different sub-datasets functions from the testing data and those functions are random samples during the synthetic data generation.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Appendix A.1.
   (b) Did you mention the license of the assets? [Yes] Our shared code repository includes license description from the existing codebase we use.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We shared our code repsitory in Appendix A.1.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

A.1 More Details about Experiment Setup

Our code for the experiments is built upon the codebase of HyperBO [5], and is available at `https://github.com/Evensgn/hyperbo`.

The synthetic super-dataset we use includes 20 datasets (search spaces) with 10 sub-datasets in each dataset. Each sub-dataset includes noisy observations at 300 input locations in its respective search space. The dimensions of the search spaces are randomly sampled between 2 and 5. The train/test splits in the two setups are as described in the main body of the paper. As all the datasets are generated to be *i.i.d.* samples, we use the first 16 datasets to be training datasets and the remaining 4 datasets to be testing datasets in Setup A. Similarly, we use the first 8 sub-datasets of each dataset to form the training data and use the remaining 2 sub-datasets of each dataset to form the testing data in Setup B.

We used the Matern32 kernel and constant meen function for all GPs. Each dataset D_i ($i \in \{1, \ldots, N\}$) corresponds to $\mathcal{GP}_i$ parameterized by θ_i, which includes the following parameters: constant (the value of the constant mean function), length-scale (which has the same number of dimensions as the search space), signal variance. In addition, D_i is also parameterized by the noise variance σ_i.

For the distribution classes for each of these parameters, we use a Normal distribution for constant and use a Gamma distribution for each of the remaining parameters. HyperBO+ is given the distribution classes used for the synthetic dataset generation but not the distribution parameters (μ and σ for Normal distribution, α and β for Gamma distribution). The specific distribution parameters we use to generate the synthetic dataset for the experiment section presented in the main body of the paper are as follows: constant is sampled from $\mathrm{Normal}(\mu = 1, \sigma = 1)$, each dimension of length-scale is sampled

from $\text{Gamma}(\alpha = 10, \beta = 30)$, signal variance is sampled from $\text{Gamma}(\alpha = 1, \beta = 1)$, and noise variance is sampled from $\text{Gamma}(\alpha = 10, \beta = 100000)$. Additional distribution parameter values of synthetic data generation we have tried and additional experiment results are shown below in Appendix A.2.

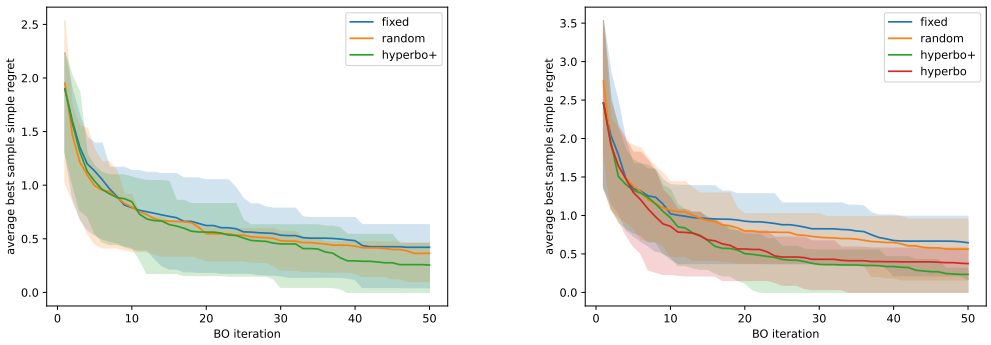
For the fixed misspecified hierarchical GP, the specific parameters are as follows: constant is sampled from $\text{Normal}(\mu = 0, \sigma = 1)$, each dimension of length-scale is sampled from $\text{Gamma}(\alpha = 1, \beta = 10)$, signal variance is sampled from $\text{Gamma}(\alpha = 1, \beta = 5)$, and noise variance is sampled from $\text{Gamma}(\alpha = 10, \beta = 100)$.

