Supplementary Material for 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, shinbone.wang@foxmail.com, k.li@exeter.ac.uk

A. Illustrative Example of the Methodology

A.1 Example of Promising Region Convergence

To illustrate the dynamics of promising region identification during the first-phase search, we provide a visualization in Figure 1. This figure demonstrates the progression of probability density functions for promising regions during hyperparameter optimization on a transformer model for the LM1B dataset, focusing on the LF problem. It can be seen that the targeted regions are relatively scattered at the beginning and will gradually become focused around the regions that potentially cover the optima.

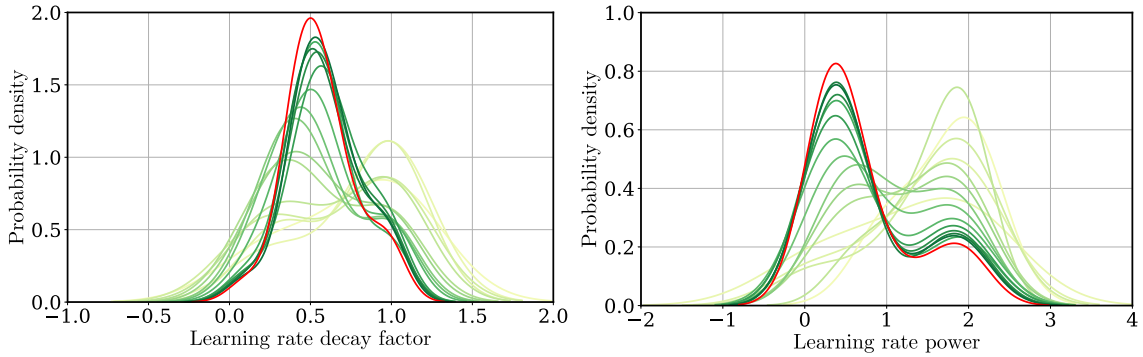

Figure 1: This figure shows the progression of PDFs for promising regions during hyperparameter optimization on a transformer model for the LM1B dataset, focusing on the LF problem. We display PDFs for two out of four hyperparameters, with colors changing from yellow to green to indicate iteration progress. The red line represents the true PDF of the promising solutions in the LF problem.

A.2 Illustrative Example of Leveraging the Learned Prior

Figure 2 illustrates how the incorporation of the prior distribution $\varphi(\mathbf{x})$, learned from the first-phase search, reshapes the original sampling distribution via equation (1). Specifically, the modified density $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$ integrates both the initially identified promising region $\varphi_{\text{pro}}(\mathbf{x})$ and the learned prior, using a weighting parameter w. As shown in the figure, when the learned prior $\varphi(\mathbf{x})$ is more concentrated around the true optimum than the original promising region, the reweighted density $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$ will better align with the true optimum $\varphi_*(\mathbf{x})$. This effect is theoretically supported by Proposition 1, which proves that overlap with the true optimum increases under such conditions.

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

B. Theoretical Analysis

B.1 Proposition

Proposition 1. *Assume the OVL between $\varphi(\mathbf{x})$ and the PDF of the true promising solutions $\varphi_*(\mathbf{x})$ is less than the overlapping between the leraned prior $\varphi_{\text{pro}}(\mathbf{x})$ and $\varphi_*(\mathbf{x})$. Then, the modified sampling function $\tilde{\varphi}_{\text{pro}}(\mathbf{x}) = w_1\cdot\varphi(\mathbf{x}) + w_2\cdot\varphi_{\text{pro}}(\mathbf{x})$, where $w_1 + w_2 = 1$, will have a greater or equal overlapping with $\varphi_*(\mathbf{x})$ compared to the overlapping of $\varphi(\mathbf{x})$ with $\varphi_*(\mathbf{x})$.*

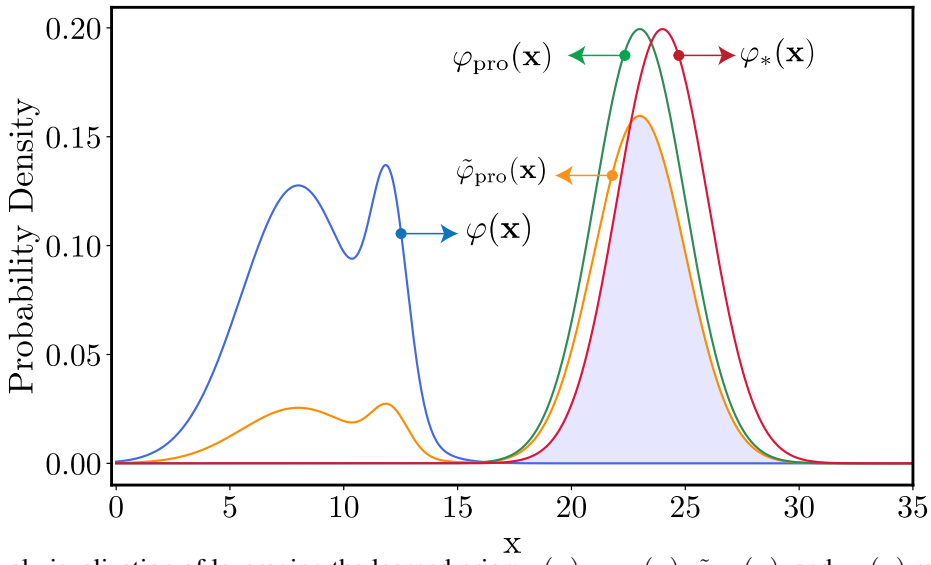

Figure 2: Conceptual visualization of leveraging the learned prior: $\varphi(\mathbf{x})$, $\varphi_{\text{pro}}(\mathbf{x})$, $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$, and $\varphi_*(\mathbf{x})$ represent the original sampling distribution, the leraned prior, the modified density function incorporating the prior with a weight of $w = 0.5$, and the density function of the real optimum.

Proposition 1 suggests that incorporating the learned prior into the sampling distribution enhances its alignment with the distribution of the real optimal solutions.

Proof. Using the definition of $\rho(\varphi_1, \varphi_2)$ in $\rho(\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x})) = \int_{\mathbf{x}\in\mathcal{X}} \min\{\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x})\}\, d\mathbf{x}$, the OVL between $\varphi(\mathbf{x})$, $\varphi_{\text{pro}}(\mathbf{x})$, $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$ and $\varphi_*(\mathbf{x})$ are computed as follows:

$$\begin{aligned}
\rho(\varphi(\mathbf{x}), \varphi_*(\mathbf{x})) &= 1 - \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} |\varphi(\mathbf{x}) - \varphi_*(\mathbf{x})|\, d\mathbf{x}, \\
\rho(\varphi_{\text{pro}}(\mathbf{x}), \varphi_*(\mathbf{x})) &= 1 - \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} |\varphi_{\text{pro}}(\mathbf{x}) - \varphi_*(\mathbf{x})|\, d\mathbf{x}, \\
\rho(\tilde{\varphi}_{\text{pro}}(\mathbf{x}), \varphi_*(\mathbf{x})) &= 1 - \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} |\tilde{\varphi}_{\text{pro}}(\mathbf{x}) - \varphi_*(\mathbf{x})|\, d\mathbf{x}.
\end{aligned} \tag{2}$$

where

$$\rho(\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x})) = \int_{\mathbf{x}\in\mathcal{X}} \min\{\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x})\}\, d\mathbf{x} = 1 - \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} |\varphi_1(\mathbf{x}) - \varphi_2(\mathbf{x})|\, d\mathbf{x}. \tag{3}$$

Given that thus $\varphi(\mathbf{x})$ has a smaller overlap with $\varphi_*(\mathbf{x})$ than $\varphi_{\text{pro}}(\mathbf{x})$, it follows that:

$$\begin{aligned}
&\rho(\varphi(\mathbf{x}), \varphi_*(\mathbf{x})) - \rho(\varphi_{\text{pro}}(\mathbf{x}), \varphi_*(\mathbf{x})) \\
&= 1 - \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} |\varphi(\mathbf{x}) - \varphi_*(\mathbf{x})|\, d\mathbf{x} - \left(1 - \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} |\varphi_{\text{pro}}(\mathbf{x}) - \varphi_*(\mathbf{x})|\, d\mathbf{x}\right) \\
&= \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} |\varphi_{\text{pro}}(\mathbf{x}) - \varphi_*(\mathbf{x})|\, d\mathbf{x} - \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} |\varphi(\mathbf{x}) - \varphi_*(\mathbf{x})|\, d\mathbf{x} \\
&= \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} |\varphi_{\text{pro}}(\mathbf{x}) - \varphi_*(\mathbf{x})| - |\varphi(\mathbf{x}) - \varphi_*(\mathbf{x})|\, d\mathbf{x} \leq 0
\end{aligned} \tag{4}$$

Given the above, if $\rho(\varphi(\mathbf{x}), \varphi_*(\mathbf{x})) < \rho(\tilde{\varphi}_{\text{pro}}(\mathbf{x}), \varphi_*(\mathbf{x}))$, it implies that $\varphi(\mathbf{x})$ has a smaller overlap with $\varphi_*(\mathbf{x})$ than $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$.

This can be further analyzed as:

$$
\begin{aligned}
&\rho\left(\varphi(\mathbf{x}), \varphi_*(\mathbf{x})\right) - \rho\left(\tilde{\varphi}_{\mathrm{pro}}(\mathbf{x}), \varphi_*(\mathbf{x})\right) \\
&= 1 - \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} \left|\varphi(\mathbf{x}) - \varphi_*(\mathbf{x})\right| d\mathbf{x} - \left(1 - \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} \left|w_1\cdot\varphi(\mathbf{x}) + w_2\cdot\varphi_{\mathrm{pro}}(\mathbf{x}) - \varphi_*(\mathbf{x})\right| d\mathbf{x}\right) \\
&= \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} \left|w_1\cdot\varphi(\mathbf{x}) + w_2\cdot\varphi_{\mathrm{pro}}(\mathbf{x}) - \varphi_*(\mathbf{x})\right| d\mathbf{x} - \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} \left|\varphi(\mathbf{x}) - \varphi_*(\mathbf{x})\right| d\mathbf{x} \\
&\leq \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} w_1\cdot\left|\varphi(\mathbf{x}) - \varphi_*(\mathbf{x})\right| + w_2\cdot\left|\varphi_{\mathrm{pro}}(\mathbf{x}) - \varphi_*(\mathbf{x})\right| - \left|\varphi(\mathbf{x}) - \varphi_*(\mathbf{x})\right| d\mathbf{x} \\
&= \frac{1}{2}\int_{\mathbf{x}\in\mathcal{X}} w_2\cdot\left|\varphi_{\mathrm{pro}}(\mathbf{x}) - \varphi_*(\mathbf{x})\right| - w_2\cdot\left|\varphi(\mathbf{x}) - \varphi_*(\mathbf{x})\right| d\mathbf{x} \\
&= \frac{w_2}{2}\int_{\mathbf{x}\in\mathcal{X}} \left|\varphi_{\mathrm{pro}}(\mathbf{x}) - \varphi_*(\mathbf{x})\right| - \left|\varphi(\mathbf{x}) - \varphi_*(\mathbf{x})\right| d\mathbf{x} \leq 0
\end{aligned}
\tag{5}
$$

where the inequality is obtained with the following formula:

$$
\begin{aligned}
&\left|w_1\cdot\varphi(\mathbf{x}) + w_2\cdot\varphi_{\mathrm{pro}}(\mathbf{x}) - \varphi_*(\mathbf{x})\right| \\
&= \left|w_1\cdot\varphi(\mathbf{x}) - w_1\cdot\varphi_*(\mathbf{x}) + w_2\cdot\varphi_{\mathrm{pro}}(\mathbf{x}) - w_2\cdot\varphi_*(\mathbf{x})\right| \\
&\leq w_1\cdot\left|\varphi(\mathbf{x}) - \varphi_*(\mathbf{x})\right| + w_2\cdot\left|\varphi_{\mathrm{pro}}(\mathbf{x}) - \varphi_*(\mathbf{x})\right|
\end{aligned}
\tag{6}
$$

This implies that $\tilde{\varphi}_{\mathrm{pro}}(\mathbf{x})$ has a greater overlap with $\varphi_*(\mathbf{x})$ than $\varphi(\mathbf{x})$ does. □

B.2 Proofs of Theorem 1

In the context of sequential design, let $\mathcal{F}_N$ denote the σ-algebra generated by the random variables $\mathbf{x}^1, Z^1, \ldots, \mathbf{x}^N, Z^N$ where Z^i is the observation of $f_l(\mathbf{x}^i)$. Additionally, let $\mathcal{F}_{N,\tilde{\mathbf{x}}}$ be the σ-algebra generated by $\mathbf{x}^1, Z^1, \ldots, \mathbf{x}^N, Z^N, \tilde{\mathbf{x}}, \tilde{Z}$ with $\tilde{Z}$ the observation of $f_h(\tilde{\mathbf{x}})$. Then, the EI-based sequential design of `Lamda+BO` takes the following form:

$$
\mathbf{x}_{N+1} = \arg\max_{\tilde{\mathbf{x}}\in\Omega} \mathbb{E}_N\left[M_N^0 - M_{N,\tilde{\mathbf{x}}}^{\rho}\right], \tag{7}
$$

in which y_h^* is the threshold of promising solutions in HF problems lower bounded by the α quantile of $\{f_\ell(\mathbf{x})|\forall\mathbf{x}\in\mathcal{S}\}$ due to the overlapping between HF and LF landscape, and

$$
M_{N,\tilde{\mathbf{x}}}^{\rho} = \min_{\mathbf{x}\in\mathcal{X},\, \mathbb{P}(f_h(\mathbf{x})<y_h^*|\mathcal{F}_{N,\tilde{\mathbf{x}}})=1,\, \sigma_f(\mathbf{x}|\mathcal{F}_{N,\tilde{\mathbf{x}}})=0} \tilde{f}(\mathbf{x}), \tag{8}
$$

$$
M_N^0 = \min_{\mathbf{x}\in\mathcal{X},\, \mathbb{P}(f_h(\mathbf{x})<y_h^*|\mathcal{F})=1,\, \sigma_f(\mathbf{x}|\mathcal{F}_N)=0} \tilde{f}(\mathbf{x}). \tag{9}
$$

When GPs are non-degenerate, i.e., $\sigma_f = 0$ only if $\mathbf{x}\in\mathcal{D}$ equation (7) becomes equivalent to equation (10). Specifically, M_N^0 will be equal to $f_{\mathcal{D}}^*$ in equation (11), while $M_{N,\tilde{\mathbf{x}}}^{\rho}$ will be the predicted promising HF objective value under threshold y_h^* implicitly defined by equation (1). Next, we present the criteria for *asymptotic convergence* of `Lamda+BO`. The proof of the first statement, i.e., the convergence of the acquisition function, compromises three steps.

$$
\mathbf{x}^{n+1} = \arg\max \tilde{\varphi}_{\mathrm{pro}}(\mathbf{x})\mathrm{EI}(\tilde{\mathbf{x}}|\mathcal{D}). \tag{10}
$$

$$
\mathrm{EI}(\tilde{\mathbf{x}}|\mathcal{D}) = \sigma_f(\tilde{\mathbf{x}})\big(z\Phi_f(z) + \phi_f(z)\big), \tag{11}
$$

Step 1. `Lamda+BO` serves as a stepwise uncertainty reduction (SUR) sequential design. For $N\geq 2$, a minimization version of equation (7) can be given as

$$
\mathbf{x}_{N+1} = \arg\min_{\tilde{\mathbf{x}}\in\mathcal{X}} \mathbb{E}_N\left[H_{N,\tilde{\mathbf{x}}}\right], \tag{12}
$$

in which

$$
H_{N,\tilde{\mathbf{x}}} = M_{N,\tilde{\mathbf{x}}}^{\rho} - M_N^0 = \mathbb{E}_{N,\tilde{\mathbf{x}}}\left[M_{N,\tilde{\mathbf{x}}}^{\rho} - \min_{\mathbf{x}\in\mathcal{X},\, \mathbb{P}(f_h(\mathbf{x})<y_h^*)} \tilde{f}(\mathbf{x})\right].
$$

