TRANSFER LEARNING FOR BAYESIAN HPO WITH END-TO-END META-FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Hyperparameter optimization (HPO) is a crucial component of deploying machine learning models, however, it remains an open problem due to the resource-constrained number of possible hyperparameter evaluations. As a result, prior work focuses on exploring the direction of transfer learning for tackling the sample inefficiency of HPO. In contrast to existing approaches, we propose a novel **D**eep **K**ernel Gaussian Process surrogate with **L**andmark **M**eta-features (DKLM) that can be jointly meta-trained on a set of source tasks and then transferred efficiently on a new (unseen) target task. We design DKLM to capture the similarity between hyperparameter configurations with an end-to-end meta-feature network that embeds the set of evaluated configurations and their respective performance. As a result, our novel DKLM can learn contextualized dataset-specific similarity representations for hyperparameter configurations. We experimentally validate the performance of DKLM in a wide range of HPO meta-datasets from OpenML and demonstrate the empirical superiority of our method against a series of state-of-the-art baselines.

1 INTRODUCTION

Hyperparameter optimization (HPO) is an essential open problem in machine learning (ML) due to the black-box nature of methods' empirical performances as a function of their hyperparameters. The major challenge lies in the computational infeasibility of training and evaluating a large sample of hyperparameters in order to identify the best generalization performances. As a result, transfer learning lends itself as a promising direction for improving the sample efficiency of HPO methods (Wistuba et al., 2016; Perrone et al., 2018; Wistuba & Grabocka, 2021).

Prior approaches for transfer learning in HPO rely on exploring existing evaluations on a pool of datasets where the model under investigation is evaluated. The similarity between datasets is often captured via features describing their characteristics (a.k.a. meta-features), such as descriptive statistics of dataset features (Michie et al., 1994; Wistuba et al., 2016), or landmark measures in the form of the accuracies gathered from a set of basic classifiers (nearest neighbors, decision trees, SVMs, etc.) on the datasets (Pfahringer et al., 2000; Feurer et al., 2014). A recent trend highlights the potential of learning parametric meta-feature extractors for tabular datasets (Jomaa et al., 2021a), which are further meta-trained (Finn et al., 2017) to improve the HPO transferability to new datasets (Jomaa et al., 2021b).

Unfortunately, typical transfer learning from a set of unrelated source tasks suffers from the negative transfer phenomenon (Wang et al., 2019), which implies a poor generalization performance on target tasks that are dissimilar to the source tasks, according to a predefined dissimilarity measure, e.g. similarity of response curves. This can happen, for example, when a model is learned jointly across tasks without task-specific attributes. To resolve the negative transfer of HPO performance predictors (a.k.a. surrogates) we introduce a novel direction that conditions Gaussian Process (GP) surrogates in Bayesian Optimization on the meta-features of datasets. In that manner, we can transfer knowledge only from similar datasets, and hence conditioned on the similarity of meta-features. However, in contrast to ad-hoc dataset meta-features that are hand-crafted by domain experts, we propose a novel architecture for deep GP kernels (Wilson et al., 2016a) that are enriched with novel end-to-end neural network components that generate meta-features only from the tuples of past hyperparameter configurations and their evaluated performances. Our meta-feature network

is a set-based neural network that is invariant to the permutation/sequence of past hyperparameter evaluations.

We jointly train **D**eep **K**ernel Gaussian Process surrogate with **L**andmark **M**eta-features (DKLM) through HPO meta-learning (Wistuba & Grabocka, 2021). To validate the empirical performance of our method we present extensive results on a large-scale benchmark that involves 16 different search spaces and 101 datasets from OpenML for a total of 3.4 million hyperparameter evaluations (Pineda-Arango et al., 2021). Detailed experiments against a series of traditional HPO methods, as well as recent transfer HPO baselines, demonstrate the superiority of meta-learning the initialization of DKLM. Overall, we make the following contributions:

- Introduce the first paper that tackles the negative transfer phenomenon in Bayesian HPO, by conditioning GP surrogates on meta-features, i.e. on dataset characteristics;
- Propose an end-to-end deep GP which implicitly learns networks that generate meta-features, with no ad-hoc inductive bias from experts on manually designing meta-features;
- Demonstrate the empirical superiority of our method on a very-large-scale experimental protocol (3.4 million hyperparameter evaluations), against a large number of baselines.

2 Related Work

Hyperparameter optimization (HPO) has been extensively studied over the past decade for improving the performance of machine learning models beyond simple search techniques (Larochelle et al., 2007; Bergstra & Bengio, 2012). Non-transfer learning solutions often define a probabilistic surrogate that estimates the true hyperparameter response surface using Gaussian Processes (Rasmussen, 2003), Bayesian Neural Networks (Snoek et al., 2015; Springenberg et al., 2016), or tree-based models (Hutter et al., 2011; Bergstra et al., 2011). Hyperparameters are then selected via an acquisition function (Wilson et al., 2017) and the process is reiterated with the new set of observations until a specified budget is exhausted, e.g. runtime or number of trials.

HPO is further expedited when defined within the context of transfer learning, i.e. by leveraging related tasks (or datasets) to improve the generalization over unseen tasks. Transfer learning for HPO has been observed by modeling tasks jointly (Swersky et al., 2013; Yogatama & Mann, 2014; Perrone et al., 2018; Salinas et al., 2020), or through some weighted-combination of the surrogates (Schilling et al., 2016; Wistuba et al., 2016; Feurer et al., 2018). Other directions include pruning the hyperparameter search space (Wistuba et al., 2015a; Perrone & Shen, 2019), or learning to initialize the surrogate by identifying good initial hyperparameters (Wistuba et al., 2015b). Apart from learning a transferable surrogate, recently, transferable acquisition functions (Wistuba et al., 2018; Volpp et al., 2020) have also been proposed to replace engineered acquisition functions. The success of meta-learning for domain adaptation has also been investigated for HPO. Wistuba & Grabocka (2021) explore few-shot Bayesian Optimization by learning a deep kernel Gaussian Process surrogate across a set of tasks to quickly adapt to new a target task. Similarly, Jomaa et al. (2021b) learn a shared neural network surrogate jointly coupled with a meta-feature extractor defined over the dataset itself.

Meta-features (Vanschoren, 2018), or dataset characteristics, have also been widely adopted in HPO algorithms for warm-start initialization (Feurer et al., 2015; Wistuba et al., 2016) or as additional attributes to better marginalize the surrogate on individual tasks (Bardenet et al., 2013). Nevertheless, extracting meta-features requires direct access to the datasets, which might be difficult in real settings where only the meta-dataset is available. In this paper, we propose to extract landmark meta-features from existing evaluations (Leite et al., 2012; Sun & Pfahringer, 2013) in an end-to-end fashion using a deep Gaussian kernel approach.

