CVPR
JUNE 17-21, 2024
SEATTLE, WA

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)*

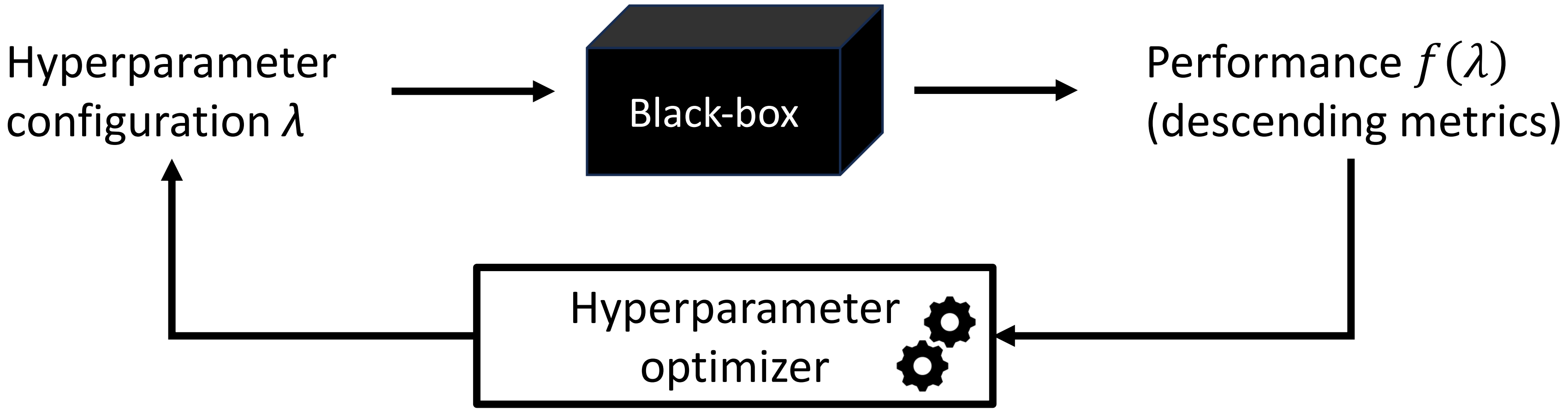
[3] *Hong Kong University of Science and Technology*

THE UNIVERSITY OF WESTERN AUSTRALIA

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Background and Motivation

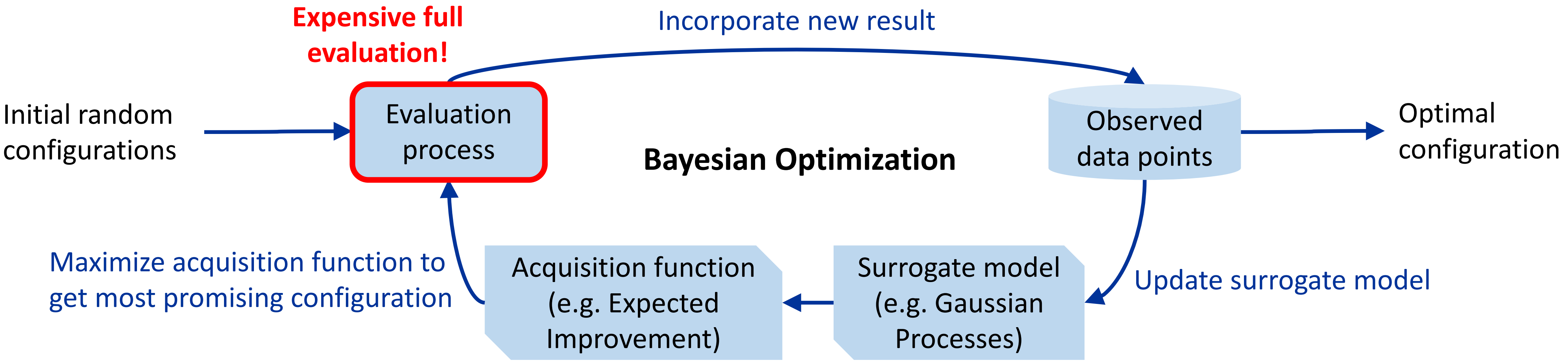
- The performance of machine learning models strongly depends on the choice of **hyperparameters**.
- **Hyperparameter Optimization (HPO)**: automatically find hyperparameters or architectures that yield SOTA performance.