The above equation holds since: $i)$ M_N^0 is independent from $\tilde{\mathbf{x}}$, and $ii)$ $\mathbb{E}_{N,\tilde{\mathbf{x}}}\left[M_{N,\tilde{\mathbf{x}}}^{\rho}\right] = M_{N,\tilde{\mathbf{x}}}^{\rho}$ for minimum operation. Therefore this strategy can be transformed into an equivalent SUR sequential design strategy for $H_{N,\tilde{\mathbf{x}}}$. Likewise, we define

$$
H_N = \mathbb{E}_N\left[M_N^{\rho} - \min_{\mathbf{x}\in\mathcal{X},\, f_h(\mathbf{x})<y_h^*} \tilde{f}(\mathbf{x})\right]. \tag{13}
$$

Step 2. (H_N) **is a supermartingale.** For well-structured GP models and well-defined smooth functions ρ^i, we have: $i)$ $\sigma_f(\mathbf{x}|\mathcal{F}_{N+1}) \leq \sigma_f(\mathbf{x}|\mathcal{F}_N)$ (based on the definition of GP predicted variance), and $ii)$ $\mathbb{P}(f_h(\mathbf{x}) < y_h^*(\mathbf{x})|\mathcal{F}_N) = 1$ is sufficient for $\mathbb{P}(f_h(\mathbf{x}) < y_h^*(\mathbf{x})|\mathcal{F}_{N+1}) = 1$ based on the non-increasing property of density estimation on an evaluated solution $\mathbf{x}^N$. Therefore, the following inequality holds:

$$H_N - \mathbb{E}_N[H_{N+1}] = \mathbb{E}_N\left[M_N^\rho - M_{N+1}^\rho\right] \geq 0, \tag{14}$$

which implies that $(H_N)_{N\in\mathbb{N}}$ is a supermartingale. Consequently, there is $H_N - \mathbb{E}_N[H_{N+1}] \to 0$ as $N \to \infty$, and also

$$\sup_{\tilde{\mathbf{x}}\in\mathcal{X}}\left[H_N - \mathbb{E}_N[H_{N,\tilde{\mathbf{x}}}]\right] \to 0. \tag{15}$$

Step 3. The acquisition function of `Lamda+BO` converges to zero almost surely. Due to the lower bound, according to equation (14), as evaluations of HF objective increase, $\tilde{\varphi}_{\text{pro}}$ tends to converge to φ. Note that for $N \to \infty$, $M_N^\rho = M_N^0$ as this convergence appears. Additionally, we also have

$$\sup_{\tilde{\mathbf{x}}\in\mathcal{X}}\mathbb{E}_N\left[M_N^\rho - M_{N,\tilde{\mathbf{x}}}^\rho\right] \geq \sup_{\tilde{\mathbf{x}}\in\mathcal{X}}\mathbb{E}_N\left[M_N^0 - M_{N,\tilde{\mathbf{x}}}^\rho\right] \geq \sup_{\tilde{\mathbf{x}}\in\mathcal{X}}\tilde{\varphi}_{\text{pro}}(\mathbf{x})\text{EI}(\tilde{\mathbf{x}}|\mathcal{D}). \tag{16}$$

Therefore, with the same proof as that of Proposition 2.9 (Bect, Bachoc, and Ginsbourger 2019), for $N \to \infty$, (15) and (16) yield $\tilde{\varphi}_{\text{pro}}(\mathbf{x})\text{EI}(\tilde{\mathbf{x}}|\mathcal{D}) \to 0$. This completes the proof for the first statement.

The second statement stands according to the global search ability of EI and corresponding dense evaluated solutions in $\mathcal{X}$. We complete the proof by providing the following facts: $i)$ $\tilde{\varphi}_{\text{pro}}(\mathbf{x})\sigma_f(\tilde{\mathbf{x}}) \to 0$ holds from the first statement; $ii)$ the lower bound y_h^* will be tight when $N \to \infty$; and $iii)$ $\sigma_f(z|\mathcal{F}_N) \to 0$ for all sequences accordingly. Based on these facts, the sequence is almost surely dense in $\mathcal{X}$. As a result, $f_{\mathcal{D}}^*$ from any sequence converges to $f_{\mathcal{X}}^*$ almost surely when $N \to \infty$.

C. Related Works

C.1 Learning from Previous Experiments

There are several approaches to leveraging previous experiments for improving HPO efficiency. For instance, meta-learning can warm-start the HPO process by selecting initial hyperparameter configurations that have performed well in similar tasks or are generally known to perform efficiently (Feurer, Springenberg, and Hutter 2015; Lindauer and Hutter 2018). Alternatively, transfer learning trains joint surrogate models across multiple tasks, enabling knowledge transfer as demonstrated by multi-task optimization (Perrone et al. 2018) and few-shot learning formulations (Wistuba and Grabocka 2021). In addition, these methods can leverage data from previous experiments to refine the search space (Wistuba, Schilling, and Schmidt-Thieme 2015; Perrone and Shen 2019; Li et al. 2022a), thereby offering a warm start for HPO. For instance, Wistuba, Schilling, and Schmidt-Thieme (2015) pruned the bad regions of search space according to the results from previous tasks. Perrone and Shen (2019) and Li et al. (2022a) utilized previous tasks to design a sub-region of the entire search space for a new task. While these approaches have demonstrated efficiency in using promising regions instead of the entire space, they require the preparation of source tasks and the evaluation of task similarities to effectively select relevant tasks for learning promising regions, which can be challenging (Perrone and Shen 2019).

C.2 Incorporating Expert Priors for Optimization

These methods concentrate on promising regions by leveraging prior knowledge from experts. A line of work uses prior information about locations of optimal solutions to accelerate the process of HPO. For instance to reduce evaluations on bad regions, Souza et al. (2021) injected priors about which parts of the input space will yield the best performance into BO's standard probabilistic model to form a pseudo-posterior, which was shown to be more sample-efficient than BO baselines. Further, Li et al. (2020a) incorporated Gaussian distributions of optimal solutions into the posterior distribution of observed data and used Thompson sampling to obtain the next solution. Ramachandran et al. (2020) used the prior distribution of optimal solutions to warp the search space, expanding around high-probability regions of optimal solutions and shrinking around low-probability regions. Truncated normal and gamma distributions were used to form the prior distributions. Additionally, Hvarfner et al. (2022) incorporated prior Gaussian distributions about locations of optimal solutions into the acquisition function, achieving competitive results across a wide range of benchmarks. Mallik et al. (2023) also integrated prior knowledge of optimal hyperparameters to enhance the efficiency of Hyperband. Although using prior distributions of locations of optimal solutions can accelerate optimization, accessing the prior for a specific task may not always be accessible.

C.3 Multi-fidelity Bayesian optimization

In the realm of MFBO, previous research primarily leverages LF to construct an accurate MF model for guiding the sampling process (Swersky, Snoek, and Adams 2013; Poloczek, Wang, and Frazier 2017; Kandasamy et al. 2019; Mikkola et al. 2022; Li et al. 2020b). One challenge in these methods is to model performance using data from various fidelities. Solutions include employing Gaussian process regression (GPR) models tailored to each fidelity level (Kandasamy et al. 2019), multi-task GPR

Algorithm 1: Pseudocode for `Lamda`

Input: Total budget Λ, maximum first-phase budget B, configuration parameters l
Output: Final solution x^*
// First-phase search
$(\varphi_{\text{pro}}(\mathbf{x}), S, \Lambda_l) \leftarrow$ `Lamda-1`(B, l)
$\Lambda \leftarrow \Lambda - \Lambda_l$
// Second-phase search
$x^* \leftarrow$ `Lamda-2`$(\varphi_{\text{pro}}(\mathbf{x}), \Lambda)$
return x^*

models for discrete fidelity levels (Swersky, Snoek, and Adams 2013; Poloczek, Wang, and Frazier 2017; Mikkola et al. 2022; Li et al. 2020b), and GPR models for a continuous fidelity space (Klein et al. 2017; Kandasamy et al. 2017). While Gaussian processes are commonly used for modelling the surrogate function, Li et al. (Li et al. 2020b; Li, Kirby, and Zhe 2021) have implemented deep neural networks to represent the relationships across different fidelities. In terms of sampling, entropy search methods (Swersky, Snoek, and Adams 2013; Poloczek, Wang, and Frazier 2017; Takeno et al. 2020) are utilized, which take into account both the information gain and the associated costs of each fidelity level. These techniques enable the sampling of new solutions at either low or high fidelity levels, gradually leading to improved hyperparameters in the HF space.

C.4 Bandit Based Multi-fidelity Method

Bandit-based methods employ LF for identifying promising solutions for HF evaluations (Falkner, Klein, and Hutter 2018; Li et al. 2022b; Awad, Mallik, and Hutter 2021). In pioneering work in the field, Li et al. (2017) introduced Hyperband, a method that improves upon the Successive Halving (SH) algorithm by integrating various early stopping strategies across multiple SH brackets. Each bracket starts with a different number of solutions at varied fidelity levels. However, Hyperband's approach of randomly sampling solutions doesn't utilize previous sample information (Falkner, Klein, and Hutter 2018). To enhance this, Falkner, Klein, and Hutter (2018) developed BOHB, which combines BO with Hyperband, using the tree Parzen estimator (TPE) (Bergstra et al. 2011) to build surrogate models for each fidelity level. While BOHB is effective, it struggles with discrete dimensions and scaling to high-dimensional problems. Addressing these challenges, Awad, Mallik, and Hutter (2021) enhanced BOHB with differential evolution for better candidate sampling. Additionally, Li et al. (2021) developed an MF ensemble model that integrates information from all fidelity levels to more accurately estimate the highest fidelity. Nevertheless, despite their utilization of LF data, these approaches still demand extensive exploration throughout the entire search space.

D. Algorithm details

In this section, we outline the workflow of our `Lamda` framework, which operates in two phases: the first-phase low-fidelity search (dubbed as `Lamda-1`) and the second-phase optimization (dubbed as `Lamda-2`). The overall workflow is depicted in Algorithm 1. The first phase, `Lamda-1`, focuses on learning prior in the LF landscape. The pseudocode for `Lamda-1` is provided in Algorithm 2. In particular, the sampling strategy in `Lamda-1` is algorithm-agnostic and can be incorporated with most HPO algorithms. The second phase, `Lamda-2`, allows the use of the prior into different algorithms to guide the search. We provide multiple pseudocode to demonstrate how `Lamda-2` can be adapted to various algorithms, including `PriorBand`, `BOHB`, `MUMBO`, vanilla `BO`, and `random search`. The details for each integration are outlined below, with modifications from the original algorithms highlighted in red:

- For `PriorBand`, `Lamda-2` replaces the expert prior $p_\pi(\mathbf{x})$ from (Mallik et al. 2023) with the learned prior $\varphi_{\text{pro}}(\mathbf{x})$ obtained in `Lamda-1`.
- For `BOHB`, `Lamda-2` modifies the uniform sampling distribution of Algorithm 3 into the incumbent distribution determined by $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$.
- For `MUMBO` and `BO`, `Lamda-2` combines $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$ with their acquisition functions. The pseudocode for these adaptations is shown in Algorithm 4 and Algorithm 5, respectively.
- For `Random Search`, `Lamda-2` replaces the uniform sampling strategy by the incumbent sampling strategy defined by $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$, as illustrated in Algorithm 6.

To formalize the concept of multi-fidelity HPO, we introduce $f_z(\cdot)$ with $z \in \ell, \ell+1, \ldots, h$ to denote the computation of $f(\cdot)$ at fidelity level z, such as the validation loss of a model trained for z epochs. Here, f_h and f_ℓ (where $\ell < h$) represent the HF and LF objectives, respectively.

E. Benchmarks

E.1 Tabular benchmarks

FCNet: We utilized benchmarks for FCNet from Yahpo-Gym (Pfisterer et al. 2022), detailed in Table 1. The selected tasks include FCNet-Naval-Propulsion, FCNet-Protein-Structure, FCNet-Slice-Localization, and FCNet-Parkinsons-Telemonitoring.

Algorithm 2: Pseudocode for `Lamda-1`

Input: Maximum first-phase budget B, threshold y^*, Δ, γ, fidelity level l, budget function λ_z
Output: PDF of the promising solutions $\varphi_{\text{pro}}(\mathbf{x})$, observation set S, consumed budget Λ_l
Initialize: $S \leftarrow \emptyset$, $\Lambda_l \leftarrow 0$, $t \leftarrow 0$, `isStable` $\leftarrow$ **False**
while $\Lambda_l < B$ **and not** `isStable` **do**
  $\mathbf{x}^t \leftarrow \arg\max_{\mathbf{x}\in\mathcal{X}} \mathrm{AF}(\mathbf{x}, \mathcal{S})$
  $y^t \leftarrow f_\ell(\mathbf{x}^t) + \epsilon$
  $S \leftarrow S \cup \{(\mathbf{x}^t, y^t)\}$
  Update $\mathcal{S}_{\text{pro}}$, $\mathcal{S}_{\text{inf}}$, $\varphi^t_{\text{pro}}(\mathbf{x})$, $\varphi^t_{\text{inf}}(\mathbf{x})$
  if $1 - \rho(\varphi^t_{\text{pro}}(\mathbf{x}), \varphi^{t+\Delta}_{\text{pro}}(\mathbf{x})) \leq \gamma$ **then**
    `isStable` $\leftarrow$ **True**
  end if
  $\Lambda_l \leftarrow \Lambda_l + \lambda_z(\mathbf{x}^t, l)$
  $t \leftarrow t + 1$
end while
$\varphi_{\text{pro}}(\mathbf{x}) \leftarrow \varphi^t_{\text{pro}}(\mathbf{x})$
return $(\varphi_{\text{pro}}(\mathbf{x}), S, \Lambda_l)$

Algorithm 3: Pseudocode for sampling in `Lamda+BOHB`

Input: Observations D, fraction of random runs ρ, percentile q, number of samples N_s, minimum number of points N_{min} to build GP models, and bandwidth factor b_w
Output: Next configuration to evaluate
Initialization: $b \leftarrow \arg\max\{D_b : |D_b| \geq N_{\text{min}} + 2\}$; $\tilde{\rho} \leftarrow \mathrm{Rand}(0, 1)$
if $\tilde{\rho} < \rho$ **or** $b = \emptyset$ **then**
  return randomly sampled configuration
else
  Compute $l(\mathbf{x})$ and $g(\mathbf{x})$ as Eqs. (2) and (3) in (Falkner, Klein, and Hutter 2018)
  Draw N_s configurations according to $\tilde{\varphi}_{\text{pro}}(\mathbf{x})$ in equation (1)
  return configuration with highest ratio $l(\mathbf{x})/g(\mathbf{x})$
end if

NAS-Bench-301: Our NAS-Bench-301 benchmarks, also sourced from Yahpo-Gym (Pfisterer et al. 2022), focus on CIFAR-10. For hyperparameter details, refer to (Pfisterer et al. 2022).

NAS-Bench-201: This benchmark encompasses 6 hyperparameters for neural architecture search. It includes statistics from 15,625 CNN models across three datasets: CIFAR-10-valid, CIFAR-100, and ImageNet16-120 (Eggensperger et al. 2021). We simplify their name as CIFAR-10, CIFAR-100, and ImageNet in this paper.

Table 1: Hyperparameter ranges for FCNet.

Parameter	Name	Type	Range
Fidelity	epoch	int	$[1, 100]$
Hyperparameter	batch size	int	$[8, 64]$ (log-scale)
	initial learning rate	con	$[5\text{e}-04, 0.1]$ (log-scale)
	dropout 1	con	$[0.0, 0.6]$
	dropout 2	con	$[0.0, 0.6]$
	number of units 1	int	$[16, 512]$
	number of units 2	int	$[16, 512]$

E.2 Surrogate benchmarks

We utilized four problems from the PD1 benchmarks and three from the JAHSBench surrogate benchmarks. Detailed information about these benchmarks is available in (Mallik et al. 2023).

Raw problems

The hyperparameters for the deep neural networks, LeNet and ResNet-18, are detailed in Table 2.