3 Preliminaries

3.1 Hyperparameter Optimization

We denote by $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ a task of interest, such that $x_i \in \mathcal{X} \subseteq \mathbb{R}^n$ represents a hyperparameter configuration in the domain of a (bounded) hyperparameter search space for some model

under investigation. Furthermore, let $y_i = f(x_i) + \epsilon$ be the response of an unknown black-box function $f := \mathcal{X} \to \mathbb{R}^+$, with ϵ as an additive i.i.d Gaussian noise with some homoscedastic variance σ^2. Typically, y represents a metric of interest that should be optimized to obtain better model generalization, e.g. validation loss. The objective of hyperparameter optimization is then to find the optimal hyperparameter such that $x_* = \arg\min_{x \in \mathcal{X}} f(x)$ given a fixed budget T of trials. HPO is commonly treated as sequential decision-making process, where a surrogate model $\hat{y} : \mathcal{X} \to \mathbb{R}$ is iteratively fit to the history $\mathcal{H}_t := \{(x_i, y_i)\}_{i=1}^t$ of evaluated hyperparameters and a policy (or acquisition function) $\mathcal{A} : (\mathcal{X} \times \mathbb{R})^* \to \mathcal{X}$ is used to select the next candidate which minimizes the expected hyperparameter response. Among the existing acquisition functions, *expected improvement* is widely adopted (Mockus, 1974).

3.2 Deep Kernel Gaussian Processes

Given a training task $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, the response can be modeled using a Gaussian process (GP), i.e. as a multivariate Gaussian distribution, such that $y \sim \mathcal{N}(m(X), k(X, X))$. A GP is a non-parametric approach that defines a prior over functions directly, and is defined by its mean function, m, and kernel function k. Given some observed data points, it is possible to compute the posterior over these functions to approximate unobserved data points as

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(m(X), \begin{pmatrix} K_n & K_* \\ K_*^T & K_{**} \end{pmatrix}\right) \tag{1}$$

with $K_n = k(X, X \mid \theta) + \sigma_n^2 \mathbb{I}$, $K_* = k(X, X_* \mid \theta)$, and $K_{**} = k(X_*, X_* \mid \theta)$. The mean and covariance of the posterior predictive distribution is then estimated as

$$\mathbb{E}[f_* \mid X, y, X_*] = K_*^T K_n^{-1} y, \ \text{cov}[f_* \mid X, X_*] = K_{**} - K_*^T K_n^{-1} K_* \tag{2}$$

The standard approach of fitting GPs is to optimize the weights of the kernel function, e.g. squared exponential kernel, θ. Nevertheless, these engineered kernels are often employed under false assumptions (Cowen-Rivers et al., 2020), which leads to sub-optimal performances.

Recently deep kernel learning (Wilson et al., 2016b) has emerged as a powerful extension that leverages the representative capacity of non-linear function approximation, e.g. neural networks, and facilitates learning the kernel directly. Specifically, we denote by $\phi : \mathcal{X} \to \mathbb{R}^N$ a mapping from the domain to a latent space which serves as an input to the kernel, such that:

$$K_{\text{deep}}(x, x' \mid \theta, w) = K(\phi(x, w), \phi(x', w) \mid \theta) \tag{3}$$

where w represents the parameters of ϕ. The weights θ and w are then jointly optimized for maximizing the marginal likelihood (Wistuba & Grabocka, 2021).

4 Deep Kernel Gaussian Process with Landmark Meta-features

Inspired by landmark meta-features (Pfahringer et al., 2000), which are typically estimated by measuring the response of given datasets to machine learning algorithms, we propose a novel deep kernel GP that is conditioned on task-specific landmark meta-features. However, instead of computing meta-features through ad-hoc approaches, we introduce a novel parametric meta-feature extractor network that is integrated into the kernel function of a GP and subsequently meta-learned over a set of source tasks together with the parameters of the GP kernel. In that manner, we learn meta-features that describe a set of tasks in terms of minimizing the estimation of the tasks' response functions. By adding the task-specific information of the meta-features, the GP surrogate can infer a more accurate response surface on a new task based on similar source tasks that share similar meta-features. Therefore, our method is the first to tackle the negative transfer phenomenon for Bayesian HPO.

4.1 Landmark Meta-Feature Networks

We propose a simple idea to learn landmark meta-features by learning a deep representation of the evaluated **set** of hyperparameter-response pairs as part of a deep kernel Gaussian process. With the success of set-based algorithms (Lee et al., 2020; Zaheer et al., 2017; Lee et al., 2019) for function

approximation, we propose to use a *Deepset* (Zaheer et al., 2017) formulation that provides a fixed-size vector representation from the dynamic set of observations. Although other methods have been developed for set-based function estimation, we focus here on deepsets because they have already been shown to perform well for learning meta-feature in task-agnostic settings (Jomaa et al., 2021a) as well as for hyperparameter optimization (Jomaa et al., 2021b).

Suppose that we are given a collection of data points $\{(x_i, y_i)\}_{i=1}^{n}$ where $x \in \mathcal{X}$ is an observed hyperparameter and $y \in \mathbb{R}$ its corresponding response. We denote by $\mathcal{H}^{t-1} := \{(x_i, y_i)\}_{i=1}^{t-1}$ an associated set of data points that have been observed prior to x_t. Furthermore, we formulate the proposed meta-feature network as:

$$\phi(x, \mathcal{H}^{t-1}, w) = \phi_1\left([x, \phi_2(\mathcal{H}^{t-1}; w_{\phi_2})];\ w_{\phi_1}\right), \tag{4}$$

$$\text{s.t. } \phi_2(\mathcal{H}^{t-1}; w_{\phi_2}) = g\left(\frac{1}{t-1}\sum_{i=1}^{t-1} f\left([x_i, y_i];\ w_f\right);\ w_g\right) \tag{5}$$

where [ ] symbolizes standard concatenation, $\phi_2 : (\mathcal{X} \times \mathbb{R})^* \to \mathbb{R}^N$ and $\phi_1 : \mathcal{X} \times \mathbb{R}^N \to \mathbb{R}^M$ are parametric neural networks with respective weights w, and where $(\mathcal{X} \times \mathbb{R})^*$ represents the set of evaluated hyperparameter and their responses. With this formulation, we ensure that the relationship of the covariates and the responses in $\mathcal{H}^{t-1}$ is properly encoded, and thus ϕ is conditioned on these latent representations. It is also important to note that ϕ_2 is permutation invariant, i.e. $\phi_2(\mathcal{H}^{t-1}) = \phi_2(\pi(\mathcal{H}^{t-1}))$, with $\pi := (\mathcal{X} \times \mathbb{R})^* \to (\mathcal{X} \times \mathbb{R})^*$ as a random permutation function. This is critical, as the ordering of the data points in $\mathcal{H}^{t-1}$ should not affect the landmark meta-features. Additionally, given $\phi_2(\mathcal{H}^{t-1})$, this information about the marginal distribution of the meta-features can be encoded with the specific attribute x which in turn allows the deep kernel GP to be marginalized over individual tasks given the context, and thus transfer (joint) learning becomes easier with minimal overhead.