$$\lambda^* = arg \min_{\lambda \in \Lambda} f(\lambda)$$

Background and Motivation

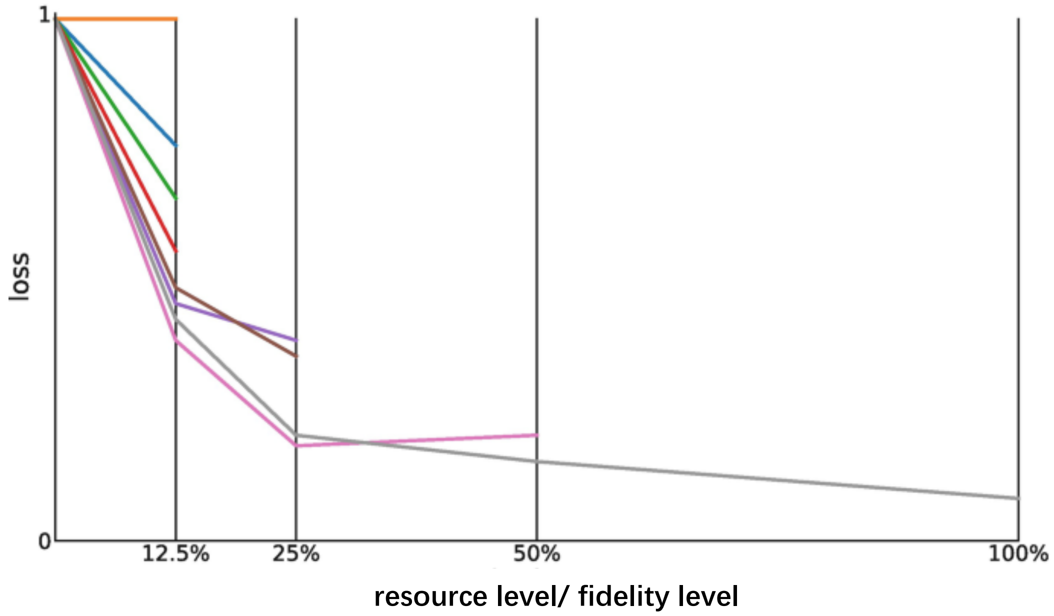
- Crucial HPO directions: model-based methods, multi-fidelity methods.
- **Model-based methods**: follow Bayesian optimization (BO) framework.



- Major limitation: require expensive full evaluation of each configuration to get its final (highest) fidelity performance.