Algorithm 4: Second-phase search with BO

Input: Input space $\mathcal{X}$, $\varphi_{\text{pro}}(\mathbf{x})$, w, N solution for the initial design of GPs, budget Λ_r, fidelity level h, budget function λ_z.
Output: Optimized design x^*.
Initialization: Sample $\{\mathbf{x}^i\}_{i=1}^n$ from distribution given by $\varphi_{\text{pro}}(\mathbf{x})$; $y^i \leftarrow f_h(\mathbf{x}^i) + \epsilon^i$, where $\epsilon^i \sim \mathcal{N}(0, \sigma^2)$; $\lambda^i \leftarrow \lambda_z(\mathbf{x}^i, h)$; $\Lambda_r \leftarrow \Lambda_r - \sum_{i=1}^n \lambda^i$; $D \leftarrow \{(\mathbf{x}^i, y^i)\}_{i=1}^n$
while $\Lambda_r > 0$ **do**
  $\varphi(\mathbf{x}) \leftarrow p(\mathbf{x}|D)$
  $\tilde{\varphi}_{\text{pro}}(\mathbf{x}) \leftarrow (1-w) \cdot \varphi(\mathbf{x}) + w \cdot \varphi_{\text{pro}}(\mathbf{x})$
  $\mathbf{x}^{n+1} \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}} \tilde{\varphi}_{\text{pro}}(\mathbf{x}) \text{AF}(\mathbf{x}, \mathcal{D})$
  $y^{n+1} \leftarrow f_h(\mathbf{x}^{n+1}) + \epsilon$
  Update $D \leftarrow D \cup \{(\mathbf{x}^{n+1}, y^{n+1})\}$
  $\Lambda_r \leftarrow \Lambda_r - \lambda_z(\mathbf{x}^{n+1}, h)$
  $n \leftarrow n + 1$
end while
return $x^* \leftarrow \arg\min_{(\mathbf{x}^i, y^i) \in D} y^i$

Algorithm 5: Second-phase search with MUMBO

Input: Input space $\mathcal{X}$, prior obtained in the first phase $\varphi_{\text{pro}}(\mathbf{x})$, prior confidence parameter w, size n of the initial design, budget for first phase Λ_r.
Output: Optimized design x^*.
Initialization: Sample $\{\mathbf{x}^i\}_{i=1}^n \sim \varphi_{\text{pro}}(\mathbf{x})$ and randomly assign fidelity levels $\{z^i\}_{i=1}^n$ with $z^i \sim \text{Uniform}(\{\ell, \ell+1, \ldots, h\})$; Compute $y^i \leftarrow f_z(\mathbf{x}^i, z^i) + \epsilon^i$, where $\epsilon^i \sim \mathcal{N}(0, \sigma^2)$; $\lambda^i \leftarrow \lambda_z(\mathbf{x}^i, z^i)$; $\Lambda_r \leftarrow \Lambda_r - \sum_{i=1}^n \lambda^i$; $D \leftarrow \{(\mathbf{x}^i, z^i), y^i\}_{i=1}^n$
while $\Lambda_r > 0$ **do**
  Fit GP to the collected observations D, $\varphi(\mathbf{x}) \leftarrow p(\mathbf{x}|D)$
  Simulate N samples of $g^*|D$
  Prepare $\alpha_{n-1}^{\text{MUMBO}}(\mathbf{x}, z)$ as given by Eq. (5) in (Moss, Leslie, and Rayson 2020)
  Update $\tilde{\varphi}_{\text{pro}}(\mathbf{x}) \leftarrow (1-w) \cdot \varphi(\mathbf{x}) + w \cdot \varphi_{\text{pro}}(\mathbf{x})$
  Find the next point and fidelity to query:
  $(\mathbf{x}^{n+1}, z^{n+1}) \leftarrow \arg\max_{(\mathbf{x}, z)} \tilde{\varphi}_{\text{pro}}(\mathbf{x}) \frac{\alpha_{n-1}^{\text{MUMBO}}(\mathbf{x}, z)}{\lambda_z(\mathbf{x}, z)}$
  Collect the new evaluation $y^{n+1} \leftarrow f_z(\mathbf{x}^{n+1}, z^{n+1}) + \epsilon^{n+1}$, $\epsilon^{n+1} \sim \mathcal{N}(0, \sigma^2)$
  Append new evaluation to observation set $D \leftarrow D \cup \{(\mathbf{x}^{n+1}, z^{n+1}), y^{n+1}\}$
  Update spent budget $\Lambda_r \leftarrow \Lambda_r - \lambda_z(\mathbf{x}^{n+1}, z^{n+1})$
end while
return $x^* \leftarrow \arg\min_{\{((\mathbf{x}^i, z^i), y^i) \in D, z^i = h\}} y^i$

Table 2: Hyperparameter ranges for LeNet and ResNet-18.

Parameter	Name	Type	Range
Fidelity	datasize	con	$[0.3, 1]$
Hyperparameter	batch size	int	$[64, 512]$ (log-scale)
	initial learning rate	con	$[5\text{e}-3, 0.1]$ (log-scale)
	momentum	con	$[0.5, 0.99]$
	weight decay	con	$[1\text{e}-5, 1\text{e}-2]$
	nesterov	cat	{True, False}

E.3 Parameters of LF problems

Table 3 shows parameters of LF problems used in the first phase and the corresponding OVL between high- and low-fidelity problems.

Algorithm 6: Second-phase search with Random Search

Input: Input space $\mathcal{X}$, prior obtained in the first phase $\varphi_{\text{pro}}(\mathbf{x})$, prior confidence parameter w, budget for first phase Λ_r, uniform distribution p_U.
Output: Optimized design x^*.
Initialization: $\varphi(\mathbf{x}) \leftarrow p_U(\mathbf{x})$
while $\Lambda_r > 0$ **do**
  $\tilde{\varphi}_{\text{pro}}(\mathbf{x}) \leftarrow (1-w) \cdot \varphi(\mathbf{x}) + w \cdot \varphi_{\text{pro}}(\mathbf{x})$
  Sample $\mathbf{x}^{n+1} \sim \tilde{\varphi}_{\text{pro}}(\mathbf{x})$
  $y^{n+1} \leftarrow f_h(\mathbf{x}^{n+1}) + \epsilon$
  Update $D \leftarrow D \cup \{(\mathbf{x}^{n+1}, y^{n+1})\}$
  $\Lambda_r \leftarrow \Lambda_r - \lambda_z(\mathbf{x}^{n+1}, h)$
end while
return $x^* \leftarrow \arg\min_{(\mathbf{x}^i, y^i) \in D} y^i$

Table 3: Parameters of LF problems used in the first phase and the corresponding OVL between high- and low-fidelity problems.

Tasks	LF parameter	OVL	Tasks	LF parameter	OVL
MFH3	epoch=4	0.713	MFH6	epoch=4	0.728
XGBoost-40981	datasize=0.3	0.885	FCNet-Naval-Propulsion	epoch=4	0.901
XGBoost-41146	datasize=0.3	0.933	FCNet-Protein-Structure	epoch=4	0.694
XGBoost-1489	datasize=0.3	0.805	FCNet-Slice-Localization	epoch=4	0.731
XGBoost-1067	datasize=0.3	0.883	FCNet-Parkinsons-Telemonitoring	epoch=4	0.854
rpart-40981	datasize=0.3	0.167	NAS-Bench-301-CIFAR-10	epoch=20	0.712
rpart-41146	datasize=0.3	0.488	NAS-Bench-201-CIFAR-10	epoch=20	0.760
rpart-1089	datasize=0.3	0.89	NAS-Bench-201-CIFAR-100	epoch=20	0.727
rpart-1067	datasize=0.3	0.642	NAS-Bench-201-ImageNet	epoch=20	0.696
ranger-40981	datasize=0.3	0.711	JAHS-CIFAR-10	epoch=4	0.574
ranger-41146	datasize=0.3	0.782	JAHS-Colorectal-Histology	epoch=4	0.523
ranger-1489	datasize=0.3	0.77	JAHS-Fashion-MNIST	epoch=4	0.532
ranger-1067	datasize=0.3	NAN	PD1-LM1B	epoch=30	0.934
glmnet-40981	datasize=0.3	0.294	PD1-WMT	epoch=4	0.926
glmnet-41146	datasize=0.3	0.962	PD1-CIFAR-100	epoch=45	0.797
glmnet-1489	datasize=0.3	0.918	PD1-ImageNet	epoch=20	0.727
glmnet-1067	datasize=0.3	0.648	NAN	NAN	NAN

E.4 OVL between low and high fidelity

Figure 3 presents the overlapping coefficients between good solutions in high- and low-fidelity settings across various HPO tasks. The overlapping coefficients are computed using $\rho(\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x})) = \int_{\mathbf{x}\in\mathcal{X}} \min\{\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x})\}\, d\mathbf{x}$ and are visualized over epochs or datasets for different benchmarks. It can be observed that the overlap generally increases with the number of iterations (e.g., epochs or dataset size), especially for FCNet, NAS-Bench-201, and RecNet-18. However, for NAS-Bench-301 and JAHS, the overlap exhibits a trend of decreasing in the middle stages before rising again. Specifically:

- FCNet and RecNet-18 show an overlap that is already close to or greater than 0.7 even at smaller iteration parameters (e.g., fewer epochs or smaller datasets), indicating strong LF-HF consistency and stability at earlier stages.
- For PD1 and NAS-Bench-201, the overlap reaches or exceeds 0.7 at specific points, such as epoch = 10 for PD1 and epoch = 20 for NAS-Bench-201, suggesting that these tasks achieve good LF-HF agreement relatively early in the optimization process.
- NAS-Bench-301 and JAHS, on the other hand, maintain relatively low overlap in the initial and middle stages. The overlap only increases significantly as the settings approach the HF level, indicating that these tasks require longer training or higher fidelity to achieve substantial LF-HF alignment.

F. Experimental results

F.1 Effectiveness of using `Lamda` in existing HPO algorithms

Figure 4 shows the learned priors on `JAHS-CIFAR-10`. 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. In these cases, the learned prior tends to resemble a near-uniform distribution and thus provides little guidance. However, it does not mislead the

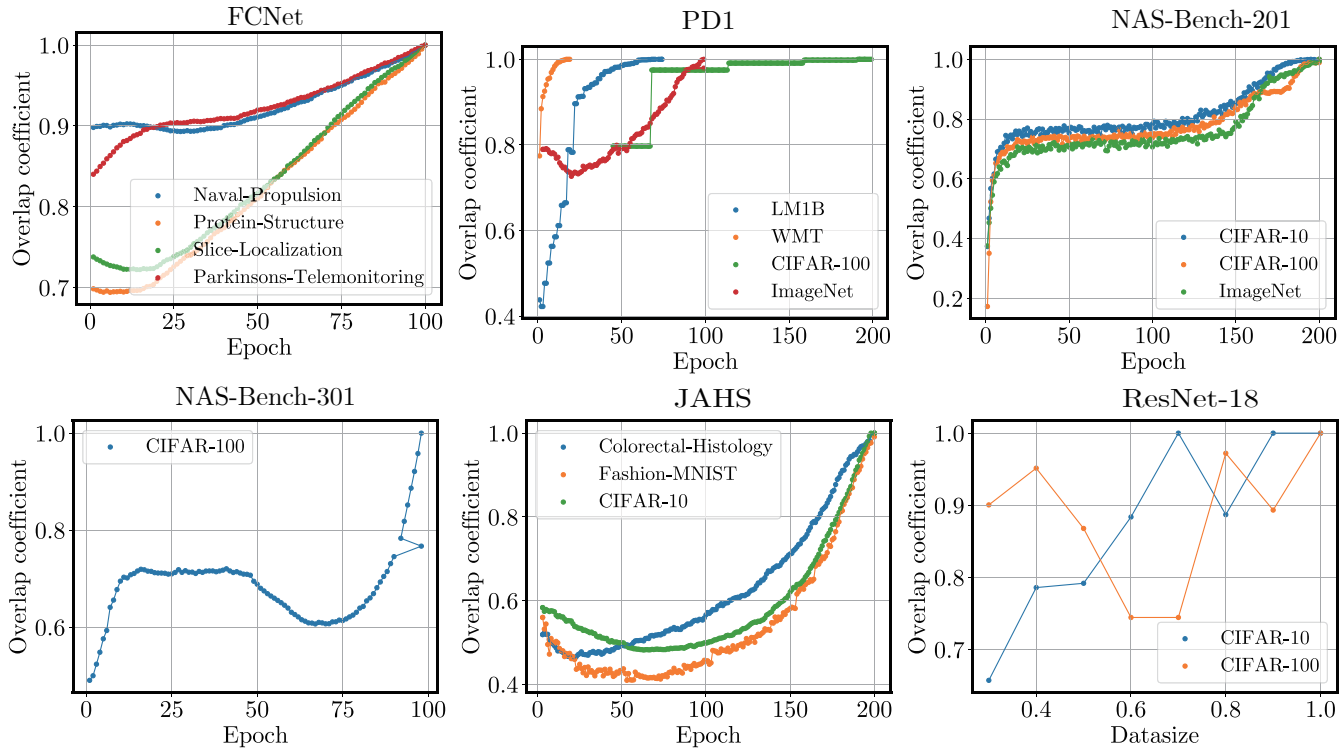

Figure 3: The overlapping coefficient of the HPO tasks.

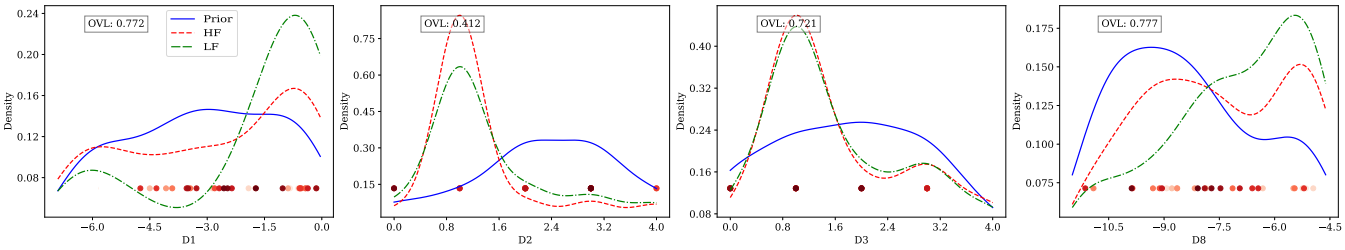

Figure 4: Prior–solution alignment and its impact on HF sampling in `JAHS-CIFAR-10`. Each subplot corresponds to one hyperparameter dimension, showing the learned prior (blue), the PDFs of top solutions at high and low fidelity (red and green), and the HF sampling trajectory (red points). Higher prior-HF overlap leads to early concentration in high-quality regions, while low-overlap dimensions show more dispersed early sampling that is progressively refined. The red points—gradually shifting from light to dark red—indicate the temporal order of HF evaluations.

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.

The convergence performance of `Lamda` under different algorithms, including `PriorBand`, `BOHB`, `MUMBO`, `BO`, and `RS`, is illustrated in Figure 5, Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12 and Figure 13. The experimental results demonstrated that using `Lamda` can enhance peformanc of the baseline algorithms.

F.2 Impact of Computational Budget on `Lamda` Performance

Figure 14 shows the validation errors of `Lamda+BOHB` under varying initial budgets B, illustrating how computational allocation affects optimization performance on `FCNet-Naval-Propulsion`. Figure 15 visualizes the learned prior distributions from `Lamda`'s first phase under different budget levels on `FCNet`, represented as probability density functions. Figure 16 shows the validation errors of `Lamda+BO` on `JAHS-CIFAR-10`, extending the budget analysis to a different task domain..

