LAMDA: Two-Phase HPO via Learning Prior from Low-Fidelity Data

Fan Li[1], **Shengbo Wang**[2], **Ke Li**[3]

[1]State Key Laboratory of Precision Manufacturing for Extreme Service Performance, College of Mechanical and Electrical Engineering, Central South University, Changsha 410083, China
[2]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
[3]Department of Computer Science, University of Exeter, EX4 4RN, Exeter, UK
fanli0525@csu.edu.cn, shnbo.wang@foxmail.com, k.li@exeter.ac.uk

Abstract

Hyperparameter Optimization (HPO) is crucial in machine learning, aiming to optimize hyperparameters to enhance model performance. Although existing methods that leverage prior knowledge—drawn from either previous experiments or expert insights—can accelerate optimization, acquiring a correct prior for a specific HPO task is non-trivial. In this work, we propose to relieve the reliance on external knowledge by learning a reliable prior *directly* from low-fidelity (LF) problems. We introduce `Lamda`, an algorithm-agnostic framework designed to boost any baseline HPO algorithm. Specifically, `Lamda` operates in two phases: (1) it learns a reliable prior by exploring the LF landscape under limited computational budgets, and (2) it leverages this learned prior to guide the HPO process. We showcase how the `Lamda` framework can be integrated with various HPO algorithms to boost their performance, and further conduct theoretical analysis towards the integrated Bayesian optimization and bandit-based Hyperband. We conduct experiments on 56 HPO problems spanning diverse domains and model scales. Results show that `Lamda` consistently enhances its baseline algorithms. Compared to nine state-of-the-art HPO algorithms, our `Lamda` variant achieves the best performance in 51 out of 56 HPO tasks while it is the second best algorithm in the other 5 cases.

Introduction

The performance of modern machine learning (ML) models depends heavily on their hyperparameter configurations (Bischl et al. 2023), such as the learning rate and the number of training epochs for deep neural networks. As a result, automatically tuning hyperparameters has attracted significant interest from both academia and industry (Li et al. 2022a). In practice, this automatic tuning process is often modeled as a black-box optimization problem known as hyperparameter optimization (HPO). Given a hyperparameter space $\mathcal{X}$, the goal of HPO is to identify the best hyperparameter configuration $\mathbf{x}^*$ that minimizes the objective function $f(\mathbf{x})$ for a given ML task:

$$\mathbf{x}^* \in \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}), \tag{1}$$

where $f(\mathbf{x})$ represents the model's performance metric (e.g., training loss or validation error) corresponding to the configuration $\mathbf{x}$. HPO is challenging due to its combinatorially large search space and the computational expense of evaluating $f(\mathbf{x})$, especially for large models.

While methods ranging from simple heuristics like random search (RS) (Bergstra and Bengio 2012) and grid search (Montgomery 2017) to more advanced approaches such as evolution strategies (Real et al. 2019) and sample-efficient Bayesian optimization (BO) (Bergstra et al. 2011; Kandasamy et al. 2018) have been recognized to be effective in HPO, they often start their HPO process from scratch when presented with a new set of ML instances. Such *cold-start* strategy can be inefficient as it ignores knowledge gained from previously solved tasks or domain experts.

In recent years, significant efforts have been made to *warm-start* the HPO process by leveraging *prior knowledge*. These approaches generally fall into two categories.

- **Data-driven**. The *prior* is learned from historical or related tasks. For instance, meta-learning can warm-start the HPO process by setting initial design as hyperparameter configurations that have performed well in similar tasks (Feurer, Springenberg, and Hutter 2015; Lindauer and Hutter 2018). Alternatively, transfer learning trains joint surrogate models across multiple tasks, enabling knowledge transfer as multi-task optimization (Perrone et al. 2018) and few-shot optimization (Wistuba and Grabocka 2021). Historical data can also refine or reduce the search space (Perrone and Shen 2019; Li et al. 2022a). However, data-driven methods rely on the assumption of task similarity and sufficient related data, which can be challenging to satisfy in real-world scenarios.
- **Expert-driven**. The *prior* is explicitly encoded based on expert knowledge about where the optimum might lie. This approach has gained traction as deep learning practitioners develop better insights into promising hyperparameter regions (Bouthillier and Varoquaux 2020), particularly in identifying regions of hyperparameters that are likely to yield good results (Souza et al. 2021; Hvarfner et al. 2022). For example, PriorBand (Mallik et al. 2023) and BO with prior (Souza et al. 2021; Hvarfner et al. 2022) inject expert-defined prior distributions into HPO algorithms. Empirical results in (Shwartz-Ziv et al. 2022) suggest that leveraging priors from high-performing configurations can outperform methods that learn initializations from similar tasks. However, obtaining accurate

prior knowledge for a specific HPO task is non-trivial.

Different from those prior-based methods, another popular way to accelerate HPO is to exploit internally generated low-fidelity (LF) problems, also known as multi-fidelity HPO (Kandasamy et al. 2019; Mikkola et al. 2022). By applying techniques such as training on smaller data subsets or fewer epochs (Klein et al. 2017; Falkner, Klein, and Hutter 2018), one can derive a LF proxy that is computationally cheaper to evaluate while still resembling the high-fidelity objective function. More interestingly, a recent work (Huang and Li 2025) experimentally demonstrated that promising regions containing good low- and high-fidelity solutions have significant overlaps in many HPO scenarios. This finding inspires us to come up with a motivating hypothesis as follows.

Can we learn a reliable prior directly from the low-fidelity proxy of the underlying HPO task without requiring external data or expert knowledge?

Building upon this hypothesis, we propose a two-phase HPO framework, named `Lamda` (**L**e**a**rning prior fro**m** low-fidelity **da**ta), which is algorithm-agnostic and serves as a booster for existing HPO algorithms.

- In the first-phase search, `Lamda` leverages LF evaluations to identify promising regions of LF problems, which are used to construct a reasonably reliable *prior* for underlining HPO algorithms. Thereafter, such learned *prior* is used to guide the search region in the second-phase search within the high-fidelity landscape.
- To demonstrate the algorithm-agnostic nature of `Lamda`, we develop five `Lamda` instances by incorporating it into different existing HPO algorithms, ranging from prior- and bandit-based methods, as well as multi-fidelity BO. In addition, using the prior-based BO (Bect, Bachoc, and Ginsbourger 2019) and bandit-based Hyperband (Mallik et al. 2023) as examples, we analyze the theoretical properties of using `Lamda` therein.
- To validate the effectiveness of our methods, we conduct experiments on 56 HPO tasks across computer vision, natural language processing, as well as tabular data. These tasks involve models with varying scales, ranging from 5-layer fully connected networks and ResNet-50 to large-scale transformers architecture and fine-tuning pretrained models with 305M parameters. Comparing to the corresponding baseline algorithms, using `Lamda` consistently boost their performance. In particular, `Lamda` variants obtain better solutions and reduce query costs by $\approx 75\%$ across all HPO tasks. Further, comparing to 9 state-of-the-art (SOTA) HPO algorithms, our best `Lamda` variant achieves the best performance in 51 out of 56 HPO tasks while it is the second best algorithm in the other 5 cases. Note that our best `Lamda` variant also requires up to 75% less queries than the SOTA HPO algorithms.