Background and Motivation

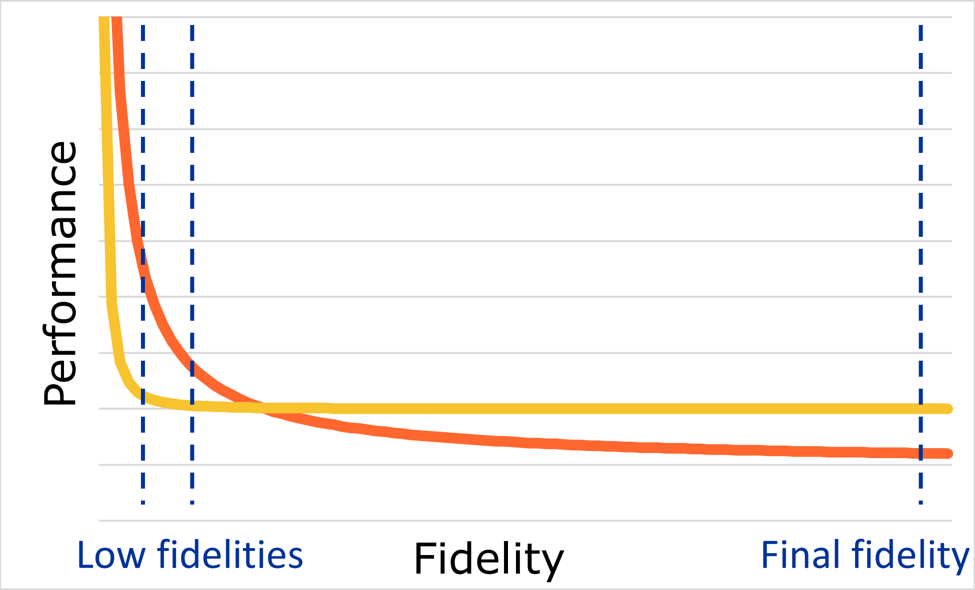
- Crucial HPO directions: model-based methods, multi-fidelity methods.
- **Multi-fidelity methods**: consider performance at different resource levels (fidelities); follow successive halving (SHA) framework.



- Major limitation: use a simple random configuration search.

Background and Motivation

- Combining model-based and multi-fidelity methods: replace the random sampling in SHA with BO.
  - Issue: learning curves of different configurations can intersect.
  - Limitation: early performance got through fixed low fidelities (e.g. 1, 2, 4, 8, …) cannot always indicate high-fidelity performance.

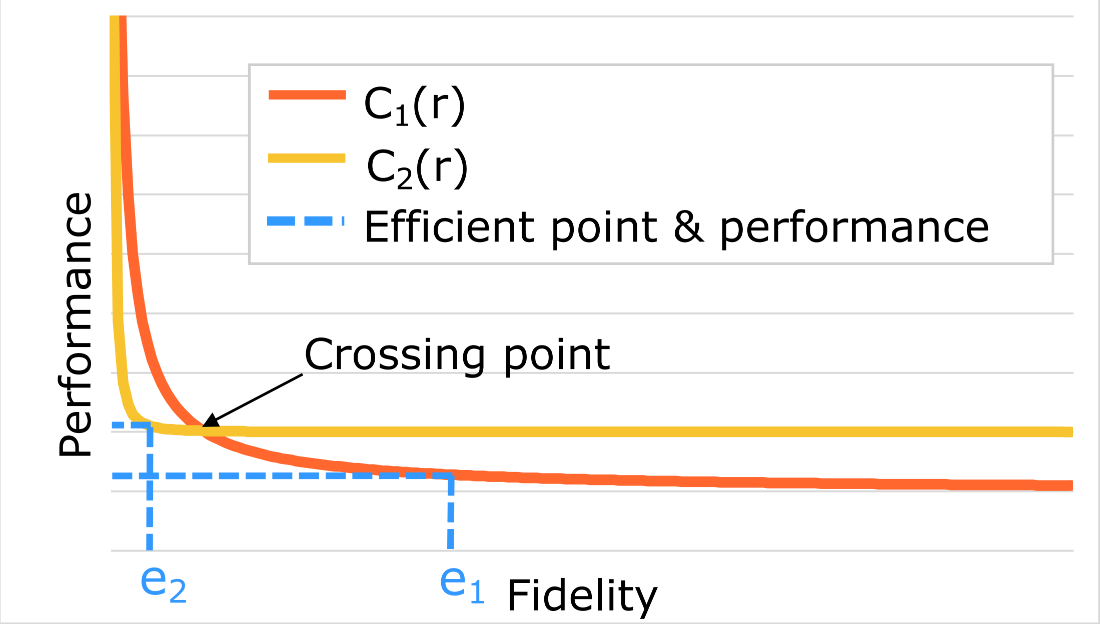


Challenge: *What is the appropriate fidelity for each configuration to fit the surrogate model?*

FastBO Methodology

- **"Efficient point"**: $e_i = \min\{\mathrm{r}\,|C_i(r) - C_i(2r) < \delta_1\}$
  - When resources are doubled (from r to $2r$), the performance improvement falls below a small threshold.
  - Get strong performance while still efficiently using resources.
  - Use as the fidelity to fit the surrogate model.