F.3 Comparison with State-of-the-Art HPO Methods

F.3.1 Results on tabular, surrogate and synthetic benchmarks This section presents the convergence curves of the proposed algorithm and its peer methods. The convergence curves of peer algorithms are presented in Figure 17 and Figure 18. `Lamda` also

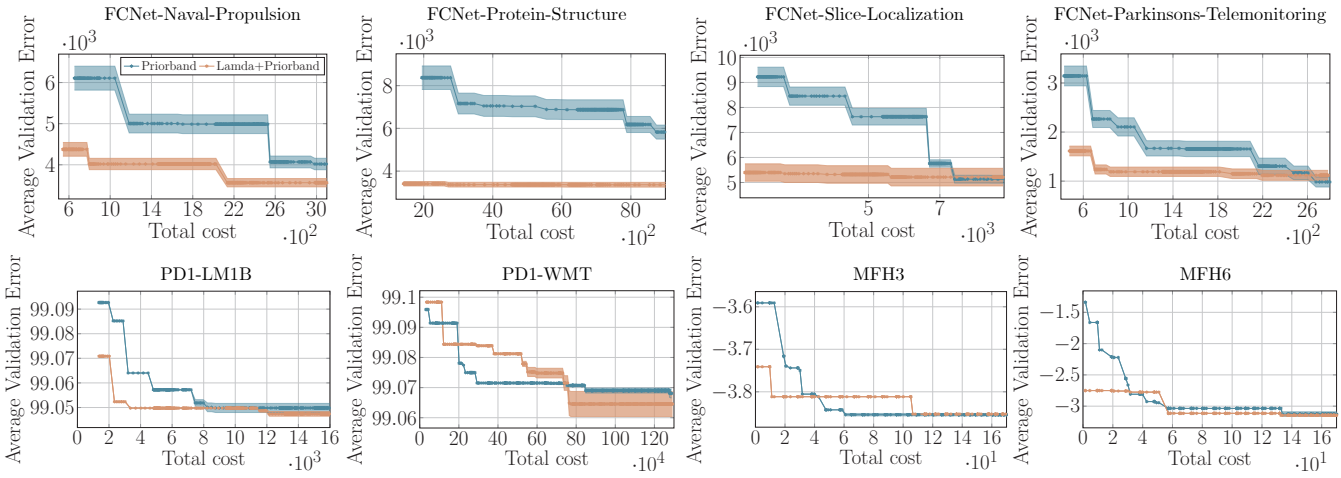

Figure 5: Validation error observed in tuning 8 HPO tasks, using `PriorBand` as the baseline.

performs good compared with peer algorithms. Table 4 shows the peer algorithms' final validation errors of the current incumbent at 100 HF evaluations horizons.

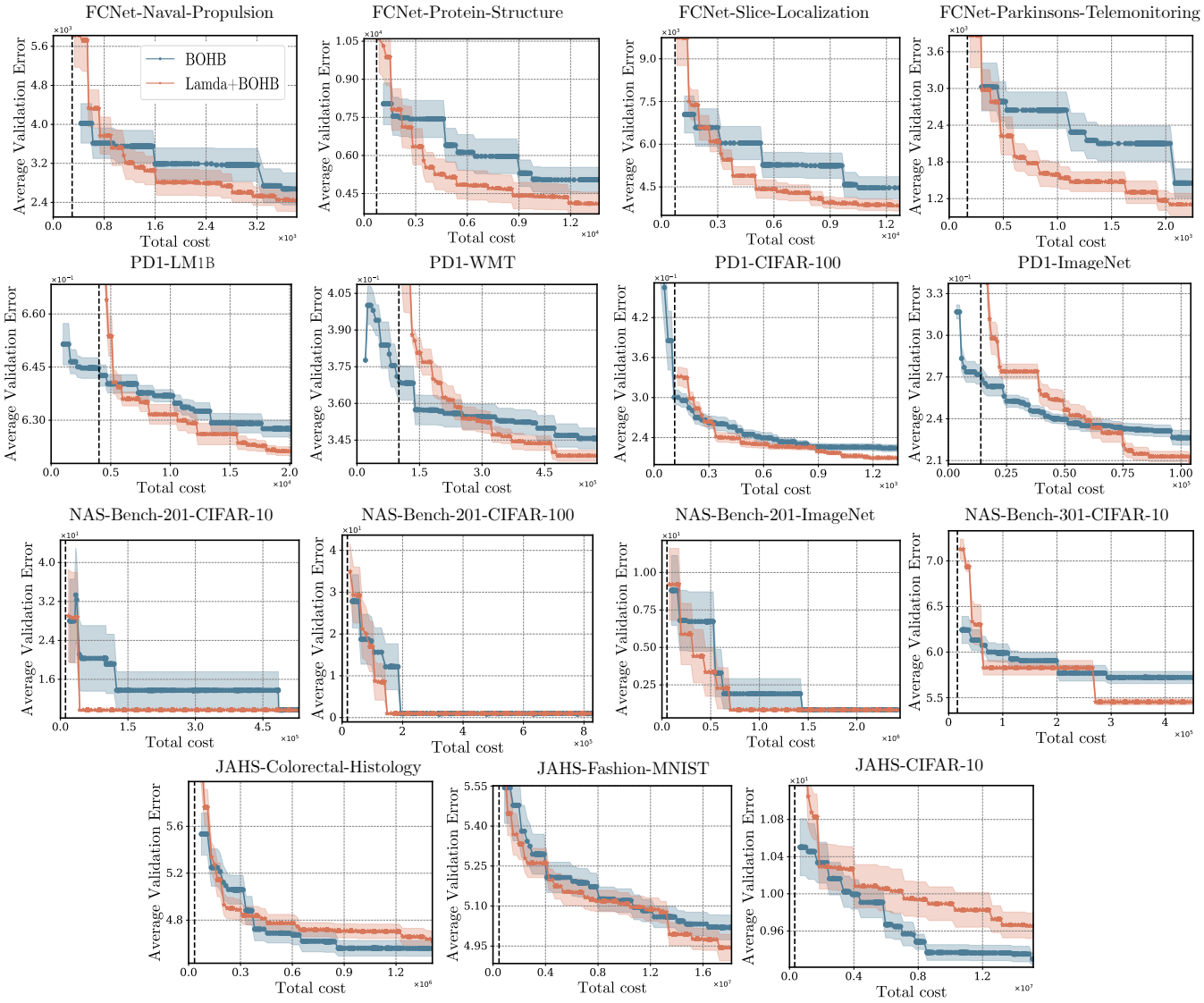

Figure 6: Validation error observed in tuning 15 HPO tasks, using `BOHB` as the baseline.

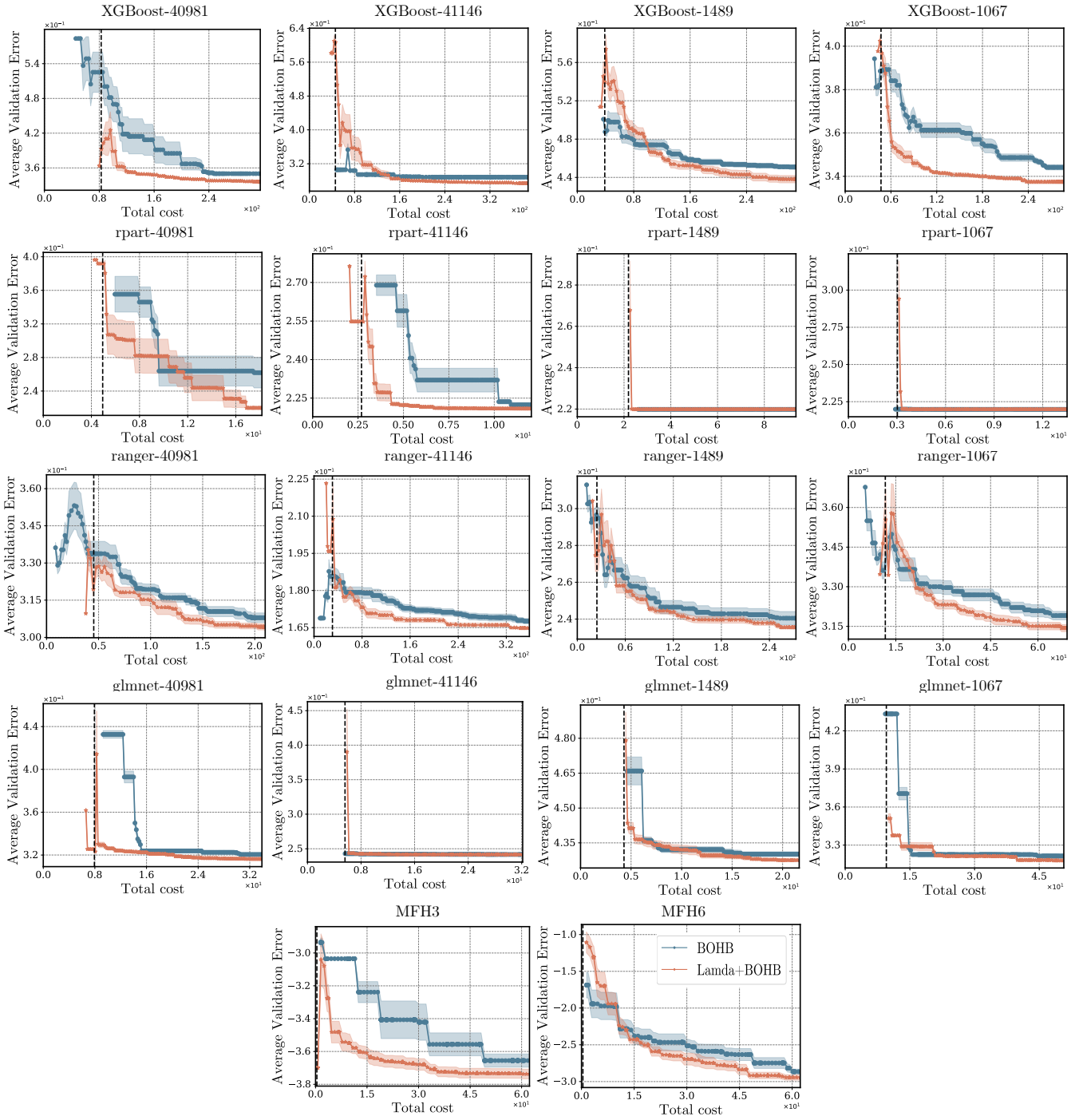

Figure 7: Validation error observed in tuning 18 HPO tasks, using `BOHB` as the baseline.

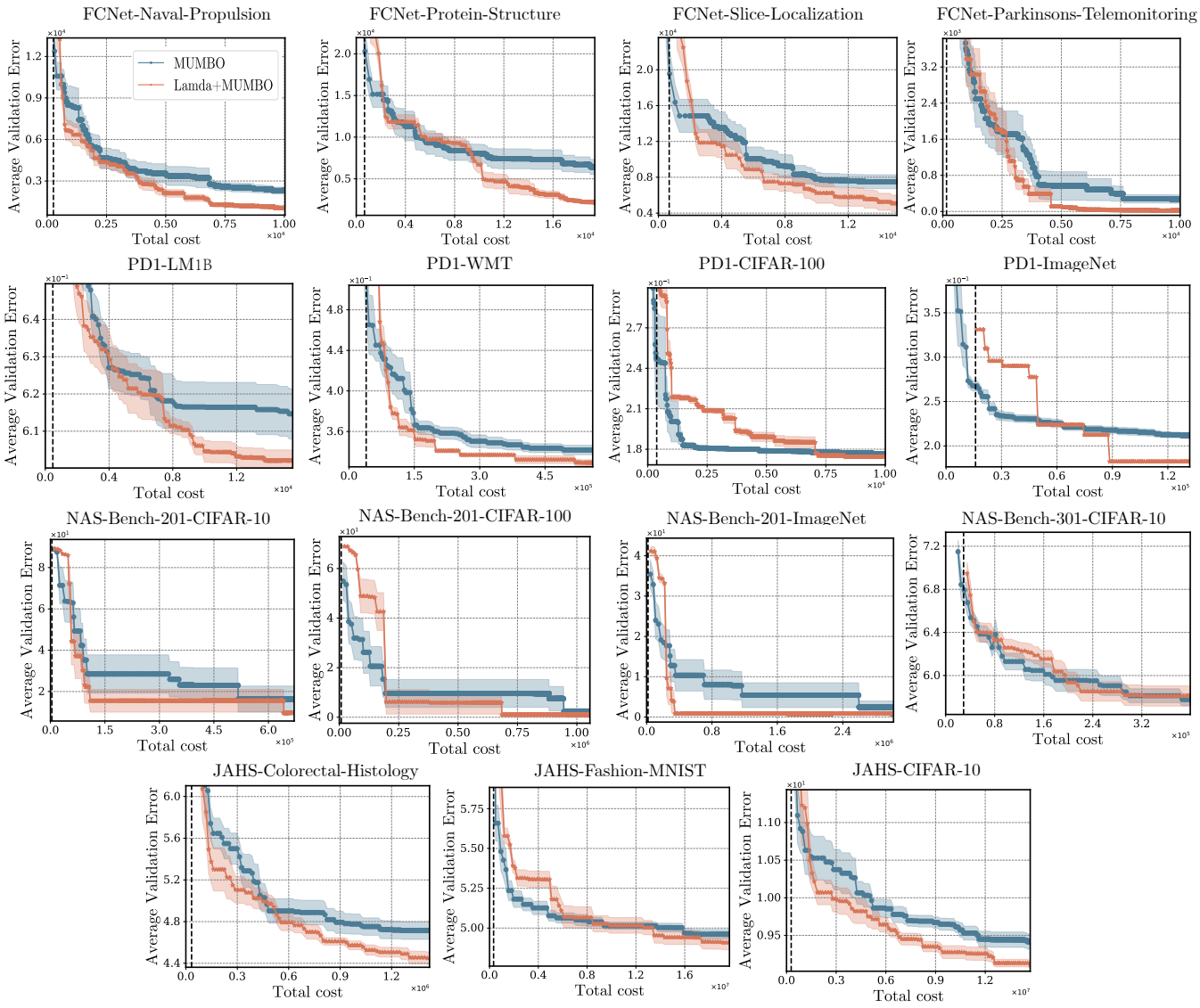

Figure 8: Validation error observed in tuning 15 HPO tasks, using `MUMBO` as the baseline.

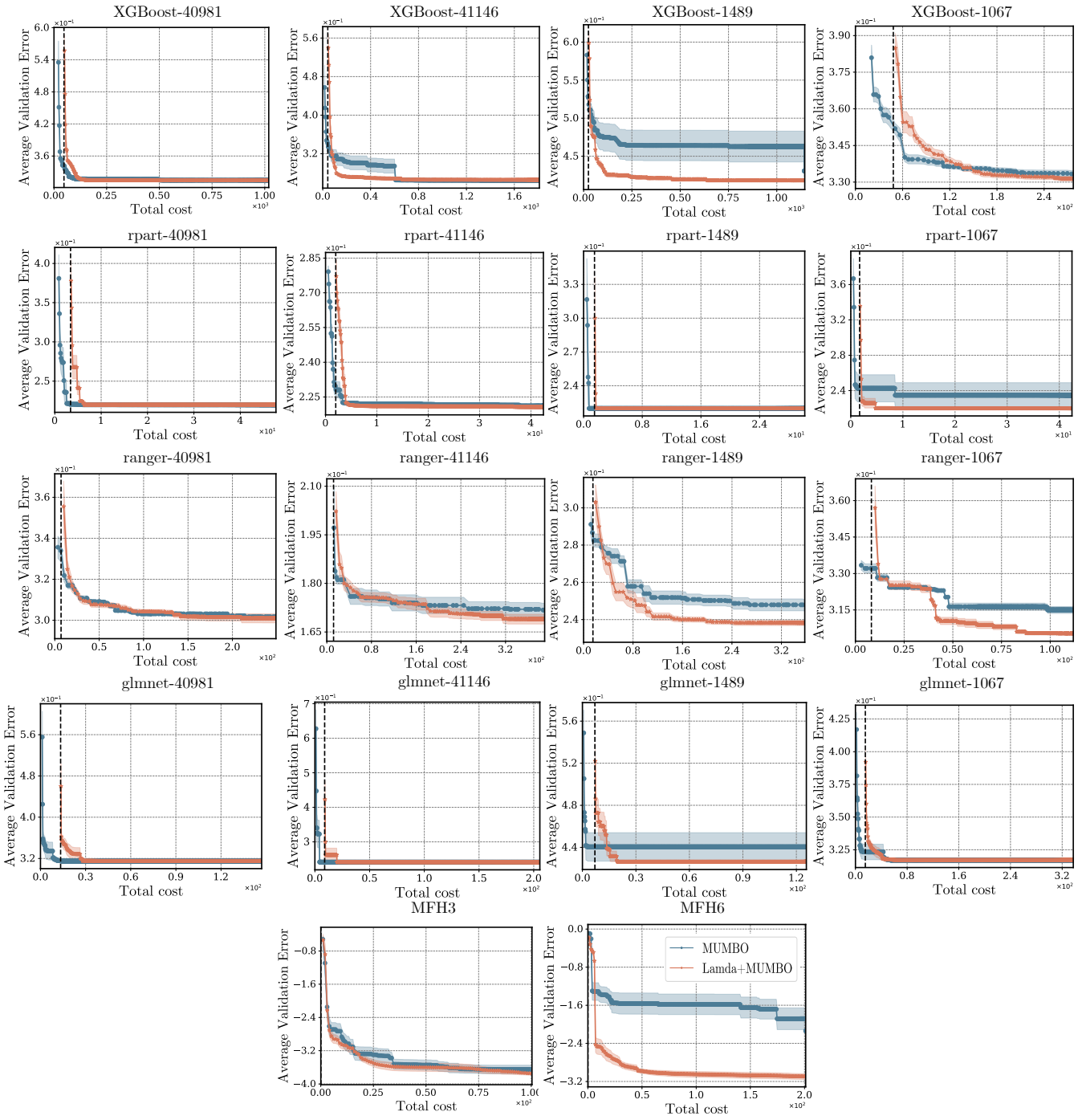

Figure 9: Validation error observed in tuning 18 HPO tasks, using MUMBO as the baseline.