We use LBFGS optimizer to fit the GP parameters of every dataset for the training of HyperBO+ and HyperBO, while the maximum number of iterations is set to 500. For running BO on the testing data, we use the GP-UCB acquisition function (results with more acquisition function types are shown in Appendix A.2) and the optimization budget is $T = 50$ iterations. At each step of BO iteration, we sample $R = 100$ pairs of (θ, δ) values to compute the acquisition function values in Equation 4. When evaluating the NLLs of hierarchical GP-based methods (the misspecified hierarchical GP and HyperBO+), we sample $Q = 500$ pairs (θ, δ) values to estimate the NLL as in Equation 1.

One experiment (one synthetic super-dataset, one acquisition function type) under the current setting takes around 6 hours to finish on a virtual machine with 96 vCPUs.

A.2 Additional Experiment Results

As described above, the experiments results in Section 4 corresponds to a configuration where the groundtruth prior distribution of the length-scale parameter is $\text{Gamma}(\alpha = 10, \beta = 30)$ and the acquisition function type for BO is GP-UCB.

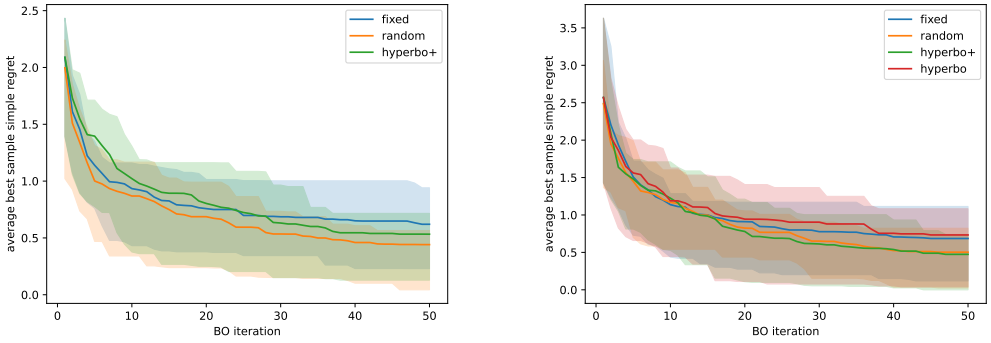


(a) Setup A (b) Setup B

Figure 3: Average simple regret of each method across test sub-datasets during BO with GP-UCB acquisition function ('fixed' stands for the fixed misspecified hierarchical GP). The highlighted areas show the 25th and 75th percentile of simple regret for each method. The groundtruth prior distribution of the length-scale parameter is $\text{Gamma}(\alpha = 2, \beta = 6)$.

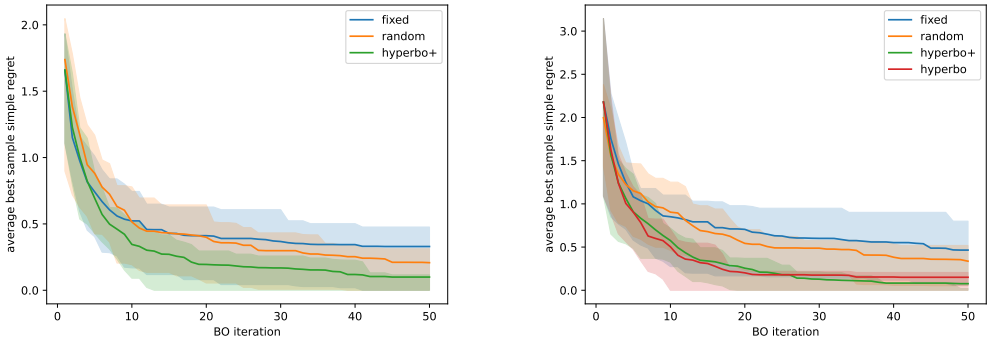
More groundtruth prior for length-scale. As the difficulty of a black-box function optimization task is particularly sensitive to the length-scale parameter, we also explored a couple of other prior distribution parameters for length-scale, while keeping the groundtruth prior distribution for other GP parameters unchanged. Figure 3 shows the results when the groundtruth prior distribution of the length-scale parameter is $\text{Gamma}(\alpha = 2, \beta = 6)$ and the acquisition function type for BO is GP-UCB. Figure 4 shows the results when the groundtruth prior distribution of the length-scale parameter is $\text{Gamma}(\alpha = 1, \beta = 5)$ and the acquisition function type for BO is GP-UCB.