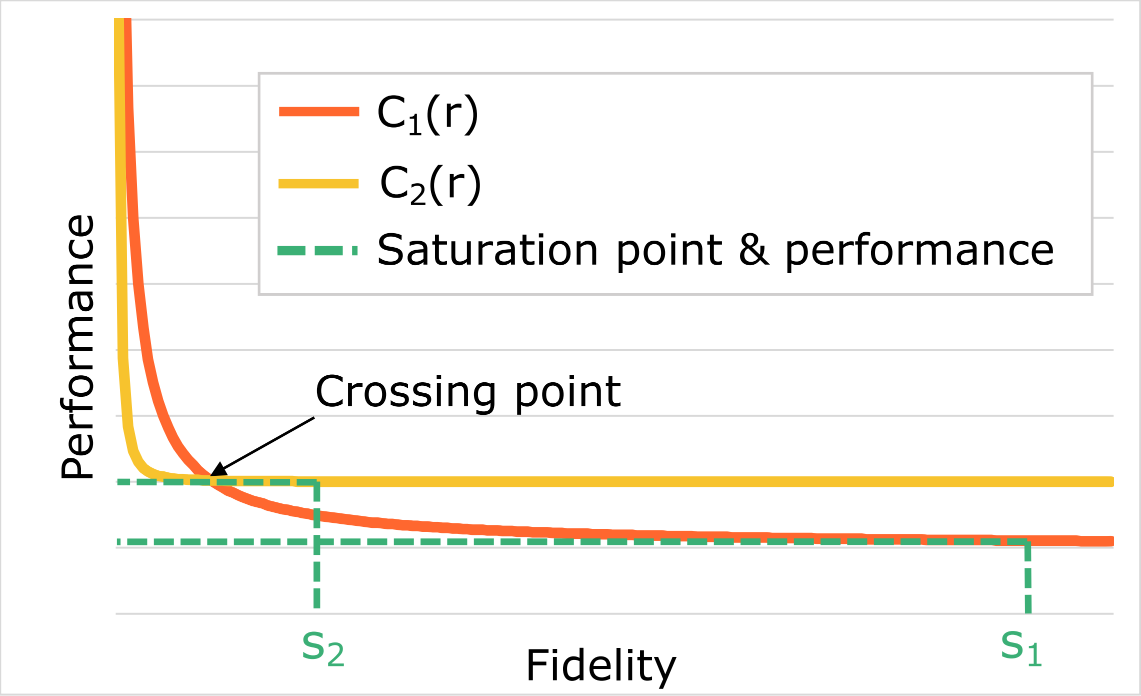
Definition 1 (Efficient point). *For a given learning curve $C_i(r)$ of hyperparameter configuration $\boldsymbol{\lambda}_i$, where r represents the resource level (also referred to as fidelity), the efficient point e_i of $\boldsymbol{\lambda}_i$ is defined as: $e_i = \min\{r \mid C_i(r) - C_i(2r) < \delta_1\}$, where δ_1 is a predefined small threshold.*



FastBO Methodology

- "**Saturation point**": $s_i = \min\{\mathrm{r} \mid \forall r' > r, \ |C_i(r') - C_i(r) < \delta_2\}$
  - Performance does not have notable variations with more resources.
  - Use as a final fidelity approximation.