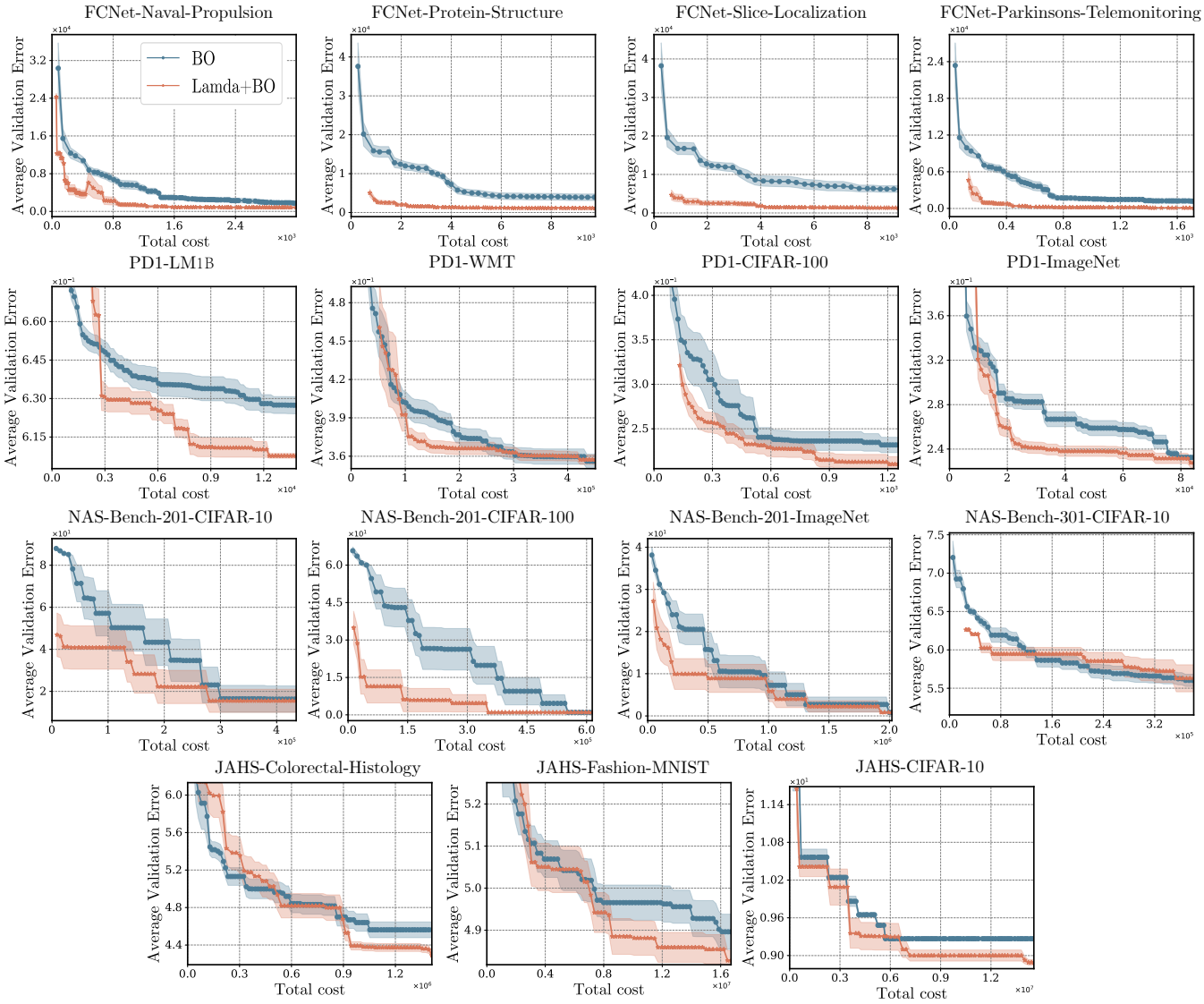

Figure 10: Validation error observed in tuning 15 HPO tasks, using BO as the baseline.

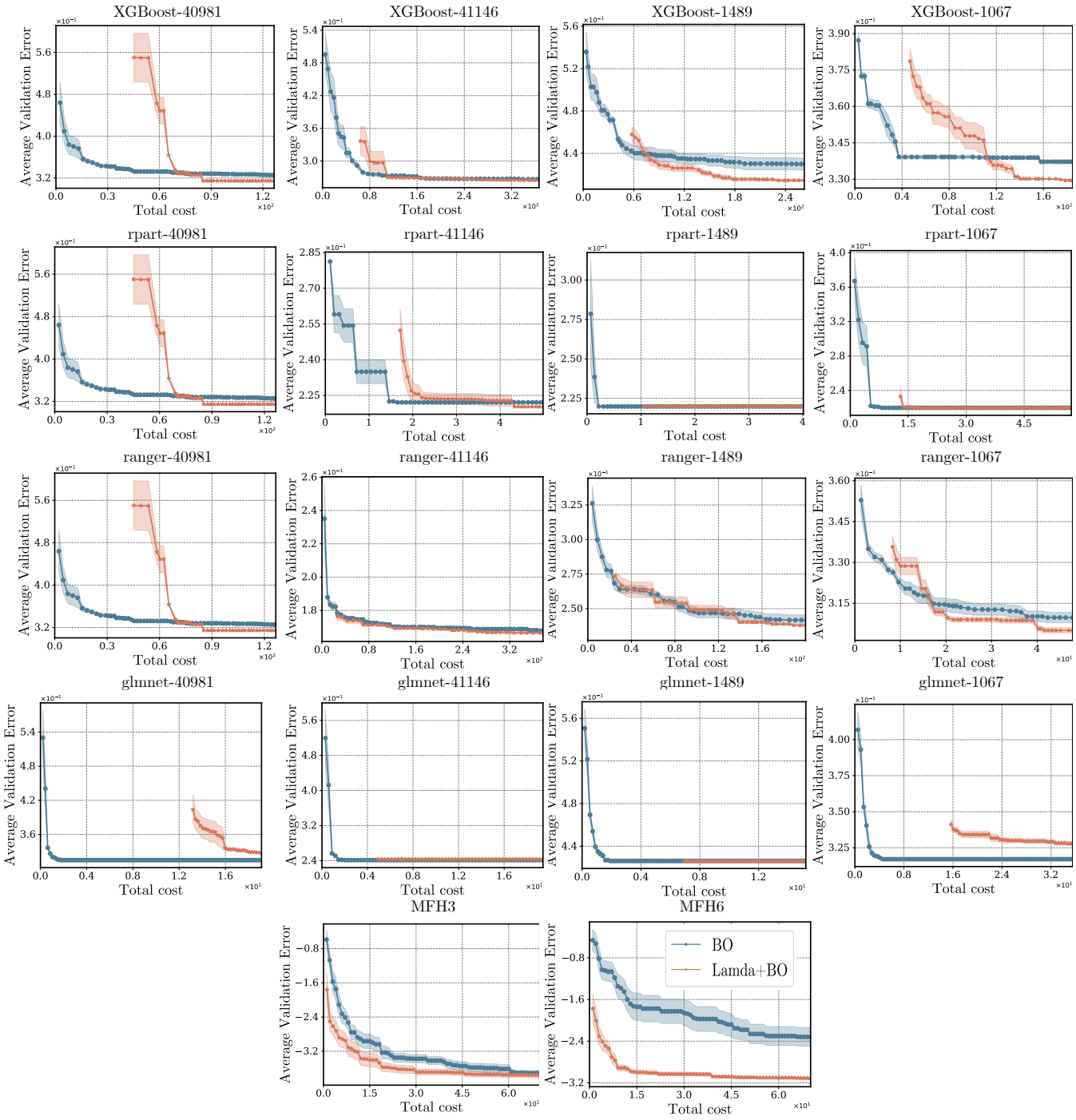

Figure 11: Validation error observed in tuning 18 HPO tasks, using BO as the baseline.

.

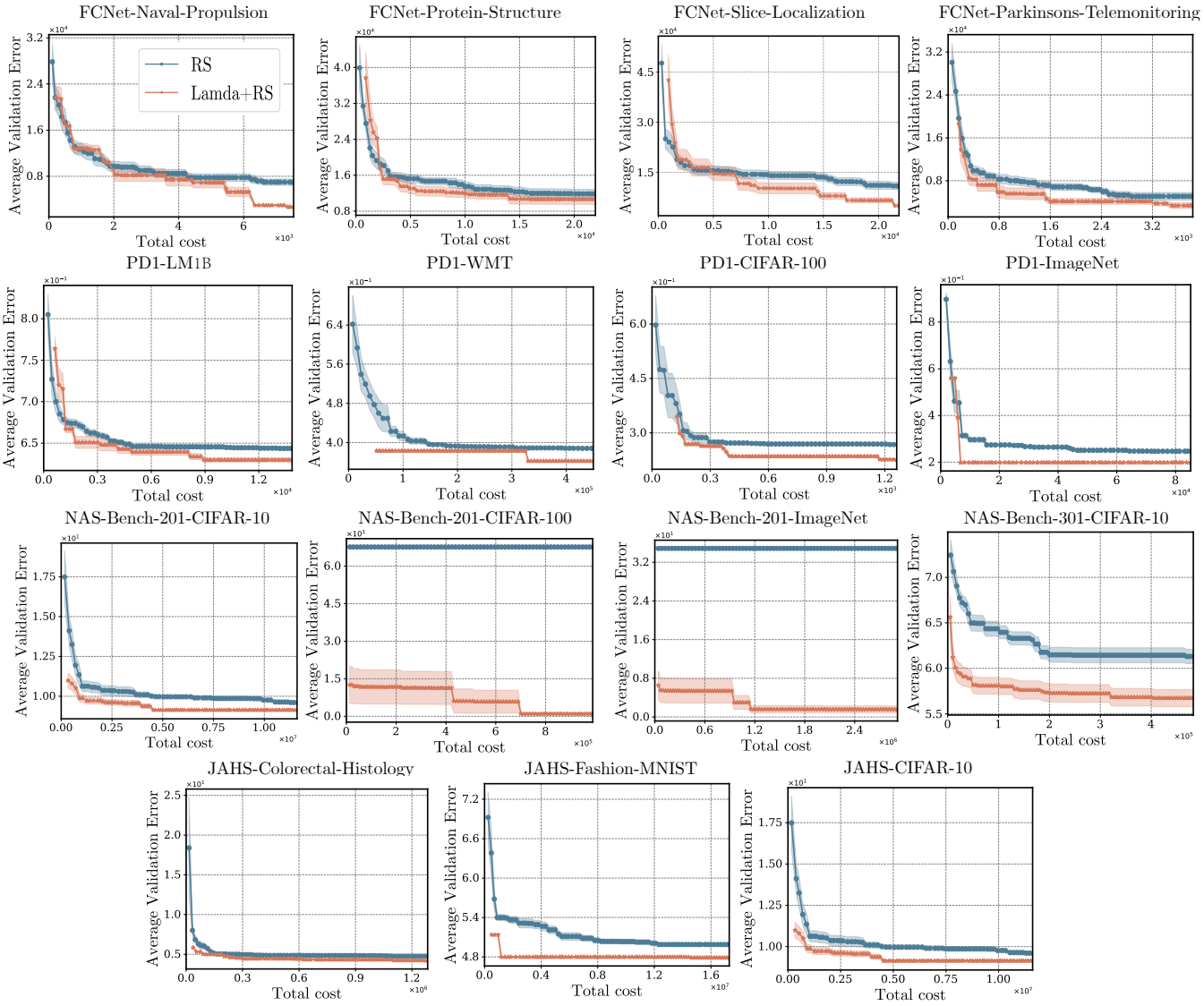

Figure 12: Validation error observed in tuning 15 HPO tasks, using `RS` as the baseline.

.

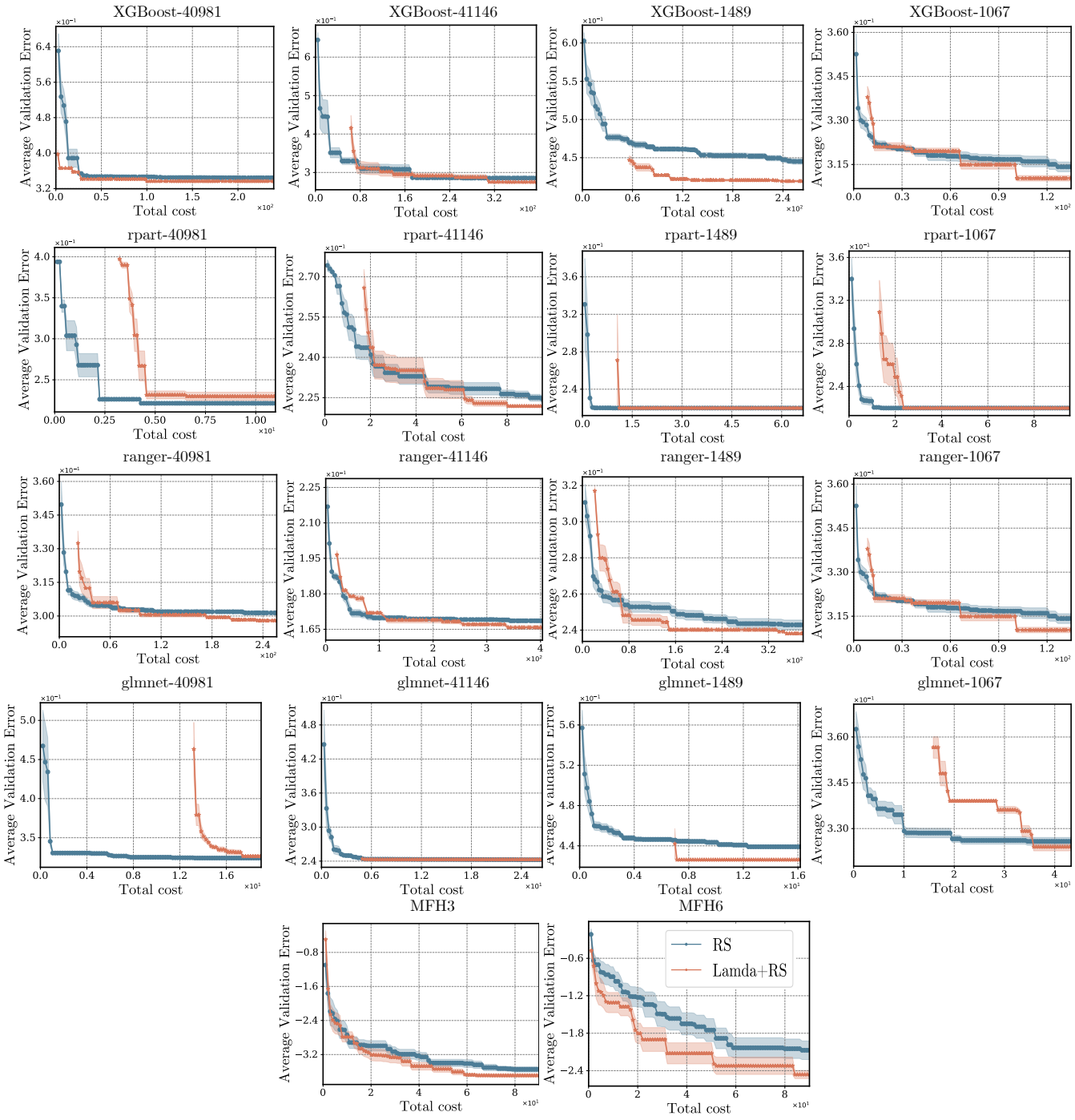

Figure 13: Validation error observed in tuning 18 HPO tasks, using `RS` as the baseline.

