Efficient Hyperparameter Optimization with Adaptive Fidelity Identification

Jiantong Jiang[1], **Zeyi Wen**[2,3], **Atif Mansoor**[1], **Ajmal Mian**[1]
[1]The University of Western Australia, [2]Hong Kong University of Science and Technology (Guangzhou), [3]Hong Kong University of Science and Technology

CVPR JUNE 17-21, 2024 SEATTLE, WA

Background

- **Hyperparameter Optimization (HPO)** aims to automatically tune hyperparameters to achieve state-of-the-art machine learning models. Two important directions of HPO are model-based and multi-fidelity methods.
- **Model-based HPO methods** are mostly based on the Bayesian optimization (BO) framework, which iterates three steps: (i) fully evaluate a hyperparameter configuration to get its final fidelity performance, (i) update the surrogate model based on the performance, and (iii) select the next configuration based on the surrogate model.
- **Multi-fidelity HPO methods** consider using performance at different resource levels, known as the *fidelities*. Traditional methods follow the successive halving (SHA) framework, which evaluates many random configurations at a low fidelity and promotes well-performing configurations to continue with increasing resources.

Motivation

Limitations of current HPO methods:

- **Model-based methods** require a costly full evaluation of each hyperparameter configuration to fit the surrogate model.
- **Multi-fidelity methods** use a simple random configuration search.
- **Model-based multi-fidelity methods** fail to find appropriate fidelities for the configurations, because low fidelity performance cannot always indicate the final fidelity performance.

Key challenge to address:

What is the appropriate fidelity for each configuration to fit the surrogate model? In other words, which fidelity can provide performance observations that reliably indicate the final fidelity performance?

Key ideas of FastBO:

- To address the key challenge, FastBO identifies an *efficient point* for each configuration to be the fidelity to fit the surrogate model, instead of evaluating all the configurations at the same, fixed fidelity.
- FastBO also identifies a *saturation point* for each configuration to be an approximation of the final fidelity.
- The learning curve modeling module enables adaptive derivation of the key points.
- The warm-up stage and post-processing stages are designed to enable judicious early-termination detection and efficient saturation-level evaluation.

Contact Information

Name Jiantong Jiang `https://jjiantong.github.io`
FastBO Code `https://github.com/jjiantong/FastBO`

Methodology

Estimation of Efficient and Saturation Points

For a given learning curve $\mathcal{C}_i(r)$ of hyperparameter configuration $\boldsymbol{\lambda}_i$, where r represents the resource level (also referred to as fidelity), we formally define the efficient point and saturation point as follows.

The efficient point e_i of $\boldsymbol{\lambda}_i$ is defined as: $e_i = \min\{r \mid \mathcal{C}_i(r) - \mathcal{C}_i(2r) < \delta_1\}$, *where δ_1 is a predefined small threshold.*

The saturation point s_i of $\boldsymbol{\lambda}_i$ is defined as: $s_i = \min\{r \mid \forall r' > r, |\mathcal{C}_i(r') - \mathcal{C}_i(r)| < \delta_2\}$, *where δ_2 is a predefined small threshold.*

Remarks:

- The efficient points balance resource usage and performance quality while capturing valuable learning curve trends, serving as the fidelities to fit the surrogate model.
- The saturation points provide high-quality performance while reducing resource wastage, serving as the approximate final fidelities.

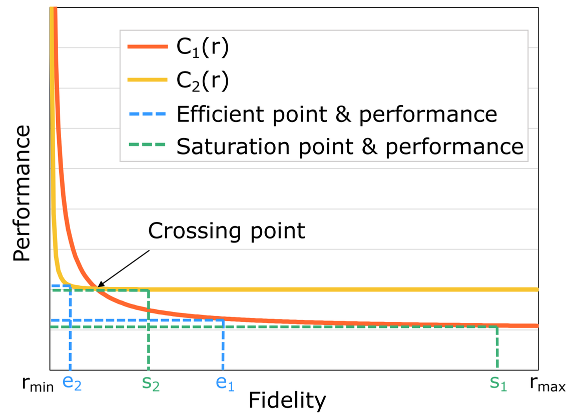


Figure 1: Illustration of the key points.

Learning Curve Modeling

FastBO estimates the learning curve for each configuration $\boldsymbol{\lambda}_i$ based on its observation set $\mathcal{O}_i^w = \{(r, y_i^r)\}_{r=r_{min},\ldots,w}$.

- Constructing a parametric learning curve model by combining three parametric models into a single model:

$$\mathcal{C}(r|\phi) = \sum_{j\in\{1,2,3\}} \omega_j c_j(r|\boldsymbol{\theta}_j),$$

- Estimating parameters ϕ in the parametric model via maximum likelihood estimation.