Definition 2 (Saturation point). *For a given learning curve $C_i(r)$ of configuration $\boldsymbol{\lambda}_i$, where r represents the resource level (also referred to as fidelity), the saturation point s_i of $\boldsymbol{\lambda}_i$ is defined as: $s_i = \min\{r \mid \forall r' > r, |C_i(r') - C_i(r)| < \delta_2\}$, where δ_2 is a predefined small threshold.*



FastBO Methodology

- Process overview

Each configuration → Warm-up → Learning curve modeling → Efficient & saturation points →

- Stop evaluating at efficient point.
- Fit surrogate model.

Warm-up

- Evaluate to the warm-up point.
- Get the early observation set.

Learning curve modeling

- Estimate the learning curve from the observation set.

Efficient & saturation points

- Adaptively extract the points from the learning curve.

…… → Post-processing → Optimal configuration

Post-processing

- Resume some best-performing configurations to saturation point.

Generality of FastBO

- FastBO can be generalized to any single-fidelity methods.
  - Extend single-fidelity methods to multi-fidelity setting: evaluate each config to the efficient point instead of to the final fidelity.
  - Even model-free methods can be improved by our extension strategy.

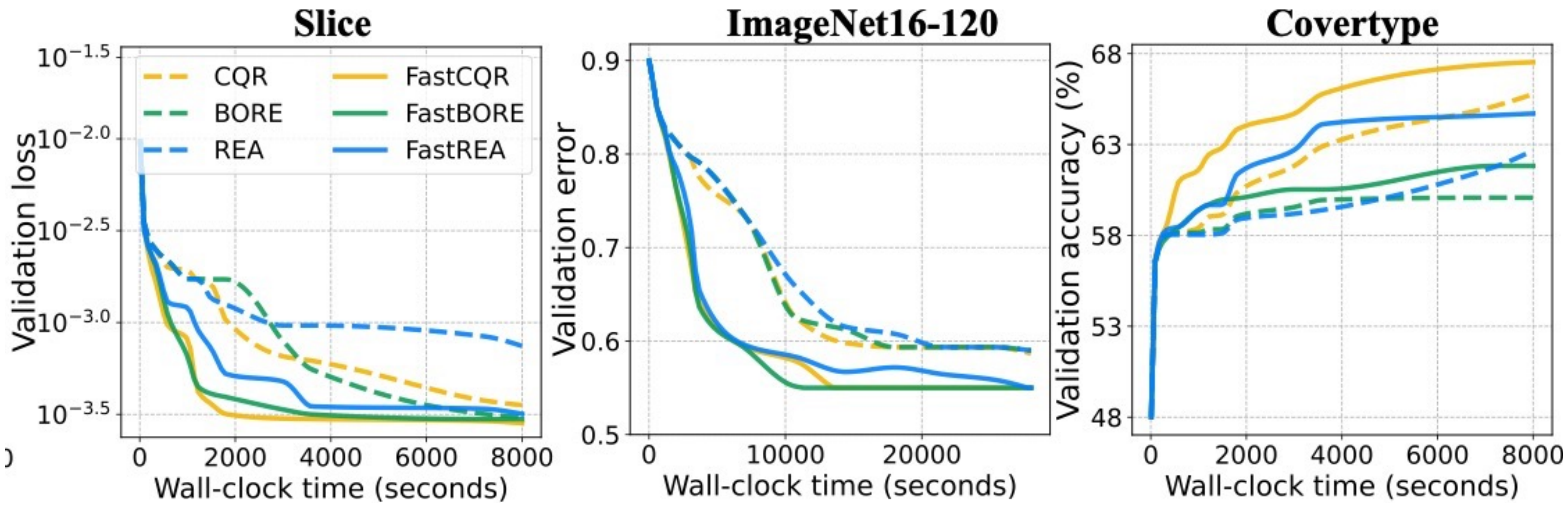


Figure 1. Performance of single-fidelity methods CQR, BORE, REA and their multi-fidelity variants using our extension method.