4.2 Meta-learning our Deep GPs

The parameters θ and w are optimized jointly by maximizing the following log marginal likelihood:

$$\underset{\theta, w}{\arg\max}\ \log p\left(\mathbf{y} \mid x, \mathcal{H}; \theta, w\right) = \underset{\theta, w}{\arg\max}\ \mathbb{E}_{d \sim \mathcal{U}(1,\ldots,D)}\ \log p\left(\mathbf{y}_d \mid x_d, \mathcal{H}_d; \theta, w\right) \tag{6}$$

$$\propto \underset{\theta, w}{\arg\min}\ \mathbb{E}_{d \sim \mathcal{U}(1,\ldots,D)}\ \mathbf{y}_d^T K_d^{-1} \mathbf{y}_d + \log \mid K_d \mid \tag{7}$$

$$\text{s.t. } K_{d,t,t'} := K\left(\phi\left(x_{d,t}, \mathcal{H}_d^{t-1}; w\right), \phi\left(x_{d,t'}, \mathcal{H}_d^{t'-1}; w\right); \theta\right)$$

By using established practices (Wistuba & Grabocka, 2021; Patacchiola et al., 2020), we can optimize Equation 7 in terms of w, θ via stochastic gradient descent (SGD), that is proven to maintain convergence guarantees (Chen et al., 2020). We direct the interested reader to the prior work for more details on optimizing the parameters of deep GPs (Wilson et al., 2016b).

Given the diverse number of tasks, which vary in the number of available data points, we propose to jointly learn the shared surrogate via first-order meta-learning (Nichol et al., 2018). Meta-learning has found resounding success in the research community as an initialization scheme (Finn et al., 2017; Wistuba & Grabocka, 2021; Jomaa et al., 2021b), which allows for fast adaption to new domains. Consequently, the meta-trained model resides on a joint minimum across all the source tasks, such that given limited information about the new (unseen) target task, it can converge faster to the new task's local optima. In this direction, we show the pseudocode of our meta-learning optimization in Algorithm 1.

5 Motivation

Meta-features help to model the posterior uncertainty. To motivate our approach, we present an ablation of the effect of our deep GP kernel with meta-features, compared to the same deep GP kernel without meta-features (i.e. Ours vs. FSBO (Wistuba & Grabocka, 2021)). We created a synthetic meta-dataset of $K = 50$ tasks in the form of randomly sampled sinusoidal functions

Algorithm 1: Meta-learning DKLM via REPTILE (Nichol et al., 2018)

Require: training dataset $\mathcal{E}$; kernel parameters θ, network parameters w; learning rate η; inner update steps v; meta-batch size n, batch size b.
while not converged **do**
  $t \sim \mathcal{U}\left([T_{\text{min}}, T_{\text{max}}]\right)$
  $D_1, \ldots, D_n \sim \mathcal{U}\left([1, \ldots, D]\right)$
  for $i = 1$ to n **do**
    Sample $t-1$ data points to form $\mathcal{H}^{t-1} \sim \mathcal{D}_i$
    Sample batch $\mathcal{B} := \{(x_i, y_i)\}_{i=1}^{b} \sim D_i$
    $\theta_i \leftarrow \theta\,; w_i \leftarrow w$
    for $j = 1$ to v **do**
      Define as $\mathcal{L}$ the objective function of Equation 7
      $\theta_i \leftarrow \theta_i + \eta \nabla_\theta \mathcal{L}$
      $w_i \leftarrow w_i + \eta \nabla_w \mathcal{L}$
  Update $\theta \leftarrow \theta + \eta \frac{1}{n} \sum_{i=1}^{n} (\theta_i - \theta)$
  Update $w \leftarrow w + \eta \frac{1}{n} \sum_{i=1}^{n} (w_i - w)$

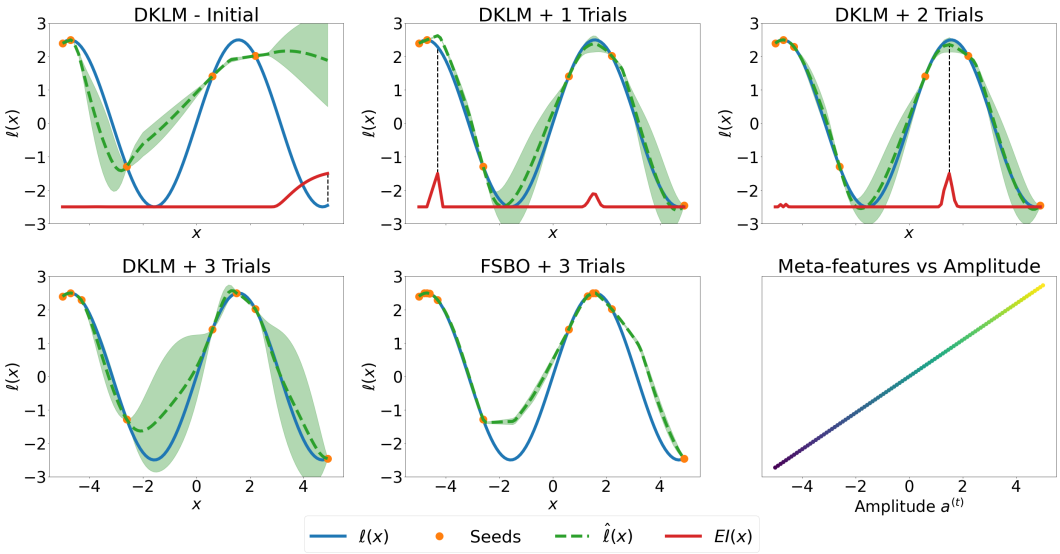

Figure 1: (top) Sequential model-based optimization of an unseen sine wave using our approach. (bottom left and middle) Our fitted surrogate after 3 trials, compared to FSBO after 3 trials given the same initial seeds. (bottom right) Correlation between amplitude and landmark meta-features.

$f^k(x) = a^{(k)} \sin(x + b^{(K)})$, $k \in \{1, \ldots, K\}$ by drawing each $a^{(k)} \sim \mathcal{U}(0.1, 5)$ and $b^{(k)} \sim \mathcal{U}(0, 2\pi)$. Furthermore, we meta-learn our deep GP on these source functions and then transfer the surrogate as an initialization for a new sinusoidal curve (with new parameters a, b) as shown in Figure 1 (top). We show the comparison of our surrogate with meta-features after 3 trials (for a total of 8 data points, including 5 initial configurations) to an FSBO deep GP that has been meta-trained identically. We notice that our surrogate (bottom row, leftmost plot) computes a better posterior variance compared to FSBO (bottom row, middle plot). The effect of the superior modeling of the uncertainty leads to better exploration in a Bayesian Optimization setup, and consequently to better empirical accuracies of the discovered hyperparameters (as will be shown in Section 6).

Meta-features capture task characteristics. We postulated that our proposed meta-features can capture task characteristics. To illustrate the argument, we create another simpler collection of $K = 50$ sinusoidal functions $f^k(x) = a^{(k)} \sin(x)$, $k \in \{1, \ldots, K\}$ by drawing each $a^{(k)} \sim \mathcal{U}(0.1, 5)$. As these sine waves change only in terms of the amplitude parameter a, then if we meta-train our meta-feature network with a 1-dimensional (1D) output layer from these source functions, it must strongly learn to correlate the 1D meta-feature with the sinusoidal amplitude a. As can

be seen in the rightmost plot of the bottom row in Figure 1, this is exactly the case. In this plot, the y-axis shows 1D meta-feature values computed from only 5 random pairs of configurations and responses $(x, f(x))$ from one random task, and the x-axis shows the amplitude of that respective task. Although our meta-feature networks have no design bias in terms of modeling sinusoidal functions, they are perfectly able to extract a latent representation of the amplitude, based on the end-to-end meta-learning of the deep GP for approximating random observations on the sine waves.