Table 1: Models used.

Model	Formula
POW3	$y = d + ax^{-\alpha}$
EXP3	$y = d + e^{-ax+b}$
LOG2	$y = d + a\log(x)$

Warm-up Stage

- Preparing the observation set $\mathcal{O}_i^w$ for each configuration.
- Early terminating configurations with consecutive performance deterioration.

Post-processing Stage

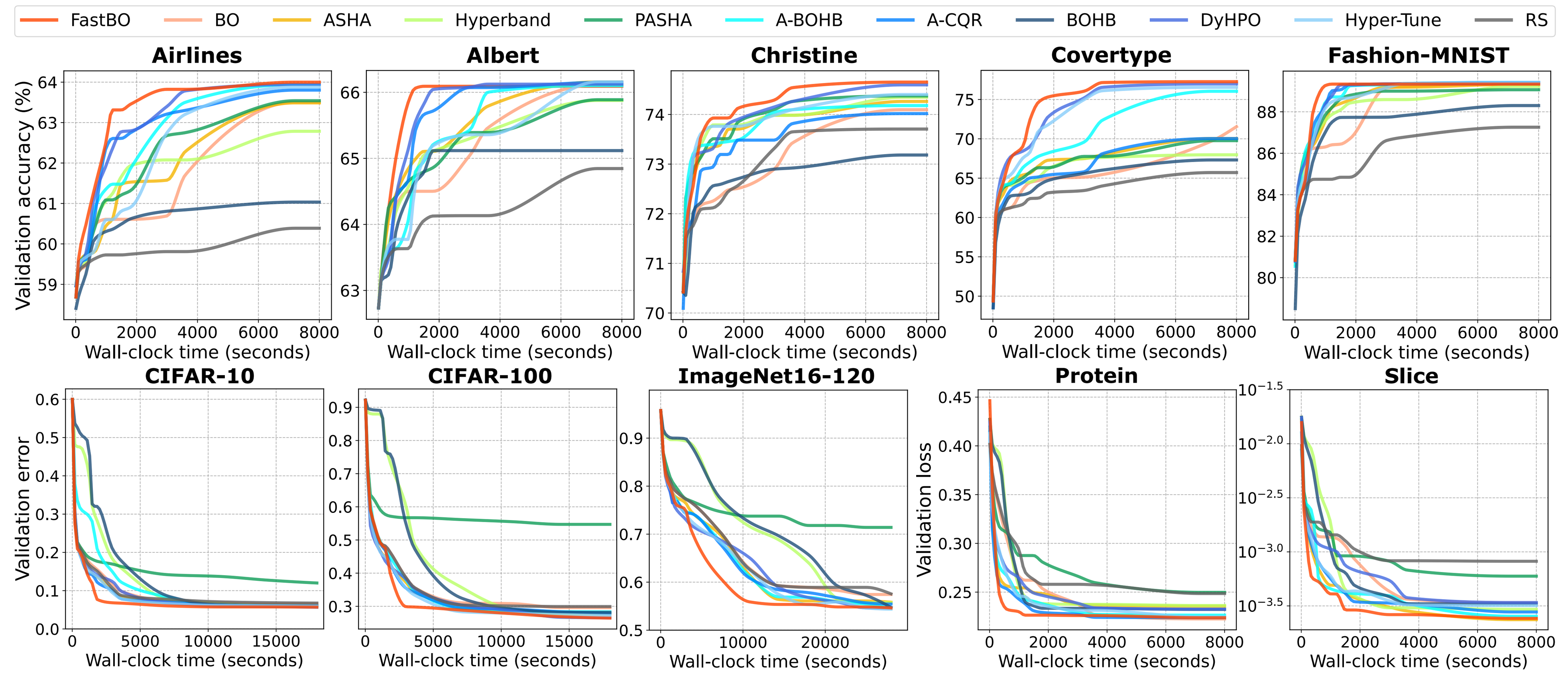
- Finding a small number of well-performing hyperparameter configurations and promoting them for saturation-level evaluations.
- Identifying the best configuration and obtaining its performance.

Acknowledgement

This research was funded by ARC Grant number DP190102443. Professor Ajmal Mian is the recipient of an Australian Research Council Future Fellowship Award (project number FT210100268) funded by the Australian Government.

Experimental Results

- FastBO shows strong **anytime performance**, gaining an *earlier* advantage.