Proposed Method

Our proposed `Lamda` comprises a two-phase search strategy (as depicted in Figure 1): ▶ the *first-phase search* initially learn a prior from the LF landscape; ▶ the *second-phase search* leverages the learned information to guide the search. We introduce the search strategies in different phases by addressing two key questions.

How to Learn a Prior from the Low-Fidelity Landscape?

Our basic idea of the *first-phase search* is to divide the LF landscape into two parts: one consists of the promising regions while the other represents the inferior ones. This can be implemented as a binary classification problem. To train such classifier, we leverage the configurations visited so far during the HPO process in the LF landscape, denoted as $\mathcal{S} = \{(\mathbf{x}^i, f_\ell(\mathbf{x}^i))\}_{i=1}^t$ where $f_\ell(\cdot)$ is the LF objective function and t is the current number of function evaluations. In particular, we adopt the classic tree-structured Parzen estimator (TPE) method (Bergstra et al. 2011) as the classifier, given the scalability and supports for both mixed continuous and discrete spaces. It uses the quantile of $\{f_\ell(\mathbf{x})|\mathbf{x} \in \mathcal{S}_\mathbf{x}\}$ to determine the classification boundary, where $\mathcal{S}_\mathbf{x} = \{\mathbf{x}^i \mid (\mathbf{x}^i, f_\ell(\mathbf{x}^i)) \in \mathcal{S}\}$ is the set of sampled configurations in $\mathcal{S}$. Specifically, we divide $\mathcal{S}$ into: $\mathcal{S}_{\text{pro}} = \{\mathbf{x} \mid f_\ell(\mathbf{x}) \leq y^*, \mathbf{x} \in \mathcal{S}_\mathbf{x}\}$ containing promising solutions, and $\mathcal{S}_{\text{inf}} = \{\mathbf{x} \mid f_\ell(\mathbf{x}) > y^*, \mathbf{x} \in \mathcal{S}_\mathbf{x}\}$ containing inferior solutions, where y^* is determined from $\alpha = \Pr(f_\ell(\mathbf{x}) < y^*)$ quantile of $\{f_\ell(\mathbf{x})|\forall \mathbf{x} \in \mathcal{S}_\mathbf{x}\}$. Then we denote

$$\varphi_{\text{pro}}(\mathbf{x}) = p(\mathbf{x} \mid \mathcal{S}_{\text{pro}}), \quad \varphi_{\text{inf}}(\mathbf{x}) = p(\mathbf{x} \mid \mathcal{S}_{\text{inf}}), \quad (2)$$

where $\varphi_{\text{pro}}(\mathbf{x})$ is the probability density function (PDF) of the promising solutions, and $\varphi_{\text{inf}}(\mathbf{x})$ is the PDF of the inferior solutions. We will adopt the kernel density estimation for $\varphi_{\text{pro}}(\mathbf{x})$ and $\varphi_{\text{inf}}(\mathbf{x})$, given its non-parametric nature and applicability to complicated distributions (Chen 2017).

Instead of searching for the optimal configurations in the LF landscape, the purpose of the *first-phase search* is to identify the promising regions, and then use its PDF $\varphi_{\text{pro}}(\mathbf{x})$ as the prior. In practice, the targeted regions are relatively scattered at the beginning and will gradually become focused around the regions that potentially cover the optima (see an illustrative example in Supplementary Material A.1). Based on this observation, we hypothesize that the *first-phase search* can be terminated when the distribution of promising regions becomes stable. To keep track of the progression of such distribution, we propose to use the overlapping coefficient (OVL) (Anderson, Linton, and Whang 2012) as a metric to quantify the similarity between two distributions.

Definition 1 (Overlapping coefficient)**.** *Let $\varphi_1(\mathbf{x})$ and $\varphi_2(\mathbf{x})$ be two PDFs on the search space $\mathcal{X}$. The overlapping coefficient ρ of the two functions is defined as:*

$$\rho(\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x})) = \int_{\mathbf{x} \in \mathcal{X}} \min\{\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x})\} \, d\mathbf{x}. \quad (3)$$

$\rho(\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}))$ ranges from 0 to 1, where $\rho = 1$ if and only if the two distributions are fully overlapped, and $\rho = 0$ if there is no intersection at all. The *first-phase search* is terminated either if the allocated computational budget is exhausted or the OVL of estimated distributions between $\Delta \in \mathbb{N}$ iterations is close enough:

$$1 - \rho\left(\varphi_{\text{pro}}^t(\mathbf{x}), \varphi_{\text{pro}}^{t+\Delta}(\mathbf{x})\right) \leq \gamma, \quad (4)$$

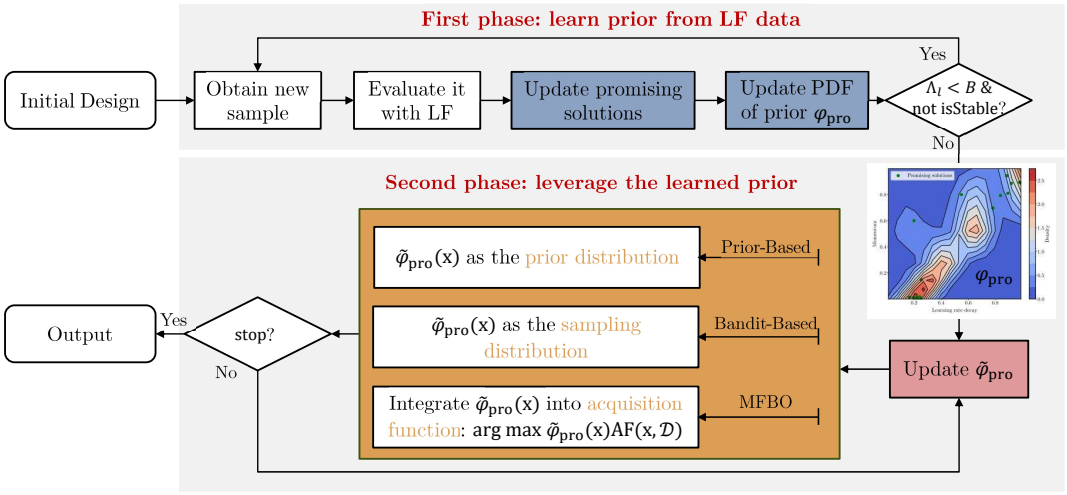

Figure 1: Flowchart of the `Lamda` framework. It consists of two phases: the first phase learns a prior from the LF landscape, and the second phase leverages the learned information to guide the search.