6 EXPERIMENTS

Our experimental protocol is designed to primarily address one simple research question: **Do deep GPs with our meta-feature networks outperform state-of-the-art HPO algorithms in the transfer and non-transfer learning settings?**

6.1 META-DATASET AND BASELINES

We evaluate our approach on HPO-B-v3, a new hyperparameter optimization benchmark designed for comparing black-box HPO methods (Pineda-Arango et al., 2021). The benchmark contains a collection of 935 black-box tasks for 16 hyperparameter search spaces (algorithms) evaluated on 101 datasets and divided into predefined training, validation, test splits. Following the same experimental protocol specified at the HPO-B benchmark, we compare our approach to the following large set of 10 HPO baselines:

1. **Random Search** (Bergstra & Bengio, 2012);
2. **GP** (Rasmussen, 2003) is a hyperparameter tuning strategy that relies on a Gaussian Process as a surrogate model with squared exponential kernels (Matern 5/2 kernel) with automatic relevance determination;
3. **DNGO** (Snoek et al., 2015) utilizes a neural network to extract adaptive basis function of hyperparameters, which in turn are fed to a Bayesian linear regression model to generate a posterior distribution;
4. **BOHAMIANN** (Springenberg et al., 2016) is based on Bayesian neural networks that are trained via a stochastic gradient Hamiltonian Monte Carlo;
5. **DGP** (Patacchiola et al., 2020) fits a deep kernel Gaussian process as a surrogate;
6. **TST-R** (Wistuba et al., 2016) is an ensemble approach where the Gaussian process surrogate of the target task is weighted with surrogates of the training datasets based on the ranking similarity of the evaluated hyperparameters;
7. **RGPE** (Feurer et al., 2018) is another ensemble approach, similar to TST-R, which estimates the weights by optimizing a ranking loss between the surrogates of the training datasets and that of the target task;
8. **ABLR** (Perrone et al., 2018) is a multi-task Bayesian linear regression approach that optimizes a shared feature extractor across the training datasets as an initialization strategy for the target task;
9. **GCP+Prior** (Salinas et al., 2020) utilizes a Gaussian Copula process (Wilson & Ghahramani, 2010) trained jointly on the training tasks, where a quantile-transformation is applied on their respective responses. The pretrained process is used as parametric prior for the target dataset;
10. **FSBO** (Wistuba & Grabocka, 2021) uses deep Kernel Gaussian processes (Patacchiola et al., 2020) to estimate the response of the target dataset. The parameters are initialized via meta-learning the joint response surface over the training datasets.

In a nutshell, our experimental protocol based on HPO-B is a large scale one by the standards of the prior papers, as it involves 10 baselines, 16 search spaces (algorithms whose hyperparameters we tune), 101 datasets, and totally 935 black-box tasks containing 6.3 million evaluations.

6.2 Implementation Details

We implement the Deep Kernel Gaussian Process using GPyTorch 1.5 (Gardner et al., 2018) with a Matern 5/2 kernel. As described in Equation 4, DKLM is composed of two modules, $\phi := \phi_1 \circ \phi_2$. The parameters of the network have been selected based on the performance on a held-out validation set. Function f is 4 dense layers with 32 hidden units and ReLU activation functions, whereas g is 1 dense layer with 32 hidden units. Finally, ϕ_1 is 4 dense layers with 32 hidden units and ReLU activation. All parameters of the Deep Kernel are estimated by maximizing the marginal likelihood. We achieve this by using gradient ascent and Adam (Kingma & Ba, 2015) with a learning rate of 0.001 and batch size of 64 with $t \in [2, 100]$.

6.3 HPO Results and Discussion

We start by comparing **non-transfer** HPO methods to our deep GPs with meta-features DKLM (RI) that are randomly initialized (no meta-learning). As depicted in Figure 2, DKLM (RI) outperforms the baselines after 100 hyperparameter trials in terms of both the mean normalized regret and the mean rank metrics. We also notice how the performance improves gradually with the increasing number of trials, indicating the impact of the posterior variance modeling of our method (Section 5) as more observations are present on the black-box responses.

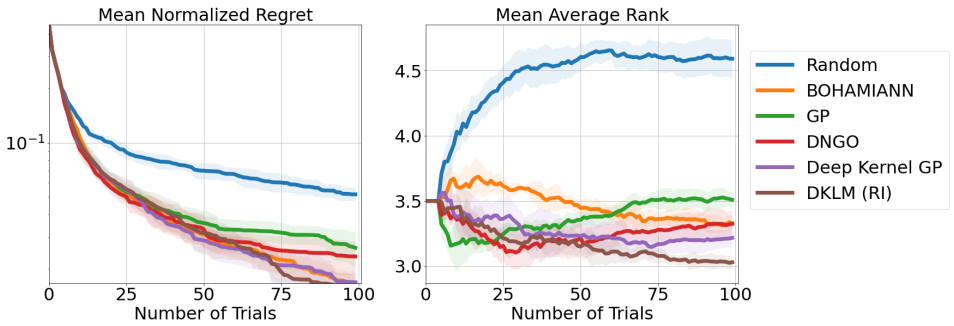

Figure 2: Aggregated comparisons of normalized regret and mean ranks across all search spaces for the non-transfer learning HPO methods on HPO-B-v3

Afterwards, we demonstrate the comparison of state-of-the-art **transfer** against our method in Figure 3. DKLM outperforms the rest of the baselines with lower mean normalized regret and lower mean rank. The superiority of landmark meta-features also becomes evident after a larger number of trials (more than 50) .

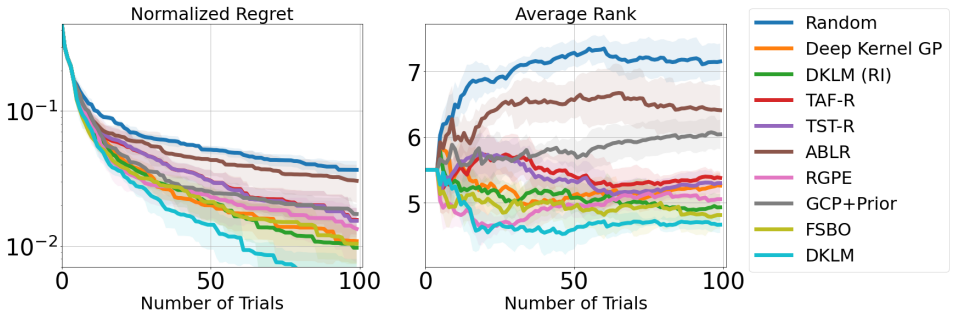

Figure 3: Aggregated comparisons of normalized regret and mean ranks across all search spaces for the transfer learning HPO methods on HPO-B-v3

To further inspect the results we show the performance of DKML and all other baselines in the selected individual search spaces of Figure 4. We notice primarily that meta-learning the initialization in DKLM improves the general performance in most cases. Nevertheless, we notice in 4796 that effect of transfer learning is not evident, as DKLM (RI) and Deep Kernel GP are better than the