(a) NAS-Bench-201 benchmark (b) FCNet benchmark

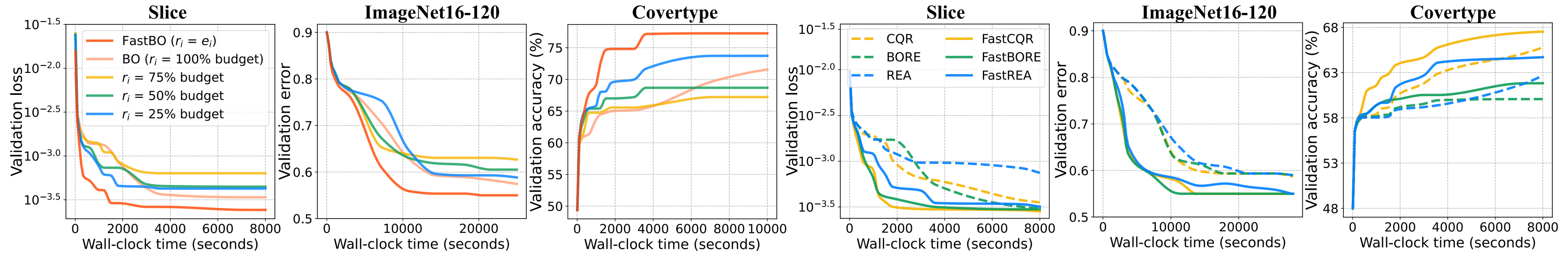
Figure 2: Comparison of anytime performance.

- FastBO shows high **efficiency on configuration identification**.

Table 2: Comparison of relative efficiency on configuration identification.

		FastBO	BO	PASHA	A-BOHB	A-CQR	BOHB	DyHPO	Hyper-Tune
Covertype	Val. error	$\mathbf{22.9}_{\pm 0.2}$	$23.0_{\pm 0.3}$	$25.1_{\pm 2.5}$	$23.5_{\pm 1.1}$	$31.6_{\pm 1.9}$	$32.5_{\pm 0.8}$	$23.0_{\pm 0.3}$	$23.0_{\pm 0.2}$
	WC time (h)	$\mathbf{0.7}_{\pm 0.3}$	$2.9_{\pm 0.7}$	$3.9_{\pm 1.0}$	$2.0_{\pm 1.0}$	$3.9_{\pm 0.2}$	$2.5_{\pm 1.0}$	$1.7_{\pm 0.6}$	$1.8_{\pm 0.7}$
	Rel. efficiency	**1.00**	0.25	0.18	0.37	0.19	0.29	0.41	0.40
ImageNet 16-120	Val. error	$\mathbf{55.3}_{\pm 0.2}$	$57.4_{\pm 1.2}$	$55.7_{\pm 0.3}$	$55.8_{\pm 1.6}$	$55.5_{\pm 0.9}$	$55.5_{\pm 1.1}$	$55.5_{\pm 1.0}$	$55.3_{\pm 2.0}$
	WC time (h)	$\mathbf{2.2}_{\pm 0.7}$	$6.6_{\pm 0.9}$	$2.5_{\pm 1.2}$	$5.9_{\pm 1.1}$	$6.0_{\pm 1.3}$	$3.2_{\pm 0.7}$	$4.3_{\pm 1.0}$	$3.4_{\pm 1.1}$
	Rel. efficiency	**1.00**	0.34	0.90	0.38	0.37	0.68	0.51	0.67
Slice	Val. loss	$\mathbf{26.3}_{\pm 2.6}$	$26.4_{\pm 4.4}$	$26.8_{\pm 9.5}$	$26.3_{\pm 6.3}$	$27.1_{\pm 4.2}$	$26.8_{\pm 5.6}$	$27.4_{\pm 2.3}$	$28.7_{\pm 1.3}$
	WC time (h)	$\mathbf{0.4}_{\pm 0.1}$	$3.1_{\pm 0.7}$	$1.2_{\pm 0.9}$	$2.1_{\pm 0.7}$	$2.5_{\pm 0.7}$	$2.2_{\pm 0.9}$	$2.5_{\pm 0.5}$	$1.8_{\pm 0.6}$
	Rel. efficiency	**1.00**	0.13	0.35	0.20	0.17	0.19	0.17	0.24

- Results shows the **effectiveness of the adaptive fidelity identification strategy**.
- The adaptive strategy of FastBO can extend any single-fidelity method, even the model-free one, to multi-fidelity setting, demonstrating the **generality of FastBO**.



(a) Impact of adaptive fidelity identification (b) Generality of FastBO

Figure 3: Performance comparison regarding adaptive fidelity identification strategy in FastBO.