Experimental Evaluation

- Anytime performance
  - FastBO can handle various performance metrics.
  - FastBO gains an advantage earlier than other methods.

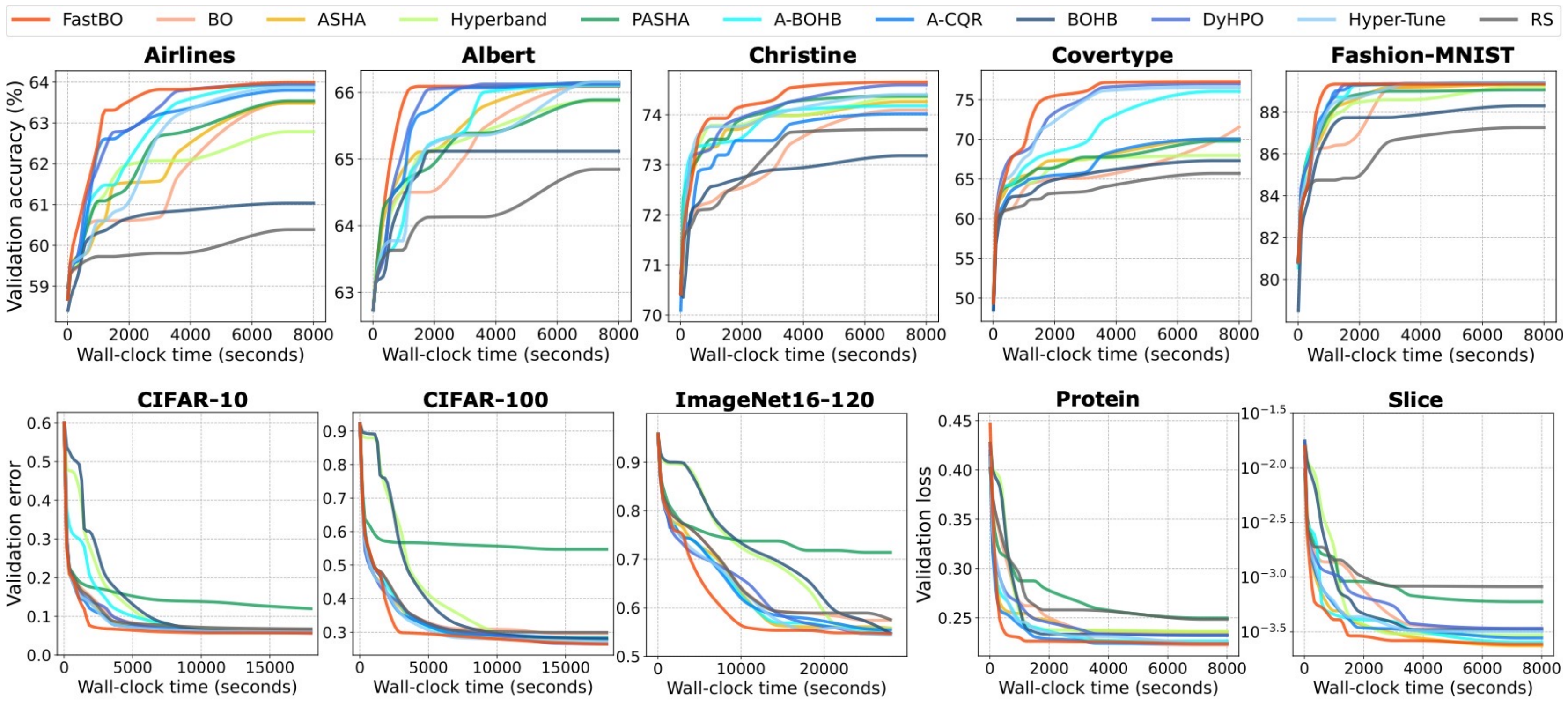


Figure 2. Performance on the LCBench, NAS-Bench-201, and FCNet benchmarks.