More acquisition function types. In addition, we also run experiments with two other acquisition functions while setting the groundtruth prior distribution of the length-scale parameter as $\text{Gamma}(\alpha = 10, \beta = 30)$. Figure 5 shows the results when the acquisition function type for BO is *Expected Improvement* (EI). Figure 6 shows the results when the acquisition function type for BO is *Probability of Improvement* (PI).



(a) Setup A (b) Setup B

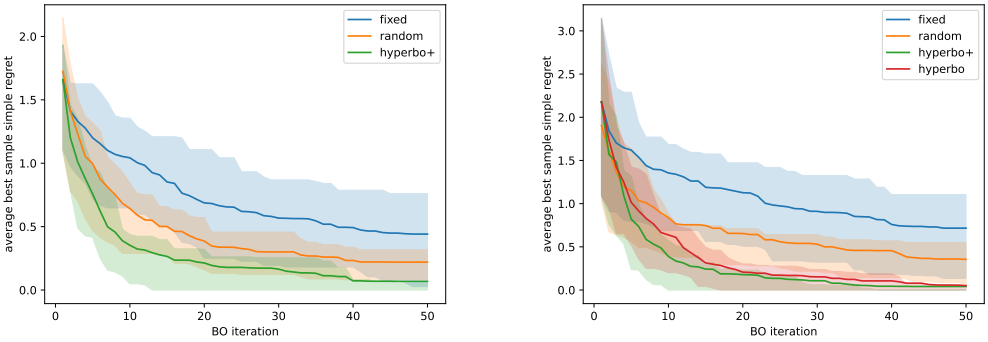
Figure 4: Average simple regret of each method across test sub-datasets during BO with GP-UCB acquisition function ('fixed' stands for the fixed misspecified hierarchical GP). The highlighted areas show the 25th and 75th percentile of simple regret for each method. The groundtruth prior distribution of the length-scale parameter is $\mathrm{Gamma}(\alpha = 1, \beta = 5)$.



(a) Setup A (b) Setup B

Figure 5: Average simple regret of each method across test sub-datasets during BO with EI acquisition function ('fixed' stands for the fixed misspecified hierarchical GP). The highlighted areas show the 25th and 75th percentile of simple regret for each method. The groundtruth prior distribution of the length-scale parameter is $\mathrm{Gamma}(\alpha = 10, \beta = 30)$.

As we can see from these results, the performance of GP-based methods are sometimes no better than or even worse than the random baseline, this is possible when the length-scale parameter is too small (the function is too volatile within the domain) or too large (the function is very smooth and therefore easy for random baseline to optimize). The two additional groundtruth prior distributions for length-scale shown here has a larger variance compared to $\mathrm{Gamma}(\alpha = 10, \beta = 30)$, the one used in the main body of the paper, thus are more likely to generate datasets with length-scales that are too small or too large. The relative ordering of performance among the compared methods can also change when a different acquisition function is used. But one common observation from the different configurations is that HyperBO+ outperforms the misspecified hierarchical GP and either outperforms or achieves similar performance with original HyperBO in all of these configurations.



(a) Setup A

(b) Setup B

Figure 6: Average simple regret of each method across test sub-datasets during BO with PI acquisition function ('fixed' stands for the fixed misspecified hierarchical GP). The highlighted areas show the 25th and 75th percentile of simple regret for each method. The groundtruth prior distribution of the length-scale parameter is $\mathrm{Gamma}(\alpha = 10, \beta = 30)$.