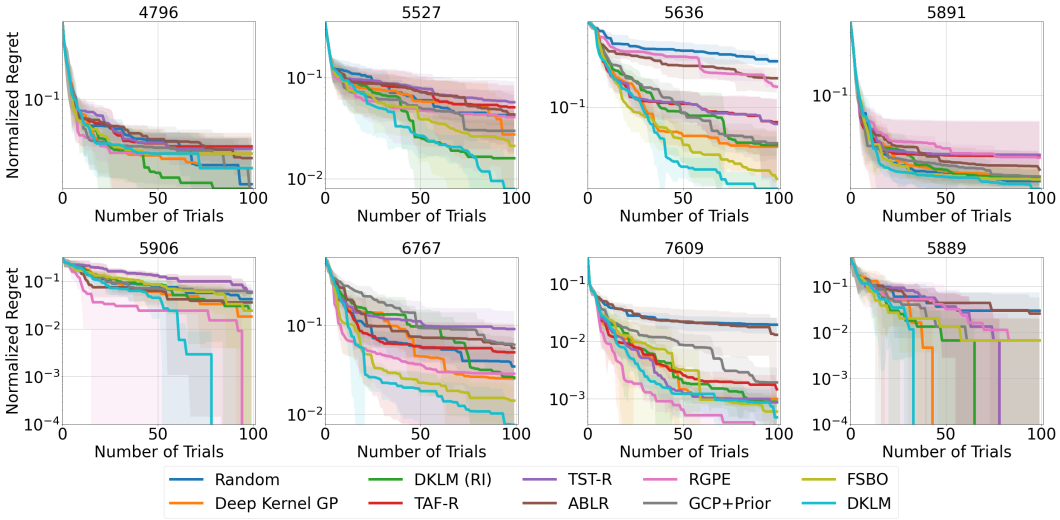

Figure 4: **Normalized regret** comparison of transfer learning HPO methods on selected benchmarks from HPO-B-v3

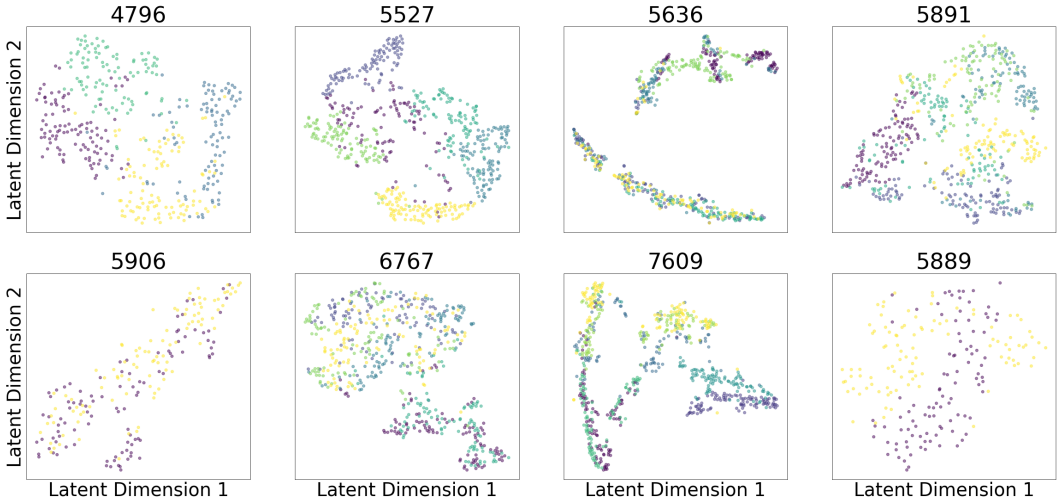

Figure 5: 2D illustration of meta-features extracted from each task in 8 selected search spaces. For each task, we sample 100 sets of 5 data points to extract landmark meta-features. We reduce the dimensionality of the meta-features into a 2D representation via TSNE (Liu et al., 2017).

meta-initialized variants, DKLM, and FSBO respectively. Still, landmark meta-features prove yet again that they are effective in a single task setting.

6.4 End-to-End Landmark Meta-features

To motivate the importance of landmark meta-features, we illustrate in Figure 5 the 2D latent dimensions of the landmark meta-features for every test task in the 16 spaces of HPO-B-v3. Each point on the graph represents a set of meta-features extracted from 100 randomly sampled data points, i.e. $t = 100$, from the individual tasks after meta-initialization of the weights of DKLM. We observe that the same color-coded meta-features, i.e. belonging to the same task, lie generally within the vicinity of each other, and distant from other tasks. As pointed out by Jomaa et al. (2021a), any meta-feature extractor should be able to preserve inter-and intra-dataset similarity, a property that is evident here.

6.5 RESILIENCE TO NEGATIVE TRANSFER

To test the phenomenon of negative transfer we design a specific ablation experiment, where the datasets in the meta-test split are uncorrelated to the datasets in the meta-train split. First of all, we define the correlation of two datasets using Kendall's τ correlation coefficient (Kendall, 1945). We sample 100 random hyper-parameter configurations and query the validation accuracies of these configurations on both datasets from an HPO-B search space, to compute the Kendall's τ coefficient between the accuracies' vectors. Afterward, for every HPO-B search space, we i) compute the correlation between each meta-test dataset and all the meta-training datasets, ii) keep the 10% of the meta-training datasets that have the smallest mean Kendall's τ coefficient to the meta-test datasets.

We report the ablation results in Figure 6, where DKLM (NT) and FSBO (NT) denote methods trained using only the 10% most uncorrelated meta-training datasets. In contrast, DKML and FSBO are trained using all the 100% meta-training dataset. We notice that performance gap of the ranks between DKLM (NT) and FSBO (NT) is larger than that between DKLM and FSBO, validating DKML's resilience to the negative transfer phenomenon.

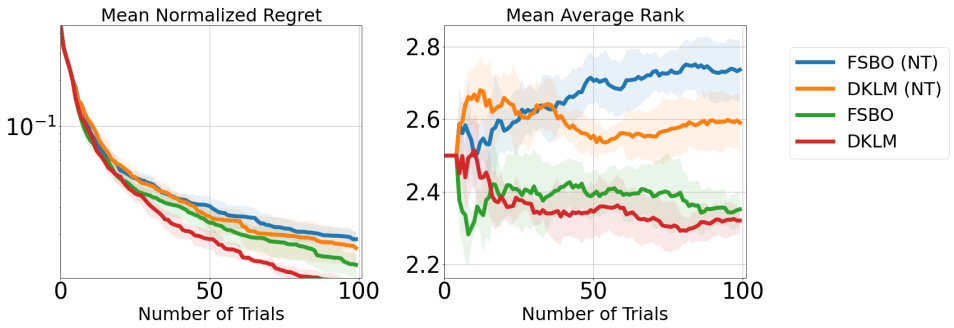

Figure 6: Aggregated comparisons of normalized regret and mean ranks across all HPO-B search spaces, where the gap between DKLM (NT) and FSBO (NT) demonstrates the resilience to negative transfer.

7 LIMITATIONS

Despite the fact that our method significantly reduces the time for fitting Machine Learning, we caution practitioners against overtuning their model for a large number of configuration trials, only to get a very small improvement in accuracy, unless it is absolutely necessary from a business need.