.

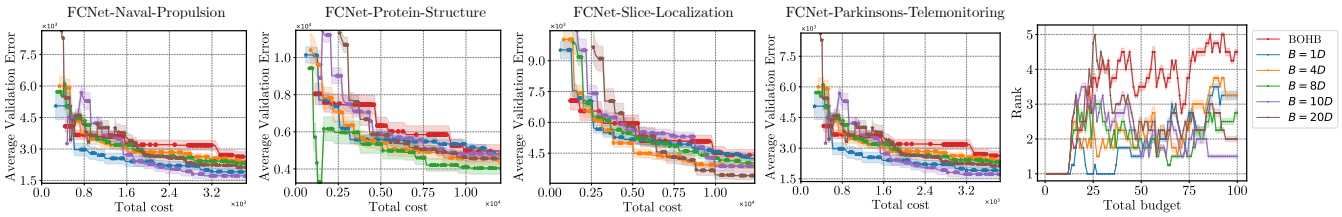

Figure 14: Validation errors of `Lamda+BOHB` under different parameters B used at the first phase.

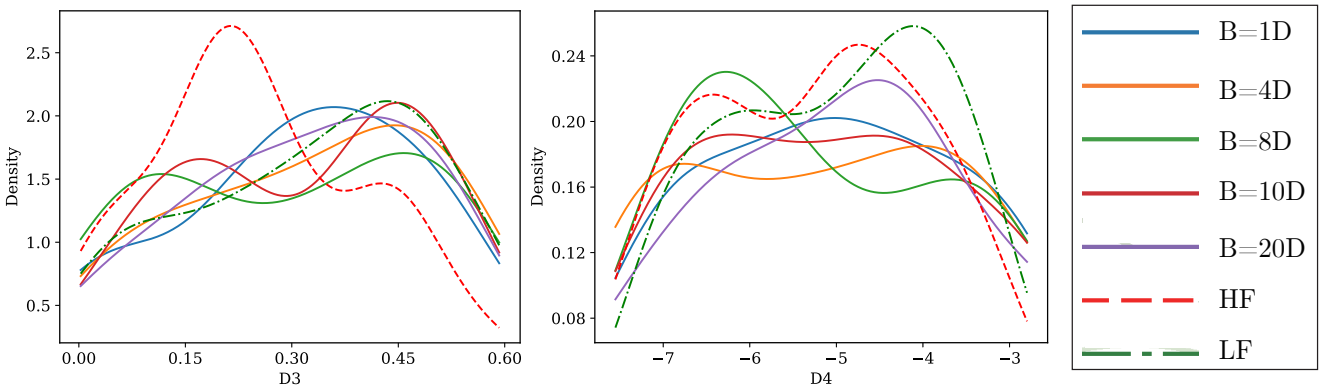

Figure 15: Prior distributions learned in the first phase of `Lamda` using different budget parameters B on the `FCNet-Naval-Propulsion` task.

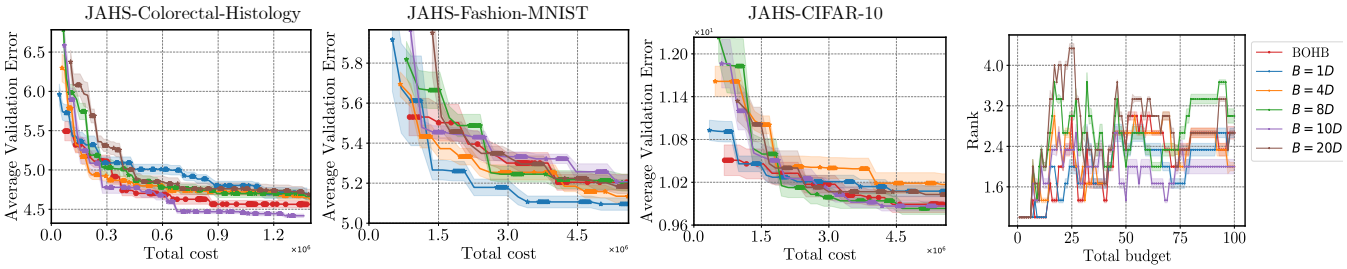

Figure 16: Validation errors of `Lamda+BOHB` under different parameters B used at the first phase on `JAHS-CIFAR-10`.

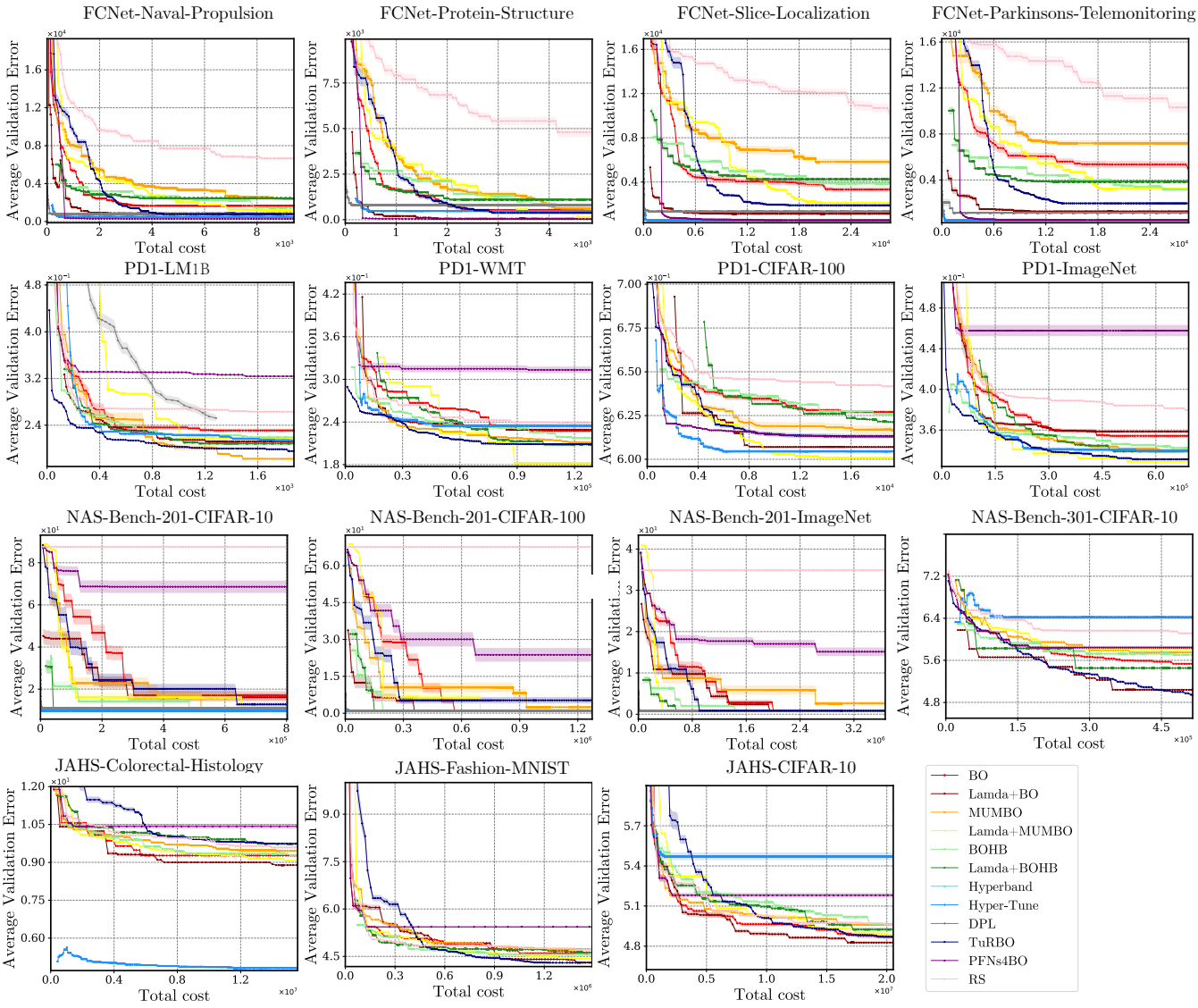

Figure 17: Validation error observed in tuning 15 HPO tasks, comparing peer algorithms.

.

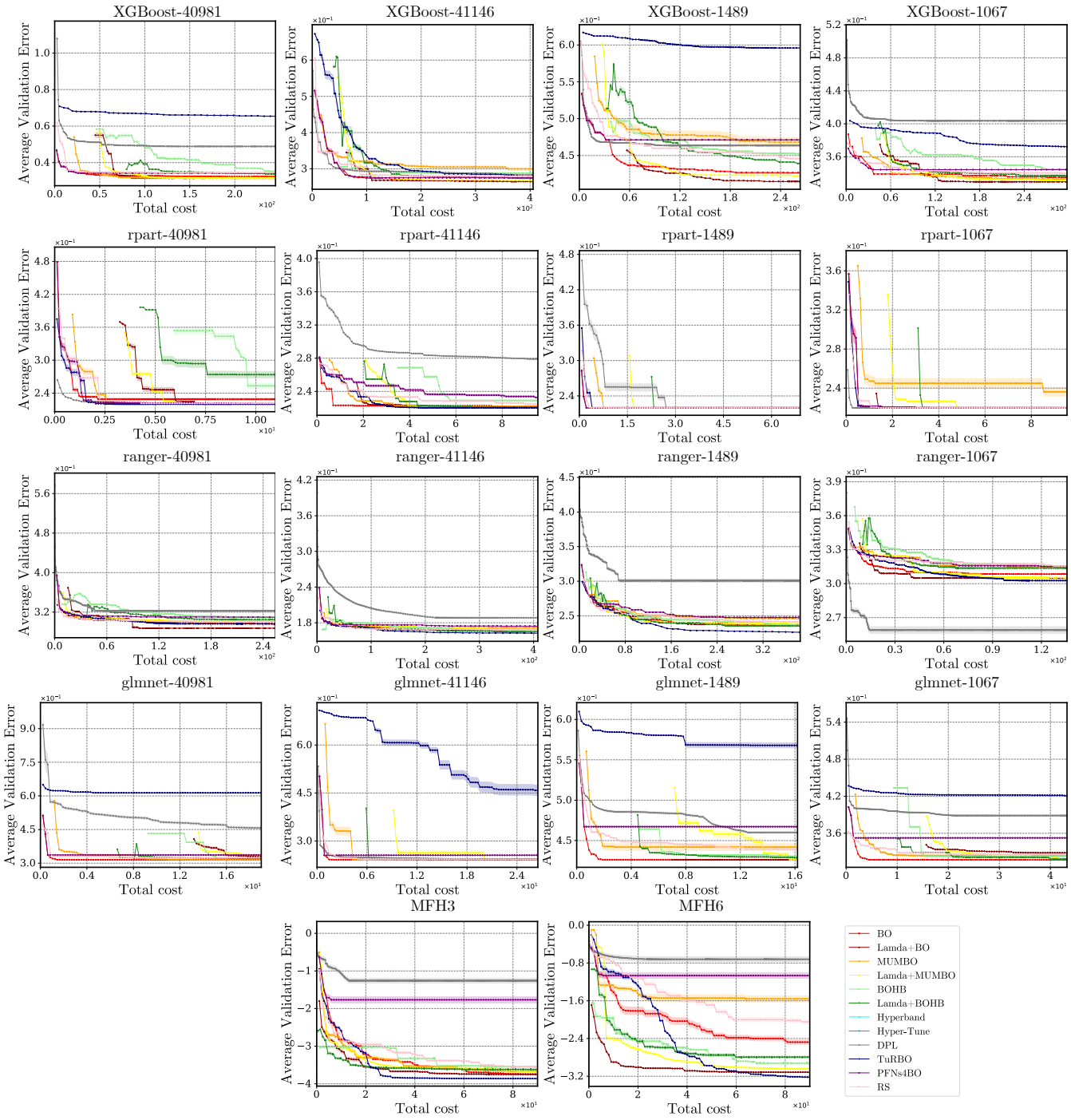

Figure 18: Validation error observed in tuning 18 HPO tasks, comparing peer algorithms.

Table 4: Comparing peer algorithms' final validation errors of the current incumbent at 100 HF evaluations horizons. Runs are averaged over 31 seeds. (The bolded part indicates: "under the Wilcoxon rank-sum test, methods incorporating priors significantly outperform their baselines.")