where γ denotes the threshold. The calculation of ρ involves a multidimensional integral, which can be numerically intractable. In practice, we employ the Monte Carlo method to estimate ρ as

$$\begin{aligned}\rho\left(\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x})\right) &= \int_{\mathbf{x}\in\mathcal{X}} \min\left\{\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x})\right\} d\mathbf{x} \\ &= \int_{\mathbf{x}\in\mathcal{X}} \min\left\{1, \frac{\varphi_2(\mathbf{x})}{\varphi_1(\mathbf{x})}\right\} \varphi_1(\mathbf{x}) d\mathbf{x} \\ &= \mathbb{E}\left[\min\left\{1, \frac{\varphi_2(\mathbf{x})}{\varphi_1(\mathbf{x})}\right\}\right] \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \min\left\{1, \frac{\hat{\varphi}_2(\mathbf{x})}{\hat{\varphi}_1(\mathbf{x})}\right\},\end{aligned} \tag{5}$$

where N is the number of samples used in the Monte Carlo sampling, and $\hat{\varphi}(\cdot)$ is an approximation of $\varphi(\cdot)$ such as using the kernel density estimation.

How to Leverage the Learned Prior for HPO?

With the prior learned in the LF landscape, we hypothesize that such information can be used to define the promising regions in the HF landscape. Instead of searching among the entire search space, the *second-phase search* is more focused within the regions defined below:

$$\tilde{\varphi}_{\text{pro}}(\mathbf{x}) = (1-w)\cdot\varphi(\mathbf{x}) + w\cdot\varphi_{\text{pro}}(\mathbf{x}), \tag{6}$$

where $\varphi(\mathbf{x})$ is the probability distribution used to guide the HPO process in the HF landscape, $\varphi_{\text{pro}}(\mathbf{x})$ is the probability distribution of the prior learned from the *first-phase search* in the LF landscape, and $w \in [0, 1)$ with is a hyperparameter that controls the trade-off between the importance of $\varphi(\mathbf{x})$ learned *on-the-fly* and $\varphi_{\text{pro}}(\mathbf{x})$ learned in the *first-phase search*. Supplementary Material A.2 provides a conceptual visualization of leveraging equation (6) during the second-phase search. The redefined promising regions will be closer to the true optimal solution if the prior learned in the first phase are closer to the optimum than those before the redefinition. Note that since $\varphi(\mathbf{x})$ is progressively updated during the HPO process with new configurations evaluated and added to the dataset, it is expected that $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$ will experience a similar trend as $\varphi_{\text{pro}}(\mathbf{x})$ in the *first-phase search*.

Integration and Comparison with Current HPO Methods

Instead of a standalone algorithm, `Lamda` plays as a booster that can be integrated with any existing HPO methods with minor adaptation and thus augmenting the performance of the baseline optimizer. For proof-of-concept, we demonstrate how `Lamda` can be combined with different algorithmic paradigms, including prior-based methods, bandit-based strategies, Bayesian optimization, and `Random Search`. Table 1 summarizes the integration strategies and the associated benefits. In essence, integrating `Lamda` simply involves substituting the sampling strategies of the baseline optimizer with $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$.

- **Prior-Based Methods:** By using `Lamda`, $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$ serves as *a priori* knowledge that represents a reasonable estimation of promising regions in the *second-phase search*. Unlike prior-based methods, which depend on prescribed knowledge or experience from domain experts, $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$ is adaptively learned from data during the *first-phase search* through the HPO process in the LF landscape. As a result, our approach is resilient to 'pathological priors'—whether misleading, lacking in informative values, or potentially adversarial—which are not uncommon when tackling new, unseen real-life black-box applications. Additionally, we expect a scenario between our data-driven priors with those elicited from experts can offer consolidated performance enhancement.
- **Bandit-Based Methods:** By using `Lamda`, $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$ serves as an effective alternative to the sampling distributions used in bandit-based methods. This restricts the HPO process to focus exclusively on the learned promising regions. In contrast, bandit-based methods begin with LF assessments to identify candidates for HF evaluation and then gradually shift the search focus towards these

identified areas. This process, which alternates between exploration and exploitation throughout the entire search space, often leads to inefficient use of computational resources by exploring less promising regions.

- **BO Methods:** Similar to bandit-based methods, BO methods use an acquisition function learned from data to explore the entire search space. This can lead to unnecessary exploration of less promising regions. By using `Lamda`, $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$ restricts the search space within the learned promising regions. Note that this strategy can be applied to the other BO variants.
- **Random Search (RS):** RS samples uniformly without guidance. Integrating `Lamda` provides data-driven sampling that focuses on promising regions.

To facilitate understanding of how `Lamda` integrates with baseline optimizers, we provide concrete examples and pseudocode demonstrations. we select `PriorBand`, `BOHB`, and `MUMBO` as representative algorithms for the prior-, bandit-, and multi-fidelity BO categories, respectively. By augmenting these with `Lamda`, we obtain `Lamda+PriorBand`, `Lamda+BOHB`, and `Lamda+MUMBO`. Additionally, we evaluate `Lamda+BO` and `Lamda+RS` to assess `Lamda`'s effectiveness in vanilla BO and random search scenarios. The integration details and pseudocode are documented in Supplementary Material D.

Furthermore, we provide theoretical analysis under both the prior-based BO framework and bandit-based Hyperband framework, indicating the rational of this augmentation. For `Lamda+BO`, it incorporates prior knowledge in the acquisition function:

$$\mathbf{x}^{n+1} = \arg\max \tilde{\varphi}_{\text{pro}}(\mathbf{x}) \text{AF}(\mathbf{x}, \mathcal{D}), \tag{7}$$

where AF is the acquisition function in vanilla BO such as the expected improvement (EI) considered in this paper. $\mathcal{D} = \{(\mathbf{x}^i, f_h(\mathbf{x}^i))\}_{i=1}^n$ where $f_h(\cdot)$ is the HF objective function. The Gaussian process regression is employed as the surrogate model of $f_h(\cdot)$. For a solution $\tilde{\mathbf{x}}$, the predicted mean and variance of the value distribution of $f_h(\tilde{\mathbf{x}})$ are $\mu_f(\tilde{\mathbf{x}})$ and $\sigma_f^2(\tilde{\mathbf{x}})$.