8 CONCLUSION

In this paper, however, we propose DKLM as a simple yet effective method to better condition deep kernel Gaussian Processes on tasks. Inspired by landmark meta-features, we design a set-based meta-feature extractor that captures the interaction between the available hyperparameters and their respective responses, and consequently generates distinct task-specific attributes. DKLM is meta-learned on a set of source tasks in an end-to-end fashion to jointly approximate the response surface over the shared hyperparameter and landmark meta-feature space. We show in a battery of experiments the significance of landmark meta-features, outperforming state-of-the-art HPO baselines in non-transfer and transfer learning settings.

ETHICS STATEMENT

In our work, we use only publicly available data without privacy concerns. Furthermore, our algorithm reduces the overall time for fitting machine learning algorithms, therefore, saving computational resources and yielding a positive impact on energy consumption.

REPRODUCIBILITY STATEMENT

We promote reproducibility as detailed below:

- We use only publicly available datasets.
- All our baselines are publicly available and provided by the HPO-B benchmark (Pineda-Arango et al., 2021).
- We clearly describe our method in Section 4 and provide implementation details in Section 6.2.
- Finally, we plan to make the source code of our method publicly available.

REFERENCES

Rémi Bardenet, Mátyás Brendel, Balázs Kégl, and Michèle Sebag. Collaborative hyperparameter tuning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pp. 199–207, 2013.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305, 2012.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pp. 2546–2554, 2011.

Hao Chen, Lili Zheng, Raed Al Kontar, and Garvesh Raskutti. Stochastic gradient descent in correlated settings: A study on gaussian processes. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Alexander I Cowen-Rivers, Wenlong Lyu, Zhi Wang, Rasul Tutunov, Hao Jianye, Jun Wang, and Haitham Bou Ammar. Hebo: Heteroscedastic evolutionary bayesian optimisation. *arXiv preprint arXiv:2012.03826*, 2020. winning submission to the NeurIPS 2020 Black Box Optimisation Challenge.

Matthias Feurer, Jost Tobias Springenberg, and Frank Hutter. Using meta-learning to initialize bayesian optimization of hyperparameters. In *Proceedings of the International Workshop on Meta-learning and Algorithm Selection co-located with 21st European Conference on Artificial Intelligence, MetaSel@ECAI 2014, Prague, Czech Republic, August 19, 2014*, pp. 3–10, 2014.

Matthias Feurer, Jost Tobias Springenberg, and Frank Hutter. Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pp. 1128–1135, 2015.

Matthias Feurer, Benjamin Letham, and Eytan Bakshy. Scalable meta-learning for bayesian optimization. *CoRR*, abs/1802.02219, 2018.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 1126–1135, 2017.

Jacob R. Gardner, Geoff Pleiss, Kilian Q. Weinberger, David Bindel, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 7587–7597, 2018.

Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization - 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers*, pp. 507–523, 2011.

Hadi S. Jomaa, Lars Schmidt-Thieme, and Josif Grabocka. Dataset2vec: learning dataset meta-features. *Data Min. Knowl. Discov.*, 35(3):964–985, 2021a.

Hadi S. Jomaa, Lars Schmidt-Thieme, and Josif Grabocka. Hyperparameter optimization with differentiable metafeatures. *CoRR*, abs/2102.03776, 2021b.

Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Hugo Larochelle, Dumitru Erhan, Aaron C. Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, pp. 473–480, 2007.

Haebeom Lee, Hayeon Lee, Donghyun Na, Saehoon Kim, Minseop Park, Eunho Yang, and Sung Ju Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 3744–3753, 2019.

Rui Leite, Pavel Brazdil, and Joaquin Vanschoren. Selecting classification algorithms with active testing. In *Machine Learning and Data Mining in Pattern Recognition - 8th International Conference, MLDM 2012, Berlin, Germany, July 13-20, 2012. Proceedings*, pp. 117–131, 2012.

Shusen Liu, Dan Maljovec, Bei Wang, Peer-Timo Bremer, and Valerio Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Trans. Vis. Comput. Graph.*, 23(3):1249–1268, 2017.

Donald Michie, David J. Spiegelhalter, and C. C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.

Jonas Mockus. On bayesian methods for seeking the extremum. In *Optimization Techniques, IFIP Technical Conference, Novosibirsk, USSR, July 1-7, 1974*, pp. 400–404, 1974.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018.

Massimiliano Patacchiola, Jack Turner, Elliot J. Crowley, Michael F. P. O'Boyle, and Amos J. Storkey. Bayesian meta-learning for the few-shot setting via deep kernels. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Valerio Perrone and Huibin Shen. Learning search spaces for bayesian optimization: Another view of hyperparameter transfer learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 12751–12761, 2019.

Valerio Perrone, Rodolphe Jenatton, Matthias W. Seeger, and Cédric Archambeau. Scalable hyperparameter transfer learning. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 6846–6856, 2018.

Bernhard Pfahringer, Hilan Bensusan, and Christophe G. Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pp. 743–750, 2000.

Sebastian Pineda-Arango, Hadi S. Jomaa, Martin Wistuba, and Josif Grabocka. HPO-B: A large-scale reproducible benchmark for black-box HPO based on openml. *CoRR*, abs/2106.06257, 2021.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning, ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, pp. 63–71, 2003.

David Salinas, Huibin Shen, and Valerio Perrone. A quantile-based approach for hyperparameter transfer learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pp. 8438–8448, 2020.

Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. Scalable hyperparameter optimization with products of gaussian process experts. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I*, pp. 33–48, 2016.

Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Md. Mostofa Ali Patwary, Prabhat, and Ryan P. Adams. Scalable bayesian optimization using deep neural networks. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 2171–2180, 2015.

Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust bayesian neural networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 4134–4142, 2016.

Quan Sun and Bernhard Pfahringer. Pairwise meta-rules for better meta-learning-based algorithm ranking. *Mach. Learn.*, 93(1):141–161, 2013.

Kevin Swersky, Jasper Snoek, and Ryan Prescott Adams. Multi-task bayesian optimization. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pp. 2004–2012, 2013.

Joaquin Vanschoren. Meta-learning: A survey. *CoRR*, abs/1810.03548, 2018.

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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime G. Carbonell. Characterizing and avoiding negative transfer. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 11293–11302, 2019.

Andrew Gordon Wilson and Zoubin Ghahramani. Copula processes. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pp. 2460–2468, 2010.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pp. 370–378, 2016a.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pp. 370–378, 2016b.

Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4148–4158, 2017.

Martin Wistuba and Josif Grabocka. Few-shot bayesian optimization with deep kernel surrogates. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Martin Wistuba, Nicolas Schilling, and Lars Schmidt-Thieme. Hyperparameter search space pruning - A new component for sequential model-based hyperparameter optimization. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II*, pp. 104–119, 2015a.

Martin Wistuba, Nicolas Schilling, and Lars Schmidt-Thieme. Learning hyperparameter optimization initializations. In *2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015, Campus des Cordeliers, Paris, France, October 19-21, 2015*, pp. 1–10, 2015b.

Martin Wistuba, Nicolas Schilling, and Lars Schmidt-Thieme. Two-stage transfer surrogate model for automatic hyperparameter optimization. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I*, pp. 199–214, 2016.