Tasks	BOHB	Lamda+BOHB	MUMBO	Lamda+MUMBO	BO	Lamda+BO	RS	Lamda+RS	PriorBand	Lamda+PriorBand
MFH3	-3.655e+00(1.339e-01)	**-3.716e+00(1.056e-01)**	-3.860e+00(3.569e-01)	-3.846e+00(2.174e-01)	-3.822e+00(2.172e-01)	-3.843e+00(5.508e-02)	**-3.572e+00(2.529e-01)**	**-3.718e+00(1.323e-01)**	-3.857e+00(3.553e-03)	-3.852e+00(2.401e-03)
MFH6	-2.864e+00(1.364e-01)	**-2.935e+00(1.427e-01)**	-1.872e+00(0)	**-3.178e+00(2.752e-01)**	-2.138e+00(9.286e-01)	**-3.112e+00(8.129e-02)**	-2.260e+00(6.577e-01)	**-2.396e+00(2.627e-01)**	-3.187e+00(1.506e-01)	-3.167e+00(5.198e-02)
XGBoost-40981	3.456e-01(9.817e-03)	3.337e-01(9.490e-03)	3.147e-01(2.235e-07)	3.147e-01(2.235e-07)	3.337e-01(4.953e-03)	**3.147e-01(1.028e-05)**	3.442e-01(1.143e-03)	**3.367e-01(1.528e-03)**	NA	NA
XGBoost-41146	2.883e-01(1.238e-03)	**2.783e-01(6.558e-03)**	2.634e-01(1.694e-03)	2.635e-01(1.063e-03)	2.660e-01(1.620e-03)	**2.646e-01(1.823e-03)**	2.824e-01(8.990e-03)	**2.768e-01(4.922e-04)**	NA	NA
XGBoost-1489	4.477e-01(8.205e-03)	**4.330e-01(2.073e-02)**	4.616e-01(5.269e-02)	**4.116e-01(1.450e-02)**	4.164e-01(1.462e-02)	4.116e-01(5.029e-03)	4.415e-01(1.642e-02)	**4.187e-01(5.514e-03)**	NA	NA
XGBoost-1067	3.453e-01(5.331e-03)	**3.377e-01(3.775e-03)**	3.345e-01(5.656e-03)	**3.301e-01(5.028e-03)**	3.357e-01(7.962e-04)	**3.298e-01(1.182e-04)**	3.399e-01(0)	**3.345e-01(0)**	NA	NA
rpart-40981	2.200e-01(6.758e-02)	**2.199e-01(8.859e-06)**	2.199e-01(7.525e-06)	2.199e-01(1.937e-06)	2.199e-01(3.759e-06)	2.200e-01(2.866e-04)	2.213e-01(8.945e-04)	**2.199e-01(1.214e-05)**	NA	NA
rpart-41146	2.222e-01(1.529e-03)	**2.210e-01(9.219e-04)**	2.210e-01(4.312e-04)	2.205e-01(3.331e-04)	2.210e-01(1.799e-03)	**2.204e-01(1.356e-04)**	2.233e-01(3.626e-03)	**2.216e-01(1.541e-03)**	NA	NA
rpart-1089	2.198e-01(0)	2.198e-01(0)	2.198e-01(0)	2.198e-01(0)	2.198e-01(0)	2.198e-01(0)	2.198e-01(0)	2.198e-01(0)	NA	NA
rpart-1067	2.198e-01(0)	2.198e-01(0)	2.388e-01(0)	2.198e-01(0)	2.198e-01(0)	2.198e-01(0)	2.198e-01(0)	2.198e-01(0)	NA	NA
ranger-40981	3.087e-01(4.186e-03)	**3.038e-01(6.079e-03)**	2.993e-01(2.547e-03)	3.022e-01(3.181e-03)	3.024e-01(7.710e-03)	**2.835e-01(1.074e-02)**	3.019e-01(1.860e-03)	**2.967e-01(2.828e-03)**	NA	NA
ranger-41146	1.645e-01(6.803e-03)	1.645e-01(2.341e-03)	1.683e-01(2.645e-03)	**1.679e-01(4.282e-03)**	1.670e-01(3.795e-03)	1.673e-01(1.883e-03)	1.685e-01(1.981e-03)	**1.641e-01(2.814e-03)**	NA	NA
ranger-1489	2.328e-01(5.845e-03)	2.358e-01(2.925e-03)	2.440e-01(1.350e-03)	2.372e-01(2.984e-03)	2.336e-01(2.404e-03)	**2.316e-01(1.231e-03)**	2.464e-01(4.325e-03)	**2.381e-01(1.180e-03)**	NA	NA
ranger-1067	3.182e-01(6.926e-03)	**3.136e-01(6.902e-03)**	3.138e-01(5.727e-03)	**3.055e-01(5.693e-04)**	3.142e-01(5.990e-03)	**3.061e-01(5.558e-03)**	3.180e-01(2.325e-03)	**3.089e-01(3.449e-03)**	NA	NA
glmnet-40981	3.200e-01(2.952e-03)	**3.161e-01(4.786e-04)**	3.149e-01(0)	3.149e-01(0)	3.149e-01(0)	3.269e-01(6.331e-03)	3.241e-01(4.395e-04)	**3.247e-01(1.148e-03)**	NA	NA
glmnet-41146	2.415e-01(3.053e-04)	2.413e-01(5.809e-04)	2.411e-01(0)	2.411e-01(0)	2.411e-01(0)	2.430e-01(3.797e-04)	2.425e-01(4.839e-04)	**2.429e-01(5.945e-04)**	NA	NA
glmnet-1489	4.290e-01(2.869e-03)	**4.274e-01(5.247e-04)**	4.392e-01(0)	**4.262e-01(0)**	4.262e-01(0)	4.262e-01(0)	4.392e-01(6.759e-03)	**4.262e-01(0)**	NA	NA
glmnet-1067	3.214e-01(4.097e-04)	**3.178e-01(7.455e-05)**	3.171e-01(0)	3.171e-01(0)	**3.171e-01(0)**	3.282e-01(6.786e-03)	3.232e-01(5.589e-03)	**3.210e-01(4.693e-03)**	NA	NA
FCNet-Naval-Propulsion	2.537e+03(1.014e+03)	**2.411e+03(6.557e+02)**	2.458e+03(7.802e+02)	**1.002e+03(5.606e+02)**	2.627e+02(2.871e+02)	**2.121e+02(2.884e+02)**	6.766e+03(6.833e+02)	**3.370e+03(1.043e+03)**	4.461e+03(8.544e+02)	**3.956e+03(4.427e+02)**
FCNet-Protein-Structure	5.030e+03(2.183e+03)	**4.334e+03(2.098e+03)**	6.529e+03(4.761e+03)	**2.015e+03(6.318e+02)**	2.764e+03(3.452e+03)	**1.156e+03(1.424e+02)**	1.367e+04(6.222e+03)	**1.097e+04(5.827e+03)**	6.273e+03(4.760e+03)	**3.501e+03(1.508e+02)**
FCNet-Slice-Localization	4.418e+03(5.756e+02)	**4.006e+03(3.853e+02)**	8.197e+03(4.281e+03)	**3.475e+03(2.819e+03)**	4.825e+03(5.299e+03)	**1.378e+03(1.161e+02)**	1.368e+04(1.960e+03)	**5.048e+03(3.287e+03)**	5.275e+03(1.709e+03)	**4.269e+03(5.377e+03)**
FCNet-Parkinsons-Telemonitoring	1.580e+05(4.104e+02)	**1.139e+03(7.311e+02)**	1.364e+02(2.477e+02)	**1.197e+01(9.573e+00)**	4.236e+02(7.323e+02)	**2.623e-05(6.821e+01)**	**5.333e+03(5.352e+03)**	4.275e+03(3.180e+03)	3.997e+02(8.256e+02)	4.041e+02(1.419e+03)
NAS-Bench-301-CIFAR-10	5.760e+00(2.620e-01)	**5.454e+00(1.091e-01)**	5.761e+00(1.368e-01)	**5.787e+00(3.964e-01)**	5.492e+00(2.995e-01)	5.629e+00(5.881e-01)	6.201e+00(4.134e-01)	**5.538e+00(4.347e-01)**	NA	NA
NAS-Bench-201-CIFAR-10	9.712e+00(8.138e-10)	**9.712e+00(2.543e-10)**	9.792e+00(6.103e-10)	**9.712e+00(1.272e-10)**	9.712e+00(1.424e-09)	9.712e+00(9.664e-10)	**8.760e+01(0)**	9.712e+00(1.017e-09)	NA	NA
NAS-Bench-201-CIFAR-100	1.000e+00(0)	**1.000e+00(0)**	1.000e+00(0)	1.000e+00(0)	1.000e+00(0)	1.000e+00(0)	6.764e+01(0)	**1.000e+00(0)**	NA	NA
NAS-Bench-201-ImageNet	8.333e-01(3.709e-10)	**8.333e-01(2.914e-10)**	8.333e-01(1.854e-10)	**8.133e-01(0)**	8.333e-01(2.384e-10)	8.333e-01(3.444e-10)	3.487e+01(0)	**8.333e-01(4.239e-10)**	NA	NA
JAHS-CIFAR-10	9.252e+00(1.828e-01)	9.495e+00(5.682e-01)	9.340e+00(5.540e-01)	**9.002e+00(1.398e-01)**	9.268e+00(1.037e-01)	**9.118e+00(2.973e-01)**	9.536e+00(3.553e-01)	**9.201e+00(1.504e-01)**	NA	NA
JAHS-Colorectal-Histology	4.467e+00(3.072e-01)	4.542e+00(1.801e-01)	4.596e+00(3.287e-01)	**4.451e+00(2.517e-01)**	4.576e+00(2.848e-01)	**4.288e+00(5.159e-02)**	4.940e+00(3.295e-01)	**4.209e+00(5.453e-02)**	NA	NA
JAHS-Fashion-MNIST	4.966e+00(9.998e-02)	4.960e+00(2.068e-01)	5.006e+00(1.478e-01)	**4.904e+00(2.414e-01)**	4.850e+00(1.387e-01)	4.864e+00(8.280e-02)	4.980e+00(0)	**4.788e+00(0)**	NA	NA
PD1-LM1B	6.255e-01(6.187e-03)	**6.209e-01(4.692e-03)**	6.092e-01(3.626e-02)	**6.042e-01(1.300e-02)**	6.277e-01(1.818e-02)	**6.063e-01(2.018e-04)**	6.459e-01(1.125e-02)	**6.315e-01(9.575e-03)**	9.905e+01(0)	9.905e+01(3.879e-03)
PD1-WMT	3.438e-01(1.379e-02)	**3.394e-01(1.243e-02)**	3.357e-01(4.336e-03)	**3.326e-01(8.931e-03)**	3.578e-01(1.671e-02)	3.579e-01(2.453e-02)	3.762e-01(0)	**3.624e-01(0)**	9.907e+01(2.832e-03)	9.906e+01(0)
PD1-CIFAR-100	2.141e-01(3.115e-02)	**2.097e-01(1.186e-02)**	1.767e-01(2.363e-03)	**1.743e-01(4.328e-04)**	2.345e-01(4.867e-02)	**1.946e-01(4.658e-02)**	2.722e-01(5.905e-03)	**2.269e-01(8.842e-03)**	NA	NA
PD1-ImageNet	2.282e-01(3.206e-02)	**2.050e-01(1.184e-02)**	1.942e-01(0)	**1.824e-01(0)**	2.263e-01(1.753e-03)	**2.145e-01(1.641e-02)**	2.360e-01(0)	**1.986e-01(0)**	NA	NA

F.3.2 Results on raw problems In this part, we evaluate BOHB and Lamda+BOHB on three raw HPO tasks. Figure 19 shows the performance curves on three vision problems. We observe that the performance of Lamda+BOHB is worse than BOHB at the initial iteration. However, it quickly outperforms BOHB after some resources. The main reason is that the promising regions at the high and low fidelity have great overlapping as shown Figure 20.

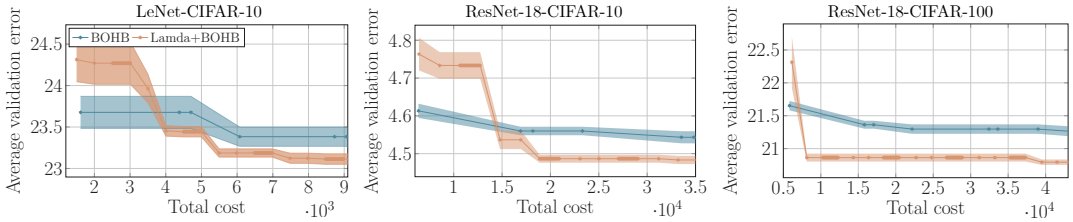

Figure 19: Validation error observed in tuning raw problems.

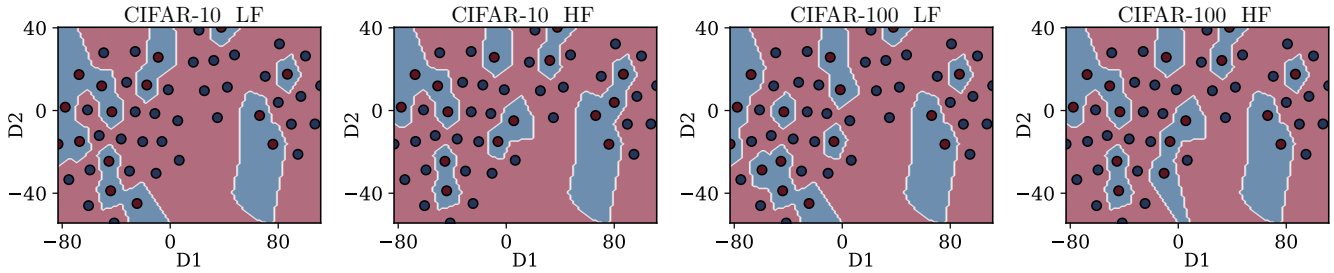

Figure 20: Visualizing the top 30% of solutions (represented by blue regions) in both high and low fidelity while optimizing the hyperparameters of ResNet-18 on CIFAR-10 and CIFAR-100 datasets.

F.3.3 Peer comparison on hyperparameter optimization for fine-tuning pretrained image classification models In this section, we additionally adopt the hyperparameter optimization from (Pineda-Arango et al. 2024) for fine-tuning pretrained image classification models on different datasets. A total of 20 problems are used to evaluate the performance of the algorithms, and the results are presented in Table 5. It can be observed that Lamda+BOHB achieves the best performance across all problems. The overall ranking of the algorithms during the optimization process is illustrated in the Figure 21, showing that Lamda+BOHB consistently ranks first after consuming a portion of the resources. The convergence curves in Figure 22 further highlight its superiority in the second phase, where it quickly identifies high-quality solutions and accelerates the optimization process. The experimental results further demonstrate that Lamda consistently enhances BOHB.

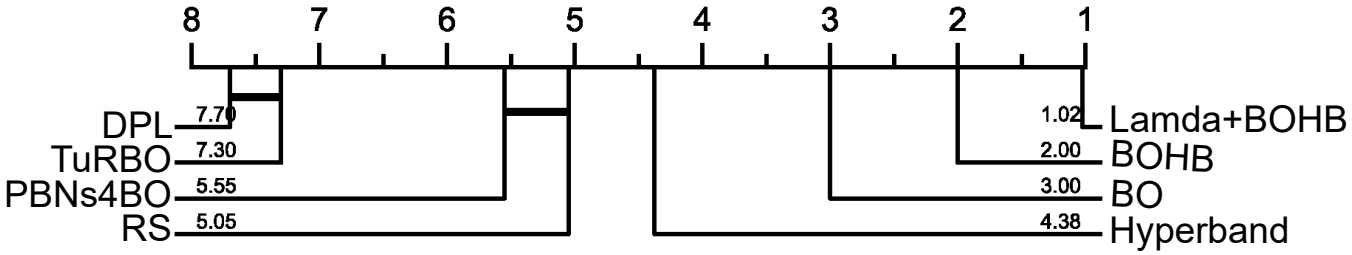

Figure 21: Comparing average relative ranks of peer algorithms across 20 HPO tasks for fine-tuning pretrained image classification models.

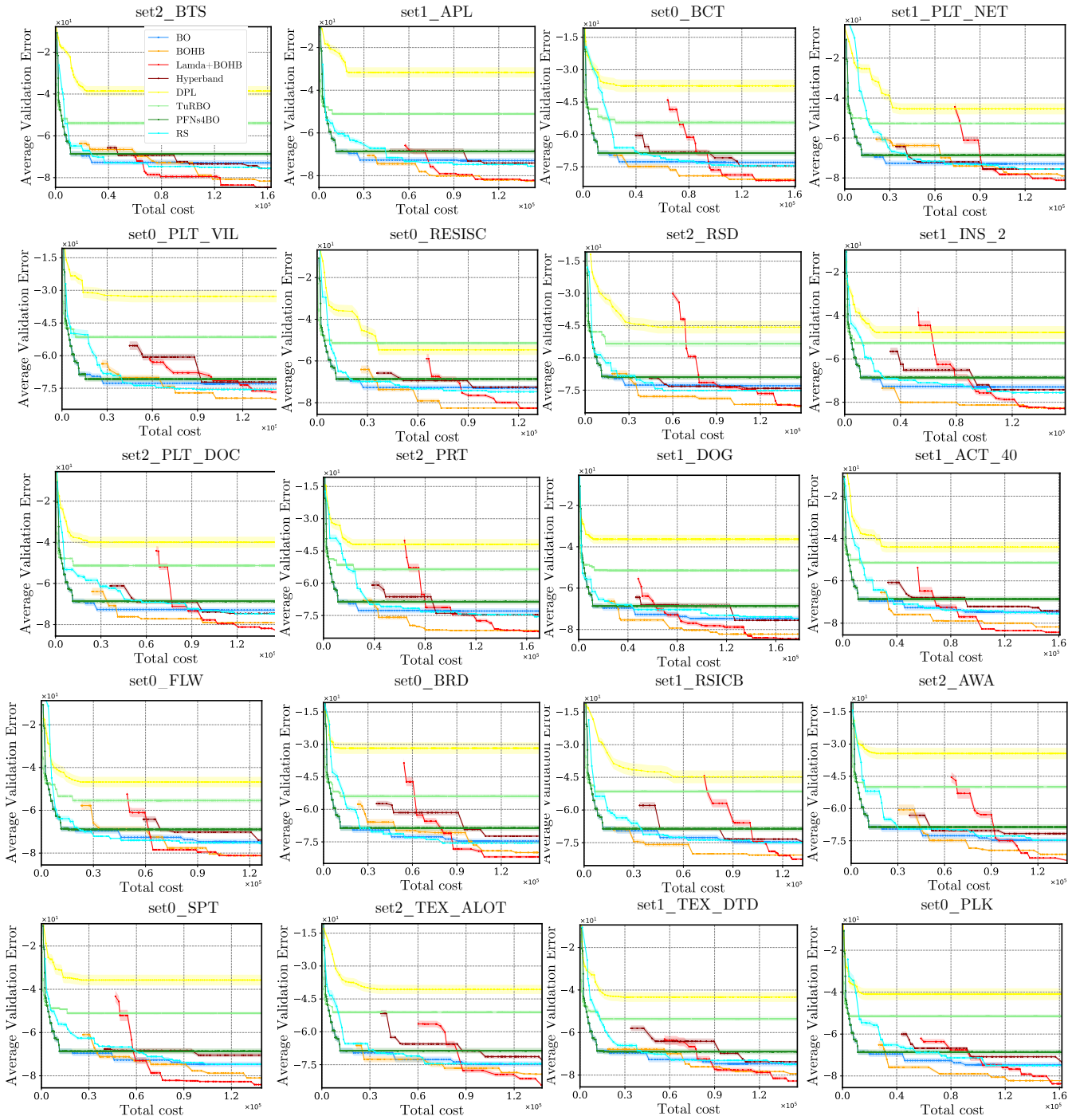

Figure 22: Validation error observed in tuning 20 HPO tasks for fine-tuning pretrained image classification models, comparing peer algorithms.