In `Lamda+BO`, we apply the widely used EI as the acquisition function in equation (7) given as

$$\text{EI}(\tilde{\mathbf{x}}|\mathcal{D}) = \sigma_f(\tilde{\mathbf{x}})\big(z\Phi_f(z) + \phi_f(z)\big), \tag{8}$$

where $z = \frac{f^*_{\mathcal{D}} - \mu_f(\tilde{\mathbf{x}})}{\sigma_f(\tilde{\mathbf{x}})}$, $f^*_{\mathcal{D}} = \min_{(\mathbf{x}, f_h(\mathbf{x})) \in \mathcal{D}} f_h(\mathbf{x})$, Φ_f and ϕ_f denote the cumulative distribution function and probability density function, respectively.

Theorem 1. *Given $\mathcal{D}_n$, $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$, and applying the EI into equation (7), assume the GP models are non-degenerated. Let $\mathcal{D}$ be the collected observations with $(\mathbf{x}^1, f_h(\mathbf{x}^1))$ fixed while $\{(\mathbf{x}^i, f_h(\mathbf{x}^i))\}_{i=2}^n$ are sequentially chosen by*

$$\mathbf{x}^{n+1} = \arg\max \tilde{\varphi}_{\text{pro}}(\mathbf{x}) \text{EI}(\tilde{\mathbf{x}}|\mathcal{D}). \tag{9}$$

Then, as $n \to \infty$, almost surely: the acquisition function converges to zero; and the evaluated best objective $f^*_{\mathcal{D}} \to f^*_h$, where f^*_h represents the global optimum of $f_h(\cdot)$.

The proof is given in Supplementary Material B.2. Unlike the theory in (Hvarfner et al. 2022), the convergence property of equation (9) does not need a decaying factor to force exponentially decreasing of the priors of promising LF regions. Based on the dynamic update of $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$, it is sufficient to capture the promising regions in the HF landscape according to the overlap.

For bandit-based method, we study the `Lamda+PriorBand` in the theoretical routine of Hyperband. The algorithm involves two loops. In the outer loop, at the k-th round, the algorithm allocates $B_{k,s} = 2^k + \text{poly}(k)$ budgets and $n_{k,s} = 2^s$ configurations randomly sampled from $\tilde{\varphi}_{\text{pro}}(x)$, for $s = 0, 1, \ldots, s_{\max}$, subject to $s_{\max} + \log_2(s_{\max}) < k$, where $\text{poly}(k)$ is some polynomial function w.r.t k. In the inner loop, the successive halving algorithm is leveraged to find the best arm among the $n_{k,s}$ arms with $B_{k,s}$ budget. In the context of multi-arm bandits, each configuration $\mathbf{x}^i$ corresponds to an independent arm to pull, whose reward with the j-th pull is denoted by $l_{i,j} = f_j(\mathbf{x}^i)$. We assume there exists $\lim_{k\to\infty} l_{i,k} = \nu_i$ for all $\mathbf{x}^i \in \mathcal{X}$, and denote $\nu_* = \inf_{x\in\mathcal{X}} \nu_i$. Denote also that the distribution of v as F satisfying $P(\nu - \nu_* \le \epsilon) = F(\nu_* + \epsilon)$ for any ϵ. The inverse function is defined by $F^{-1}(y) = \inf\{x : F(x) \le y\}$. In addition, there exists a monotonically decreasing function $\gamma(t) : N \to R$ satisfying $\sup_i |l_{i,t} - \nu_i| \le \gamma(t)$.

Theorem 2. *For fixed $\delta \in (0, 1)$. Let $\hat{\nu}_B$ be the empirically best-performing arm output from successive halving of round $k_B = \log_2(B)$ of the outer loop, and let $s_B < k_B$. Then, there is:*

$$\hat{\nu}_B - \nu_* \le 3\left(F^{-1}\left(\frac{\log(4k_B^3/\delta)}{2^{s_B}} - \nu_*\right) + \gamma\left(\frac{2^{k_B}-1}{k_B}\right)\right), \tag{10}$$

where s_B satisfies $2^{k_B} + \text{poly}(k_B) > 4s_B\mathbf{H}(F, \gamma, 2^{s_B}, 2k_B^3/\gamma)$ with $\mathbf{H}(F, \gamma, n, \delta) = 2n\int_{p_n}^1 \gamma^{-1}(\frac{F^{-1}(t)-\nu_*}{4})dt + \frac{10}{3}\log(2/\delta)\gamma^{-1}\left(\frac{F^{-1}(p_n)-\nu_*}{4}\right)$ and $p_n = \frac{\log_2(2/\delta)}{n}$.

Since all configurations for successive halving tasks are sampled randomly from a probability distribution described by $\tilde{\varphi}_{\text{pro}}$, the theoretical results, specifically Corollary 3 in (Li et al. 2017), still hold in this case. Different from Hyperband that relied on non-adaptive grid search exhausting $c\log_2(B)$ overall budgets with some constant c, we sample configurations and allocate budget through both grid search and adaptive design based on $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$. Theoretically, this requires the same order of budgets as Hyperband. It will be a quite interesting question to ask how the fact of overlapping can help avoiding the grid search of Hyperband, which will be our future work.

Experiment Setup

We seek to answer the following research questions (RQs) through our empirical study in the following paragraphs.

- *RQ1*: Whether integrating `Lamda` into existing HPO algorithms does yield improvements in optimization performance.

Algorithm	Original Sampling Strategy	Integrated Strategy	Benefits
Prior-Based	Expert-defined prior $p(x)$	Replace $p(x)$ with $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$	① Avoid dependent on domain experts; ② Adapt to unseen real-life applications.
Bandit-Based	Uniform sampling the or sampling using densidy of good solution over full space	Replace sampling function with $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$	① Restrict the sampling focus on the learned promising regions; ② Avoid using different fidelity to exploration and exploitation throughout the entire search space.
BO-Based	Posterior-based acquisition over full space	Integrate $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$ into acquisition function: $\arg\max \tilde{\varphi}_{\text{pro}}(\mathbf{x})\text{AF}(\mathbf{x}, \mathcal{D})$	① Restrict the search space within the learned promising regions.