Experimental Evaluation

- Efficiency on configuration identification
  - FastBO saves 10% to 87% wall-clock time when achieving up to 9.6% better performance values.

Table 2. Comparison of relative efficiency on configuration identification. FastBO is set as the baseline with a relative efficiency of 1.00. Wall-clock time (abbr. WC time) reports the elapsed time spent for each method on finding configurations with similar performance metrics, i.e., validation error ($\times 10^{-2}$) for Covertype and ImageNet16-120 and validation loss ($\times 10^{-5}$) for Slice.

Dataset	Metric \ Method	**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

Experimental Evaluation

- Evaluating results demonstrate the effectiveness of the adaptive fidelity identification strategy.

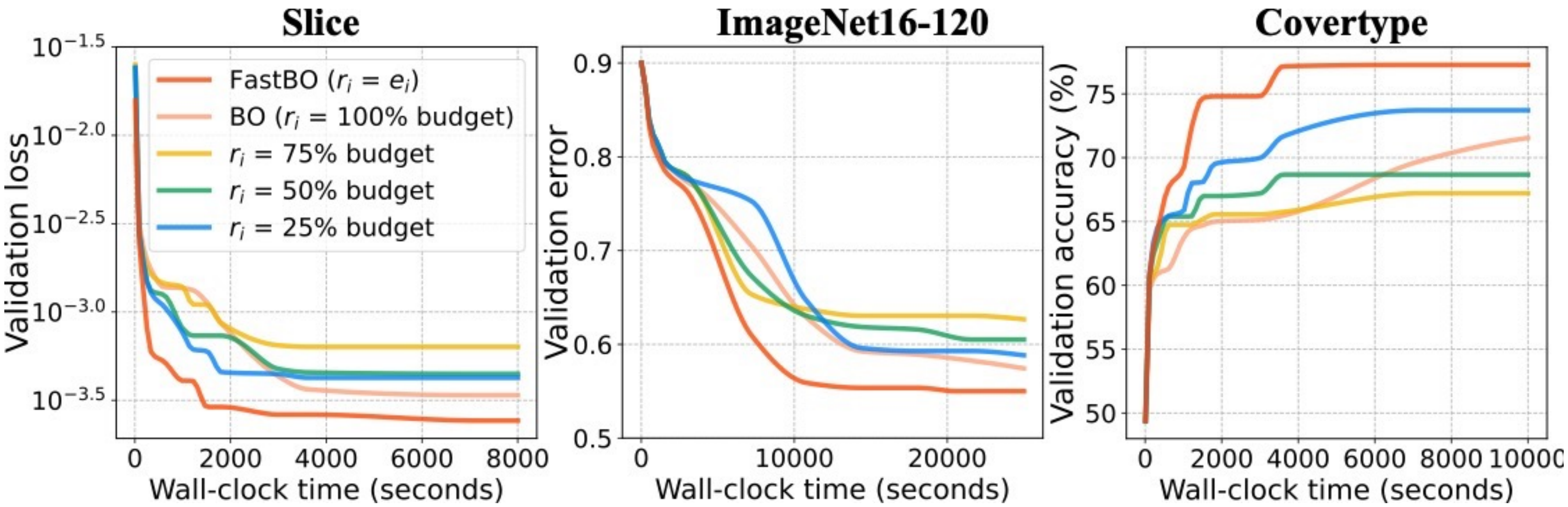


Figure 3. Performance of FastBO that adaptively sets $r_i = e_i$ with the schemes that use fixed r_i for all configurations.

Conclusion

1. We propose a multi-fidelity model-based HPO method that adaptively decides the fidelities for configurations, thanks to the introduced concepts of efficient and saturation points.

2. We develop a learning curve modeling module to adaptively extract the key points, a warm-up stage for early-termination detection, and a post-processing stage for efficient evaluation.

3. Our strategy can be used to extend any single-fidelity methods to multi-fidelity setting.