Table 5: Comparing peer algorithms' final validation errors of the current incumbent at 100 HF evaluations horizons (20 HPO tasks for fine-tuning pretrained image classification models). Runs are averaged over 31 seeds. (The bolded part indicates: "under the Wilcoxon rank-sum test, methods incorporating priors significantly outperform their baselines.")

Data Name	BO	BOHB	Lamda+BOHB	Hyperband	DPL	TuRBO	PFNs4BO	RS
mtlbm_extended_set2_BTS	-7.295e+01(1.282e+01)	-8.256e+01(0.000e+00)	**-8.466e+01(2.709e+00)**	-7.556e+01(0.000e+00)	-3.857e+01(2.394e+01)	-5.392e+01(1.042e+01)	-6.862e+01(1.273e+01)	-7.556e+01(0.000e+00)
mtlbm_extended_set1_APL	-7.295e+01(1.282e+01)	-8.256e+01(0.000e+00)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-3.164e+01(2.416e+01)	-5.105e+01(7.422e+00)	-6.862e+01(1.273e+01)	-7.556e+01(0.000e+00)
mtlbm_extended_set0_BCT	-7.295e+01(1.282e+01)	-8.256e+01(0.000e+00)	**-8.409e+01(3.032e+00)**	-7.556e+01(0.000e+00)	-3.747e+01(2.900e+01)	-5.446e+01(1.274e+01)	-6.862e+01(1.273e+01)	-7.455e+01(3.034e+00)
mtlbm_extended_set1_PLT_NET	-7.295e+01(1.282e+01)	-8.166e+01(2.709e+00)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-4.541e+01(2.757e+01)	-5.270e+01(8.711e+00)	-6.862e+01(1.273e+01)	-7.556e+01(0.000e+00)
mtlbm_extended_set0_PLT_VIL	-7.295e+01(1.282e+01)	-8.256e+01(0.000e+00)	**-8.466e+01(2.709e+00)**	-7.556e+01(0.000e+00)	-3.273e+01(2.761e+01)	-5.142e+01(8.522e+00)	-7.078e+01(1.197e+01)	-7.556e+01(0.000e+00)
mtlbm_extended_set0_RESISC	-7.295e+01(1.282e+01)	-8.256e+01(0.000e+00)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-5.465e+01(2.450e+01)	-5.142e+01(8.522e+00)	-6.862e+01(1.273e+01)	-7.478e+01(2.349e+00)
mtlbm_extended_set2_RSD	-7.295e+01(1.282e+01)	-8.256e+01(0.000e+00)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-4.570e+01(2.934e+01)	-5.342e+01(1.467e+01)	-6.896e+01(1.288e+01)	-7.523e+01(9.984e-01)
mtlbm_extended_set1_INS_2	-7.295e+01(1.282e+01)	-8.256e+01(0.000e+00)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-4.778e+01(2.946e+01)	-5.270e+01(8.980e+00)	-6.862e+01(1.273e+01)	-7.556e+01(0.000e+00)
mtlbm_extended_set2_PLT_DOC	-7.295e+01(1.282e+01)	-8.223e+01(9.984e-01)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-3.996e+01(2.678e+01)	-5.131e+01(8.203e+00)	-6.862e+01(1.273e+01)	-7.478e+01(2.349e+00)
mtlbm_extended_set2_PRT	-7.295e+01(1.282e+01)	-8.256e+01(0.000e+00)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-4.190e+01(2.468e+01)	-5.356e+01(1.065e+01)	-6.862e+01(1.273e+01)	-7.556e+01(0.000e+00)
mtlbm_extended_set1_DOG	-7.467e+01(1.215e+01)	-8.256e+01(0.000e+00)	**-8.489e+01(1.331e+00)**	-7.556e+01(0.000e+00)	-3.633e+01(2.196e+01)	-5.142e+01(8.522e+00)	-6.862e+01(1.273e+01)	-7.462e+01(2.825e+00)
mtlbm_extended_set1_ACT_40	-7.467e+01(1.215e+01)	-8.256e+01(0.000e+00)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-4.404e+01(2.534e+01)	-5.142e+01(8.392e+00)	-6.862e+01(1.273e+01)	-7.556e+01(0.000e+00)
mtlbm_extended_set0_FLW	-7.467e+01(1.215e+01)	-8.256e+01(0.000e+00)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-4.677e+01(2.420e+01)	-5.547e+01(1.106e+01)	-6.896e+01(1.288e+01)	-7.556e+01(0.000e+00)
mtlbm_extended_set0_BRD	-7.467e+01(1.215e+01)	-8.256e+01(0.000e+00)	**-8.500e+01(1.694e+00)**	-7.556e+01(0.000e+00)	-3.176e+01(2.669e+01)	-5.392e+01(1.042e+01)	-6.862e+01(1.273e+01)	-7.556e+01(0.000e+00)
mtlbm_extended_set1_RSICB	-7.467e+01(1.215e+01)	-8.256e+01(0.000e+00)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-4.478e+01(2.931e+01)	-5.146e+01(8.506e+00)	-6.862e+01(1.273e+01)	-7.500e+01(1.694e+00)
mtlbm_extended_set2_AWA	-7.467e+01(1.215e+01)	-8.256e+01(0.000e+00)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-3.442e+01(2.638e+01)	-4.993e+01(9.759e+00)	-6.862e+01(1.273e+01)	-7.466e+01(2.709e+00)
mtlbm_extended_set0_SPT	-7.467e+01(1.215e+01)	-8.178e+01(2.349e+00)	**-8.556e+01(0.000e+00)**	-7.523e+01(9.984e-01)	-3.574e+01(2.475e+01)	-5.105e+01(7.422e+00)	-6.862e+01(1.273e+01)	-7.462e+01(2.825e+00)
mtlbm_extended_set2_TEX_ALOT	-7.467e+01(1.215e+01)	-8.256e+01(0.000e+00)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-4.067e+01(2.077e+01)	-5.105e+01(7.422e+00)	-6.862e+01(1.273e+01)	-7.462e+01(2.825e+00)
mtlbm_extended_set1_TEX_DTD	-7.467e+01(1.215e+01)	-8.223e+01(9.984e-01)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-4.332e+01(2.276e+01)	-5.356e+01(1.065e+01)	-6.896e+01(1.288e+01)	-7.500e+01(1.694e+00)
mtlbm_extended_set0_PLK	-7.467e+01(1.215e+01)	-8.256e+01(0.000e+00)	**-8.556e+01(0.000e+00)**	-7.556e+01(0.000e+00)	-4.097e+01(2.802e+01)	-5.146e+01(8.506e+00)	-6.862e+01(1.273e+01)	-7.500e+01(1.694e+00)

F.4 Sensitivity of `Lamda` to Parameter Settings

We investigate the impact of parameters in `Lamda` within the `BOHB` framework, including different thresholds (γ) for stopping the first phase, the interval for calculating the overlapping coefficient (Δ), the quantile (α) used in the TPE, and the weight (w). Four tasks at the XGBoost benchmark, each involving a 10-dimensional hyperparameter optimization problem, are used for the experiments. The γ values are tested at $0.5, 0.2$, and 0.1. The impact of γ on the algorithm is relatively minor as shown in Figure 23, likely due to the constraints imposed by the maximum budget B. The interval Δ for calculating the overlapping coefficient is tested with values of $3, 5, 15$, and 35. As shown in Figure 24, the algorithm's performance remains consistent across different Δ settings, with only minor variations: low Δ may slightly accelerate premature stabilization detection, while high Δ introduces small delays in verifying stability. For the TPE quantile α, values of $5, 15$, and 25 are tested. As illustrated in Figure 25, variations in α have no significant effect on performance, indicating that it is less critical in the framework.

Regarding the weight w, we examine settings of $1, 0.8, 0.5, 0.3$, and 0.1. The results in Figure 26 indicate that while the optimal w slightly vary across tasks, its overall impact on the algorithm's performance is minimal. In addition, values above 0.1 consistently outperform the original `BOHB`. Further analysis of rankings show that settings with $w > 0.1$ achieve better results compared to $w = 0.1$. Notably, a setting of $w = 1$ shows superior performance during the early stages of the optimization. These results also indicate the efficiency of using the prior. Overall, the algorithm's performance is not sensitive to those parameters.

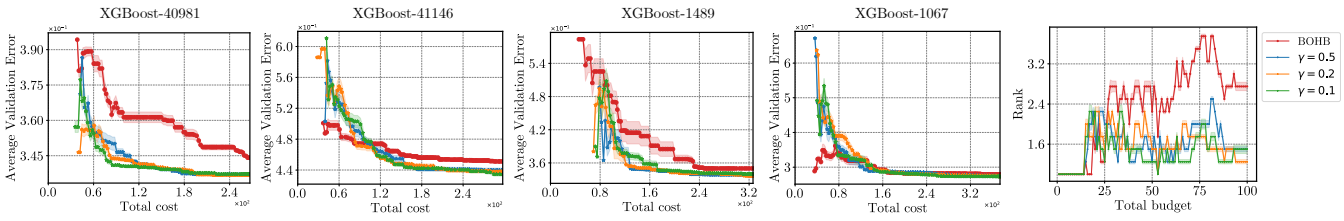

Figure 23: Validation error observed of `Lamda+BOHB` under different parameter γ at the first phase.

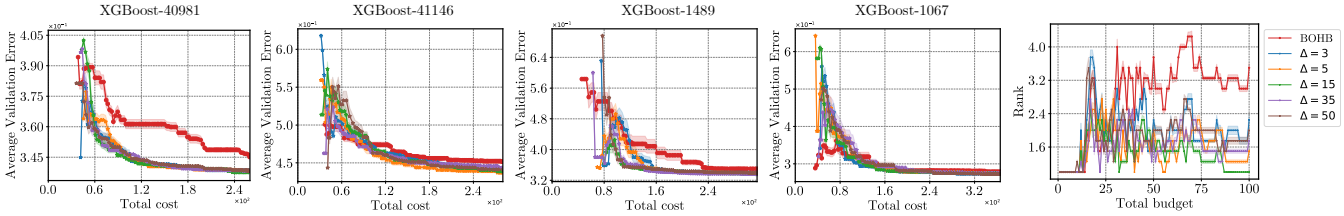

Figure 24: Validation error observed of `Lamda+BOHB` under different parameter Δ at the first phase.

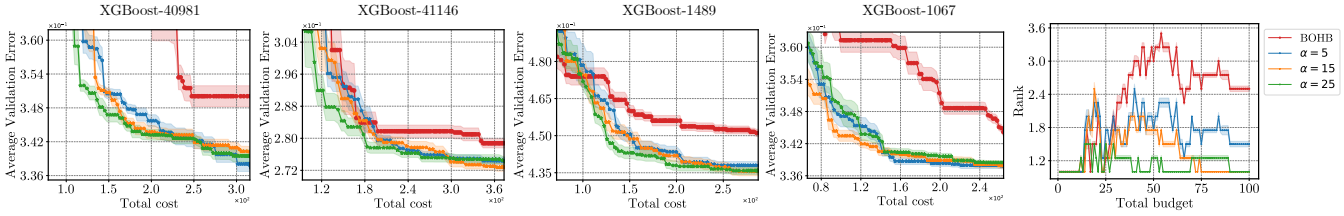

Figure 25: Validation error observed of `Lamda+BOHB` under different parameter α at the first phase.

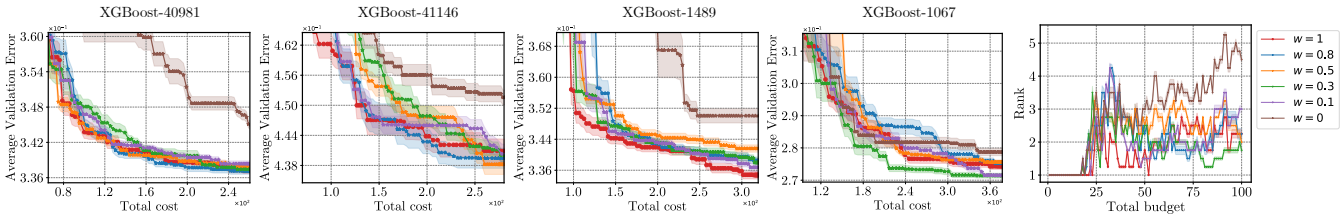

Figure 26: Validation error observed of `Lamda+BOHB` under different parameter w at the first phase.

References

Awad, N. H.; Mallik, N.; and Hutter, F. 2021. DEHB: Evolutionary Hyberband for Scalable, Robust and Efficient Hyperparameter Optimization. In *IJCAI'21: Proc. of the Thirtieth International Joint Conference on Artificial Intelligence*, 2147–2153. ijcai.org.

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.

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*.

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.

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.

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.; Dasarathy, G.; Schneider, J. G.; and Póczos, B. 2017. Multi-fidelity Bayesian Optimisation with Continuous Approximations. In *ICML'17: Proc. of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1799–1808. PMLR.

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, C.; Gupta, S.; Rana, S.; Nguyen, V.; Robles-Kelly, A.; and Venkatesh, S. 2020a. Incorporating Expert Prior Knowledge into Experimental Design via Posterior Sampling. *CoRR*, abs/2002.11256.

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, S.; Xing, W.; Kirby, R. M.; and Zhe, S. 2020b. Multi-Fidelity Bayesian Optimization via Deep Neural Networks. In *NeurIPS'20: Proc. of the Annual Conference on Neural Information Processing Systems 2020*.

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.

Li, Y.; Shen, Y.; Jiang, J.; Gao, J.; Zhang, C.; and Cui, B. 2021. MFES-HB: Efficient Hyperband with Multi-Fidelity Quality Measurements. In *AAAI'21: Proc. of the AAAI Conference on Artificial Intelligence*, 8491–8500. AAAI Press.

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.

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.

Moss, H. B.; Leslie, D. S.; and Rayson, P. 2020. MUMBO: Multi-task Max-Value Bayesian Optimization. In *ECML/PKDD'20: Proc. of the 2020 European Conference on Machine Learning and Knowledge Discovery in Databases*, volume 12459 of *Lecture Notes in Computer Science*, 447–462. Springer.

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.

Poloczek, M.; Wang, J.; and Frazier, P. I. 2017. Multi-Information Source Optimization. In *NeurIPS'17: Advances in Neural Information Processing Systems 30*, 4288–4298.

Ramachandran, A.; Gupta, S.; Rana, S.; Li, C.; and Venkatesh, S. 2020. Incorporating expert prior in Bayesian optimisation via space warping. *Knowl. Based Syst.*, 195: 105663.

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.

Swersky, K.; Snoek, J.; and Adams, R. P. 2013. Multi-Task Bayesian Optimization. In *NeurIPS'13: Advances in Neural Information Processing Systems 26*, 2004–2012.

Takeno, S.; Fukuoka, H.; Tsukada, Y.; Koyama, T.; Shiga, M.; Takeuchi, I.; and Karasuyama, M. 2020. Multi-fidelity Bayesian Optimization with Max-value Entropy Search and its Parallelization. In *ICML'J20: Proc. of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 9334–9345. PMLR.

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.

Wistuba, M.; Schilling, N.; and Schmidt-Thieme, L. 2015. Hyperparameter Search Space Pruning - A New Component for Sequential Model-Based Hyperparameter Optimization. In *ECML PKDD'15: Machine Learning and Knowledge Discovery in Databases - European Conference*, volume 9285 of *Lecture Notes in Computer Science*, 104–119. Springer.