Martin Wistuba, Nicolas Schilling, and Lars Schmidt-Thieme. Scalable gaussian process-based transfer surrogates for hyperparameter optimization. *Mach. Learn.*, 107(1):43–78, 2018.

Dani Yogatama and Gideon Mann. Efficient transfer learning method for automatic hyperparameter tuning. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, pp. 1077–1085, 2014.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 3391–3401, 2017.

A ALGORITHMS

We describe in Algorithm 2 the general procedure for BO optimization. Our surrogate model, DKLM, is fit to the history of observations $\mathcal{H}^{t-1}$ at each trial via Algorithm 3.

Algorithm 2: Sequential Model-Based Optimization with DKLM

Input: kernel parameters θ, network parameters w, black-box function f , N initial evaluations $\mathcal{H}^0 = \{(x_i, y_i)\}_{i=0}^N$ on target task, acquisition function $\mathcal{A}(\cdot)$, number of trials T,
Output: Best configuration and response (x_*, y_*)

for $t = 1$ *to* T **do**
  Fit ϕ and kernel using $\mathcal{H}^{t-1}$ in Algorithm 3 ;
  Obtain next configuration to evaluation $x = \arg\max_{x \in \mathcal{X}} \mathcal{A}(x)$;
  Observe response $y = f(x)$;
  Update History $\mathcal{H}^t = \mathcal{H}^{t-1} \bigcup \{(x, y)\}$;
end
Return best configuration $(x_*, y_*) = \arg\max_{(x,y)\sim\mathcal{H}^t} y$

Algorithm 3: Fitting DKLM to a target task

Input: Learning rates η_1 and η_2, N evaluations $\mathcal{H} = \{(x_i, y_i)\}_{i=0}^N$ on target task, kernel parameters θ, network parameters w
Output: Gaussian Process posterior $p\left(f_* | x_*, \mathcal{H}\right)$

while *not converged* **do**
  Compute marginal likelihood $\mathcal{L}$ on $\mathcal{H}$ (Equation 6);
  $\theta \leftarrow \theta + \eta_1 \nabla_\theta \mathcal{L}$;
  $w \leftarrow w + \eta_2 \nabla_w \mathcal{L}$;
end
Compute Gaussian Process posterior $p\left(f_* | x_*, \mathcal{H}\right) = \mathcal{N}(\mu, \sigma^2)$ (Equation 2)

B EXPERIMENTAL PROTOCOL

We followed the protocol presented in HPO-B, (Pineda-Arango et al., 2021), a benchmark collection of tasks collected from OpenML. The tasks are grouped in 16 search spaces, and every search space contains three splits: meta-training, meta-validation and meta-testing. According to the authors, the splitting for the percentage of tasks is 80%, 10% and 10%, respectively. We refer to the repository [1] to know the exact task IDs for every split.

In our experiments, we meta-train a model per search space, whereas we used the meta-validation data for early-stopping. We use $\eta = 0.001, v = 1, n = 1000, T_{\text{min}} = 2, T_{\text{max}} = 100, b = 64$ for meta-training (Algorithm 1). We optimize the black-box function following Sequential Model Based Optimization (SMBO, Algorithm 2), we use Expected Improvement as the acquisition function:

$$\mathcal{A}(x) = \mathbb{E}\left[\max\left\{f_*(x) - y_{\text{max}}, 0\right\}\right] \tag{8}$$

which uses the mean and variance computations presented in section 3.2 for estimating the expectation. We fit DKLM using ADAM (Kingma & Ba, 2015) with learning rates $\eta_1 = \eta_2 = 0.001$, (see Algorithm 3). We use the meta-validation dataset for choosing the number of hidden layers in Equation 5 and the learning rates η.

C IMPACT OF SAMPLE SIZE

We mentioned in Section 6.5 that the performance of DKLM improves with the increasing number of observed data points. To further examine this observation, we present in Table 1 the negative log

[1] https://github.com/releaunifreiburg/HPO-B

likelihood given a history of size t. To get these results, we sample t points for the history $\mathcal{H}^t$ and compute the negative log-likelihood on 100 separate data points upon meta-initialization of DKLM. The process is repeated 100 times for each task, for a total of 10000 data points. We highlight the trend of the negative log-likelihood by fitting a linear model to the 10000 points and reporting the slope. In most search spaces, the performance improves with the increasing number of observed data points, where we notice some over-fitting in others, e.g. space 5860 that can be alleviated by further regularization.

Table 1: Mean Negative Log-Likelihood upon meta-initialization of DKLM with 1 unit of standard deviation, lower is better.

Space	$t = 5$	$t = 10$	$t = 20$	slope
4796	0.38 $\pm$ 2.64	-0.19 $\pm$ 1.80	-0.78 $\pm$ 0.99	-0.075
5527	2.92 $\pm$ 6.12	1.22 $\pm$ 3.86	0.9 $\pm$ 2.44	-0.12
5636	-1.8 $\pm$ 3.96	-3.01 $\pm$ 0.72	-3.11 $\pm$ 0.55	-0.077
5859	22.78 $\pm$ 40.70	10.18 $\pm$ 28.51	-0.37 $\pm$ 10.09	-1.473
5860	12.76 $\pm$ 21.81	14.61 $\pm$ 19.42	16.13 $\pm$ 18.1	0.215
5889	0.22 $\pm$ 1.27	1.84 $\pm$ 1.86	3.75 $\pm$ 2.97	-0.039
5891	0.33 $\pm$ 0.90	0.02 $\pm$ 0.86	-0.28 $\pm$ 0.69	0.414
5906	2.91 $\pm$ 4.04	5.52 $\pm$ 4.17	9.22 $\pm$ 5.19	-0.003
5965	-0.62 $\pm$ 1.12	-0.93 $\pm$ 0.75	-0.73 $\pm$ 0.77	-0.015
5970	-0.99 $\pm$ 0.12	-1.11 $\pm$ 0.10	-1.22 $\pm$ 0.09	0.021
5971	-1.08 $\pm$ 0.73	-1.26 $\pm$ 0.77	-0.81 $\pm$ 1.10	0.14
6766	0.29 $\pm$ 0.31	0.95 $\pm$ 0.43	2.38 $\pm$ 0.59	-0.118
6767	1.59 $\pm$ 5.64	-0.05 $\pm$ 1.43	-0.39 $\pm$ 0.92	-0.032
6794	-0.54 $\pm$ 1.76	-1.18 $\pm$ 0.82	-1.12 $\pm$ 0.67	0.019
7609	-0.59 $\pm$ 2.42	-1.49 $\pm$ 1.74	-1.89 $\pm$ 0.61	-0.08
7607	-0.05 $\pm$ 0.77	-0.06 $\pm$ 0.87	0.21 $\pm$ 0.76	0.229
Average	2.41 $\pm$ 6.40	1.57 $\pm$ 4.69	1.37 $\pm$ 4.86	-0.062

D DETAILED EXPERIMENTAL RESULTS

We present the detailed results for the 16 search spaces in Figure 7. Additionally, we plot the 2D landmark meta-features in Figure 8.