Table 1: Integration of Lamda with different categories of HPO methods

- *RQ2*: How HPO algorithms augmented with `Lamda` perform compared to state-of-the-art methods.
- *RQ3*: Whether the amount of computation in the first phase and the total computational overhead have significant impacts on the performance of `Lamda`.
- *RQ4*: Whether variations in parameters within `Lamda` significantly affect its performance.

Benchmark Suites

Our experiments consider 56 problems that cover search spaces including mixed types and log-scaled hyperparameters. Further, they involve a wide range of downstream tasks including image classification, language modeling, tabular data processing, medical applications, and translation. They are selected from five sources. ① Tabular benchmarks include ▶ four cases from `FCNet` (Pfisterer et al. 2022), each with 6 hyperparameters; ▶ one from `NAS-Bench-301` with 34 hyperparameters (Pfisterer et al. 2022); ▶ three from `NAS-Bench-201`, each with 6 hyperparameters (Eggensperger et al. 2021); and ▶ eighteen benchmarks from `rpart` on decision tree, `glmnet` on elastic net, `ranger` on random forest, and `XGBoost` (Eggensperger et al. 2021). ② Surrogate benchmarks include ▶ four problems from `PD1 benchmarks` with 4 hyperparameters (Mallik et al. 2023; Wang et al. 2021); ▶ three problems from `JAHSBench` (Mallik et al. 2023) with 14 mixed-type hyperparameters for tuning both the neural networks architecture and training hyperparameters. ③ Training two deep neural networks include LeNet on CIFAR-10, and ResNet-18 on CIFAR-10 and CIFAR-100 with 5 hyperparameters. ④ Two synthetic Hartmann functions (Mallik et al. 2023) with three and six variables respectively. ⑤ 20 problems for fine-tuning pretrained image classification models (Pineda-Arango et al. 2024), each with 69 hyperparameters. We consider varying levels of overlap between LF and HF promising regions, with overlap statistics also reported in Table 3 of Supplementary Material. More details on all benchmarks are available in Supplementary Material E.

Peer Algorithms

We choose nine peer algorithms as the baselines to validate the effectiveness of proposed approach. They are ▶ PriorBand (Mallik et al. 2023) and PFNs4BO (Müller et al. 2023) as prior-based methods; ▶ HyperBand (Li et al. 2017), `BOHB` (Falkner, Klein, and Hutter 2018), and Hyper-Tune (Li et al. 2022b) as bandit-based methods; ▶ MUMBO (Li, Kirby, and Zhe 2021) and DPL (Kadra et al. 2023) as MFBO methods; and RS), BO and TuRBO (Eriksson et al. 2019) as other popular HPO methods. During our experiments, we used the default values for these algorithms. For our algorithm, the parameter settings are as follows: $\gamma = 0.1$, $\Delta = 5$, $\alpha = 15$ and $w = 0.5$. Additionally, we allocate a computational budget of $B = 5D$ high-fidelity resources in the first-phase search, where D denotes the problem's dimensionality. Our code for reproducing the experiments is open-sourced at [1]`Lamda`.

Experimental Results

Effectiveness of Using `Lamda` in Existing HPO Algorithms

To answer RQ1, we integrate `Lamda` into five commonly used optimizers: `PriorBand`, `BOHB`, `MUMBO`, `BO` and `RS`. For `BOHB`, `MUMBO`, `BO` and `RS`. 33 tasks from the above tabular, surrogate and synthetic benchmarks are used. For `PriorBand`, the evaluation included eight tasks: four from the original paper (PD1-LM1B, PD1-WMT, MFH3, and MFH6) and four additional tasks in FCNet. In addition, good prior are used at `PriorBand` for PD1-LM1B, PD1-WMT, MFH3, and MFH6. The results are presented and analyzed from the following three dimensions.

Overall performance: The overall performance metrics highlight the effectiveness of using `Lamda`. Table 2 summarizes the numbers of win/lose/tie obtained by using the Wilcoxon signed-rank test, while Figure 2 shows the average rank over the HPO tasks. The Wilcoxon signed-rank test results in Table 2 confirm that using `Lamda` achieves significantly better performance than the baseline in more than half of the tasks, with the remaining tasks yielding results comparable to the baseline. Specifically, for `BOHB`, `MUMBO`, `BO`, and `RS`, the integration of `Lamda` achieves 24 wins, 19 wins, 17 wins, and 29 wins, respectively. In the case of `PriorBand`, `Lamda+PriorBand` performs comparably to `PriorBand` on tasks with prior information but significantly outperforms it on tasks without prior information, demonstrating its adaptability and effectiveness in scenarios where prior knowledge is unavailable. Furthermore, the

[1]https://github.com/fanli525/Lamda

rank plot in Figure 2 also demonstrates that Lamda consistently improves the performance of all five algorithms. Overall, Lamda initially shows slower progress due to the resources required for learning the prior. However, after approximately 15 HF resources, Lamda-integrated algorithms quickly achieves top rankings, highlighting the long-term benefits of leveraging prior knowledge. In addition, Lamda-integrated algorithms achieve better solutions while reducing query costs by nearly 75%, significantly improving search efficiency.

Convergence curves: The convergence curves for each problem provide deeper insights into the benefits of the proposed framework. Figure 5 to Figure 13 in Supplementary Material F.1 shows the performance trajectories of Priorband, BOHB, MUMBO, BO, and RS under the proposed framework. The results from the convergence curves reveal that the use of Lamda leads to faster convergence toward high-quality solutions during the second phase. For Priorband, Figure 5 of Supplementary Material shows that Lamda+Priorband consistently converges faster than Priorband across all eight tasks, demonstrating the efficiency of leveraging prior knowledge. For the other four algorithms, using the prior accelerates the discovery of effective solutions compared to the baseline on most of the tasks, as shown in Figure 6 to Figure 13 in Supplementary Material F.1.

Visualize the learned priors: To evaluate the effectiveness of Lamda, we analyze the alignment between the learned priors and the distribution of high-quality solutions at the HF level, and examine how this alignment influences HF sampling behavior in Phase Two. To approximate the distribution of good solutions, we generate 10,000 hyperparameter configurations for each problem. The top 10% configurations are retained to approximate the distribution of good solutions at high and low fidelity levels. We illustrate this analysis on two representative benchmarks: FCNet-Naval-Propulsion, which exhibits a high degree of fidelity alignment (OVL = 0.901), and JAHS-CIFAR-10, which shows a lower overlap (OVL = 0.574).