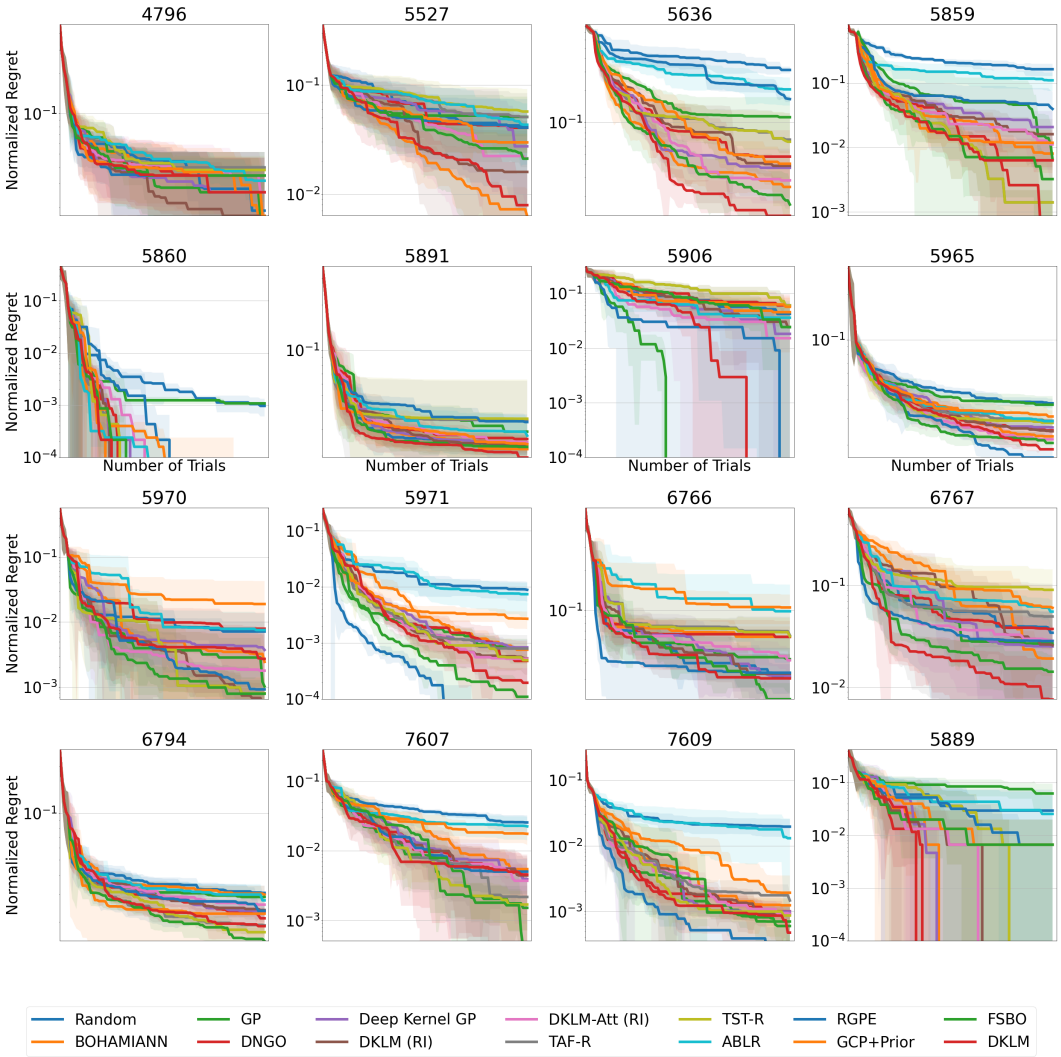

Figure 7: **Normalized regret** comparison of transfer learning HPO methods on all benchmarks from HPO-B-v3

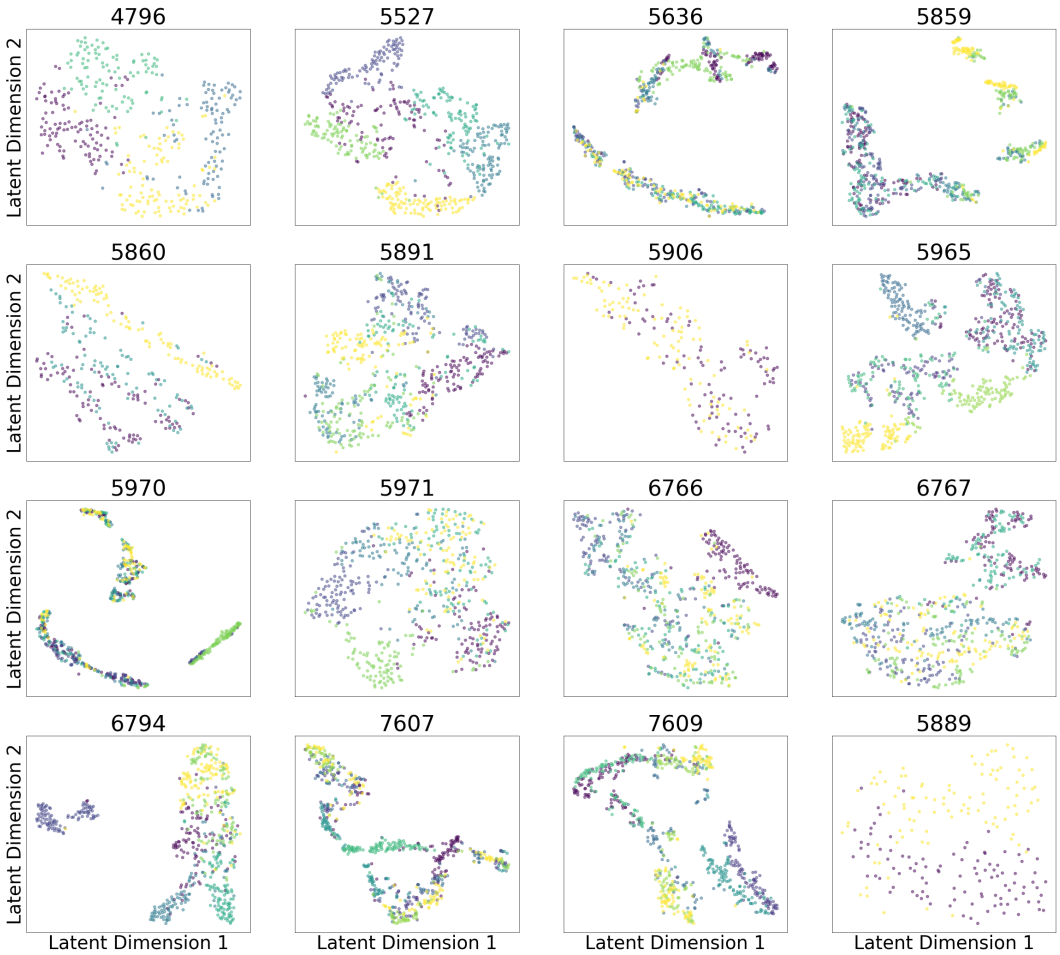

Figure 8: 2D illustration of meta-features extracted from each task in the 16 selected search spaces. For each task, we sample 100 sets of 5 data points to extract landmark meta-features. We reduce the dimensionality of the meta-features into a 2D representation via TSNE (Liu et al., 2017).