- For FCNet-Naval-Propulsion, the results are presented in Figure 3, where each subplot corresponds to a hyperparameter dimension. This problem exhibits a high overlap between the PDFs of the top low- and high-fidelity solutions (as indicated by the red and green curves). The learned priors, shown in blue, exhibit varying degrees of alignment with the PDFs of the top HF solutions across different dimensions. In D1, the alignment is particularly poor (OVL = 0.227). This is likely due to the underlying HF solution density being sharply peaked and heavy-tailed, which poses challenges for modeling. Consequently, the learned prior in this dimension is nearly flat and fails to provide meaningful guidance. In contrast, D2–D4 show moderate to strong alignment with the HF distribution (OVL $= 0.612, 0.804$, and 0.792, respectively). These dimensions appear to have more regular HF landscapes, which are more amenable to modeling. This suggests that Lamda is capable of learning informative priors when the underlying structure is smooth. To assess how this alignment impacts optimization, we examine the HF sampling behavior in phase two. In D2–D4, where the learned priors are well-aligned with the HF distribution, HF samples rapidly concentrate around high-quality regions from the early stages of the search. This demonstrates that Lamda can leverage informative priors to guide the optimization process and accelerate convergence. In D1, however, the flat prior results in more dispersed early sampling. Nevertheless, the Bayesian optimization component of Lamda progressively adjusts the search, gradually shifting samples toward better regions as more HF evaluations are collected. This adaptive behavior highlights the robustness of Lamda: even when prior information is uninformative or misaligned, it does not mislead the optimization and instead allows for stable convergence through iterative refinement.
- For JAHS-CIFAR-10, the results exhibit similar trends as presented in Figure 4 of Supplementary Material F.1. The overall overlap between low- and high-fidelity distributions is relatively low (OVL = 0.574), primarily due to poor alignment in certain dimensions—most notably D1. The learned prior tends to resemble a near-uniform distribution and thus provides little guidance. However, it does not mislead the optimization process. In contrast, dimensions such as D2, D3, and D8 show stronger prior-HF alignment (OVL > 0.7), allowing HF samples to concentrate more efficiently in promising regions.

In summary, across both benchmarks, we observe that Lamda benefits from strong prior-HF alignment by enabling early concentration of HF samples in high-performing regions, thus accelerating convergence. More importantly, in dimensions where the alignment is poor, the learned priors remain non-misleading and are adaptively refined through the Bayesian optimization process. This demonstrates that Lamda achieves a favorable balance between leveraging informative priors when possible and maintaining robustness when prior information is limited.

Comparison with State-of-the-Art HPO Methods

To answer RQ2, we evaluate the performance of Lamda framework compared to peer HPO methods across 56 tasks, including 33 tabular and surrogate benchmarks, 3 tasks on real deep neural networks, and 20 hyperparameter optimization tasks for fine-tuning pretrained image classification models. For the 33 tabular and surrogate benchmarks, Figure 4 presents the average ranks of the algorithms. The results indicate that Lamda+BO and Lamda+MUMBO consistently achieve top positions. The convergence curves in Figure 17 and Figure 18 of Supplementary Material F.3.1 show that the two algorithms deliver superior optimization results with nearly 75% fewer queries. In contrast, Hyperband and RS exhibit the lowest ranks, likely due to their reliance on random sampling strategies. The additional 3 tasks on deep neural networks and 20 tasks on pretrained image classification models further validate the advantages of our framework. As detailed in Supplementary Materia F.3.2 and F.3.3, Lamda consistently improves baseline algorithm performance.

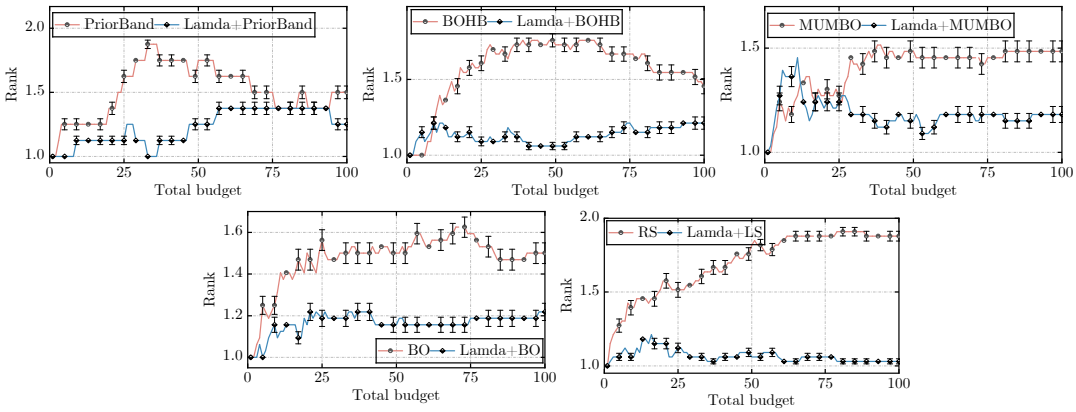

Figure 2: Comparing average relative ranks of PriorBand, BOHB, MUMBO, BO and RS under the proposed framework across 33 HPO tasks. The results show that Lamda can boost the performance of all five algorithms

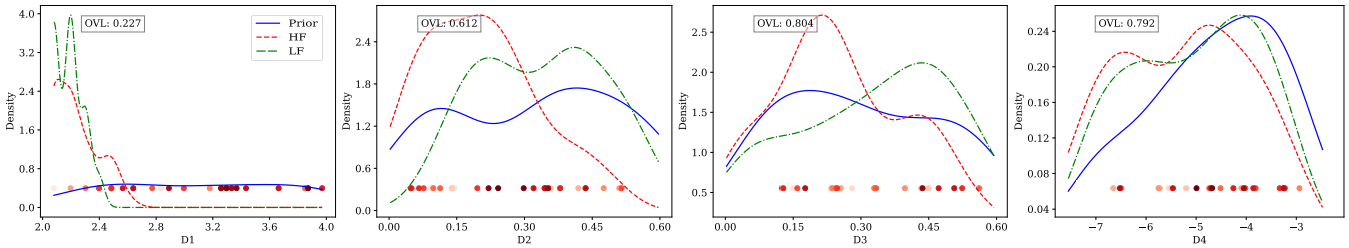

Figure 3: Prior–solution alignment and its impact on HF sampling in FCNet-Naval-Propulsion. Each subplot shows one hyperparameter dimension (D1, D2, D3, D4) with learned prior (blue), PDF of top low- and high-fidelity solutions (green, red), and HF sampling (red points). The overall overlap between the learned prior and the PDF of HF good solutions is moderate (OVL = 0.574). Among the examined dimensions, D1 shows poor alignment with near-uniform prior, giving limited but non-misleading guidance. D2, D3, and D4 have stronger alignment (OVL > 0.7), enabling more focused HF sampling. Red points darken over time, indicating early broad search followed by refinement.

Lamda+BOHB	Lamda+MUMBO	Lamda+BO	Lamda+RS	Lamda+PriorBand
24/0/9	**19**/0/14	**17**/1/15	**29**/1/3	**3**/0/5

Table 2: Performance comparison (Win/Lose/Tie) of five algorithms against their baselines over 100 HF evaluations.

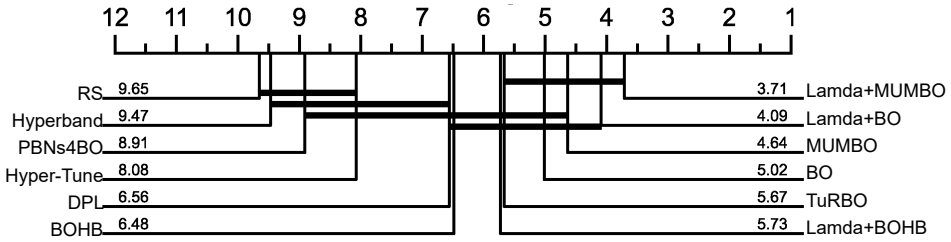

Figure 4: Comparing average relative ranks of peer algorithms across 33 HPO tasks. Lamda+BO and Lamda+MUMBO consistently maintain top positions throughout the evaluations.

Impact of Computational Budget on Lamda Performance

To answer RQ3, we investigate the impact of the computational budget (B) on Lamda's performance by evaluating multiple budget settings. We conduct controlled experiments under the BOHB framework, varying $B \in D, 4D, 8D, 10D, 20D$ across benchmark tasks that exhibit varying degrees of overlap between high- and low-fidelity solution distributions. Specifically, we compare four FCNet tasks—Naval Propulsion, Protein Structure, Slice Localization, and Parkinson's Telemonitoring—with relatively high alignment (overlap scores OVL $\in [0.69, 0.90]$), and three JAHS tasks—CIFAR-10, Colorectal Histology, and Fashion-MNIST—with significantly lower alignment (OVL ≈ 0.5). We assess two key aspects: (1) the convergence behavior of algorithm and (2) the quality of the learned priors.

The budget parameter B influences the timing of the transition to the second phase. As shown in Figure 14 of Supplementary Material F.2, Lamda consistently achieves faster

convergence than the original BOHB across different B values. As more resources are allocated to the second phase, the performance advantage of Lamda over BOHB gradually increases. In addition, selecting a smaller B enables earlier initiation of the second phase. Therefore, under a constrained overall computational budget, a smaller B can lead to better performance within a shorter time frame. We further analyze the prior distributions learned during the first phase. Using FCNet-Naval-Propulsion as an example (as shown in Figure 15 of Supplementary Material F.2), we observe that the learned PDFs are closely aligned with the true low-fidelity distributions across different B settings. Notably, larger B values yield priors that more accurately capture the underlying distribution. Similar results are also observed on JAHS-CIFAR-10, as shown in Figure 16 of Supplementary Material F.2.

Sensitivity of Lamda to Parameter Settings

To answer RQ4, a comprehensive sensitivity analysis of Lamda with respect to several hyperparameters, including γ, Δ, α, and w, is presented in Supplementary Material F.4. The results indicate that the algorithm's performance is not sensitive to those parameters.

Conclusion and Future Works

This paper proposed Lamda, an algorithm-agnostic framework designed as a booster to enhance any baseline HPO algorithms. The core idea is to divide an HPO task into two phases. In the first phase, Lamda learns a reliable prior by exploring the LF landscape under limited computational budgets. In the second phase, the learned prior is used to guide the HPO process. We provided a theoretical analysis of the regret bounds when Lamda is applied to BO and HyperBand. Our experimental results demonstrated that five Lamda instances significantly enhances the performance of the corresponding baseline HPO algorithm.

In the future, we plan to extend Lamda by introducing multiple interconnected LF landscapes. This may increase the robustness of the learned priors. Lamda also includes parameters such as the quantile threshold for LF problems and the overlapping coefficient between low- and high-fidelity landscapes. We plan to analyze the theoretical underpinnings of how these parameters affect the convergence rate or regret bound of Lamda-augmented HPO algorithms. Last but not least, we intend to apply Lamda beyond HPO to other expensive black-box optimization problems, including computational fluid dynamics optimization (Liu, Koziel, and Ali 2017), material discovery (Khatamsaz et al. 2021), and drug design (Greenman, Green, and Gómez-Bombarelli 2021).

References

Anderson, G.; Linton, O. B.; and Whang, Y.-J. 2012. Nonparametric estimation and inference about the overlap of two distributions. *J. Financ. Econom.*, 171: 1–23.

Bect, J.; Bachoc, F.; and Ginsbourger, D. 2019. A Supermartingale Approach to Gaussian Process Based Sequential Design of Experiments. *Bernoulli*, 25(4A): 2883–2919.

Bergstra, J.; Bardenet, R.; Bengio, Y.; and Kégl, B. 2011. Algorithms for Hyper-Parameter Optimization. In *NeurIPS'11: Advances in Neural Information Processing Systems 24*, 2546–2554.

Bergstra, J.; and Bengio, Y. 2012. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.*, 13: 281–305.

Bischl, B.; Binder, M.; Lang, M.; Pielok, T.; Richter, J.; Coors, S.; Thomas, J.; Ullmann, T.; Becker, M.; Boulesteix, A.; Deng, D.; and Lindauer, M. 2023. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data. Mining. Knowl. Discov.*, 13(2).

Bouthillier, X.; and Varoquaux, G. 2020. Survey of machine-learning experimental methods at NeurIPS2019 and ICLR2020.

Chen, Y.-C. 2017. A tutorial on kernel density estimation and recent advances. *Biostat. Epidemiol.*, 1: 161 – 187.

Eggensperger, K.; Müller, P.; Mallik, N.; Feurer, M.; Sass, R.; Klein, A.; Awad, N. H.; Lindauer, M.; and Hutter, F. 2021. HPOBench: A Collection of Reproducible Multi-Fidelity Benchmark Problems for HPO. In *NeurIPS'21: Proc.of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Eriksson, D.; Pearce, M.; Gardner, J. R.; Turner, R.; and Poloczek, M. 2019. Scalable Global Optimization via Local Bayesian Optimization. In *NeurIPS'19: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 5497–5508.

Falkner, S.; Klein, A.; and Hutter, F. 2018. BOHB: Robust and Efficient Hyperparameter Optimization at Scale. In *ICML'18: Proc. of the 35th International Conference on Machine Learning, , Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 1436–1445. PMLR.

Feurer, M.; Springenberg, J. T.; and Hutter, F. 2015. 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*, 1128–1135. AAAI Press.

Greenman, K.; Green, W. H.; and Gómez-Bombarelli, R. 2021. Multi-fidelity prediction of molecular optical peaks with deep learning. *Chem. Sci.*, 13: 1152 – 1162.

Huang, M.; and Li, K. 2025. On the Hyperparameter Loss Landscapes of Machine Learning Models: An Exploratory Study. In *Accepted by KDD'25: Proc. of the 31th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM.

Hvarfner, C.; Stoll, D.; Souza, A. L. F.; Lindauer, M.; Hutter, F.; and Nardi, L. 2022. πBO: Augmenting Acquisition Functions with User Beliefs for Bayesian Optimization. In *ICLR'22: Proc. of the 10th International Conference on Learning Representations*. OpenReview.net.

Kadra, A.; Janowski, M.; Wistuba, M.; and Grabocka, J. 2023. Deep Power Laws for Hyperparameter Optimization. *CoRR*, abs/2302.00441.

Kandasamy, K.; Dasarathy, G.; Oliva, J. B.; Schneider, J. G.; and Póczos, B. 2019. Multi-Fidelity Gaussian Process Bandit Optimisation. *J. Artif. Intell. Res.*, 66: 151–196.

Kandasamy, K.; Neiswanger, W.; Schneider, J.; Póczos, B.; and Xing, E. P. 2018. Neural Architecture Search with Bayesian Optimisation and Optimal Transport. In *NeurIPS'18: Advances in Neural Information Processing Systems 31*, 2020–2029.

Khatamsaz, D.; Molkeri, A.; Couperthwaite, R.; James, J.; Arróyave, R.; Allaire, D. L.; and Srivastava, A. 2021. Efficiently exploiting process-structure-property relationships in material design by multi-information source fusion. *Acta Materialia*.

Klein, A.; Falkner, S.; Bartels, S.; Hennig, P.; and Hutter, F. 2017. Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets. In *AISTATS'17: Proc. of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, 528–536. PMLR.

Li, L.; Jamieson, K. G.; DeSalvo, G.; Rostamizadeh, A.; and Talwalkar, A. 2017. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *J. Mach. Learn. Res.*, 18: 185:1–185:52.

Li, S.; Kirby, R. M.; and Zhe, S. 2021. Batch Multi-Fidelity Bayesian Optimization with Deep Auto-Regressive Networks. In *NeurIPS'21: Proc. of the Annual Conference on Neural Information Processing Systems 2021*, 25463–25475.

Li, Y.; Shen, Y.; Jiang, H.; Bai, T.; Zhang, W.; Zhang, C.; and Cui, B. 2022a. Transfer Learning based Search Space Design for Hyperparameter Tuning. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022*, 967–977. ACM.

Li, Y.; Shen, Y.; Jiang, H.; Zhang, W.; Li, J.; Liu, J.; Zhang, C.; and Cui, B. 2022b. Hyper-Tune: Towards Efficient Hyperparameter Tuning at Scale. *Proc. VLDB Endow.*, 15(6): 1256–1265.

Lindauer, M.; and Hutter, F. 2018. Warmstarting of Model-Based Algorithm Configuration. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 1355–1362. AAAI Press.

Liu, B.; Koziel, S.; and Ali, N. T. 2017. SADEA-II: A generalized method for efficient global optimization of antenna design. *J. Comput. Des. Eng.*, 4(2): 86–97.

Mallik, N.; Bergman, E.; Hvarfner, C.; Stoll, D.; Janowski, M.; Lindauer, M.; Nardi, L.; and Hutter, F. 2023. PriorBand: Practical Hyperparameter Optimization in the Age of Deep Learning. *CoRR*, abs/2306.12370.

Mikkola, P.; Martinelli, J.; Filstroff, L.; and Kaski, S. 2022. Multi-Fidelity Bayesian Optimization with Unreliable Information Sources. *CoRR*, abs/2210.13937.

Montgomery, D. C. 2017. *Design and analysis of experiments*. John wiley & sons.

Müller, S.; Feurer, M.; Hollmann, N.; and Hutter, F. 2023. PFNs4BO: In-Context Learning for Bayesian Optimization. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, 25444–25470. PMLR.

Perrone, V.; Jenatton, R.; Seeger, M. W.; and Archambeau, C. 2018. 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*, 6846–6856.

Perrone, V.; and Shen, H. 2019. Learning search spaces for Bayesian optimization: Another view of hyperparameter transfer learning. In *NeurIPS'19: Advances in Neural Information Processing Systems 32*, 12751–12761.

Pfisterer, F.; Schneider, L.; Moosbauer, J.; Binder, M.; and Bischl, B. 2022. YAHPO Gym - An Efficient Multi-Objective Multi-Fidelity Benchmark for Hyperparameter Optimization. In *AutoML'22: Proc. of 2022 International Conference on Automated Machine Learning*, volume 188 of *Proceedings of Machine Learning Research*, 3/1–39. PMLR.

Pineda-Arango, S.; Ferreira, F.; Kadra, A.; Hutter, F.; and Grabocka, J. 2024. Quick-Tune: Quickly Learning Which Pretrained Model to Finetune and How. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Real, E.; Aggarwal, A.; Huang, Y.; and Le, Q. V. 2019. Regularized Evolution for Image Classifier Architecture Search. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 4780–4789. AAAI Press.

Shwartz-Ziv, R.; Goldblum, M.; Souri, H.; Kapoor, S.; Zhu, C.; LeCun, Y.; and Wilson, A. G. 2022. Pre-Train Your Loss: Easy Bayesian Transfer Learning with Informative Priors. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Souza, A. L. F.; Nardi, L.; Oliveira, L. B.; Olukotun, K.; Lindauer, M.; and Hutter, F. 2021. Bayesian Optimization with a Prior for the Optimum. In *PKDD'21: Machine Learning and Knowledge Discovery in Databases. Research Track-European Conference*, volume 12977 of *Lecture Notes in Computer Science*, 265–296. Springer.

Wang, Z.; Dahl, G. E.; Swersky, K.; Lee, C.; Mariet, Z. E.; Nado, Z.; Gilmer, J.; Snoek, J.; and Ghahramani, Z. 2021. Pre-trained Gaussian processes for Bayesian optimization. *arXiv preprint arXiv:2109.08215*.

Wistuba, M.; and Grabocka, J. 2021. Few-Shot Bayesian Optimization with Deep Kernel Surrogates. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.