A Systematic Study on Early Stopping Metrics in HPO and the Implications of Uncertainty

Jiawei Guan
Ren min University of China
guanjw@ruc.edu.cn

Feng Zhang
Renmin University of China
fengzhang@ruc.edu.cn

Jiesong Liu
North Carolina State University
jliu93@ncsu.edu

Xiaoyong Du
Renmin University of China
duyong@ruc.edu.cn

Xipeng Shen
North Carolina State University
xshen5@ncsu.edu

ABSTRACT

The development of hyperparameter optimization (HPO) algorithms is an important topic within both the machine learning and data management domains. While numerous strategies employing early stopping mechanisms have been proposed to bolster HPO efficiency, there remains a notable deficiency in understanding how the selection of early stopping metrics influences the reliability of early stopping decisions and, by extension, the broader HPO outcomes. This paper undertakes a systematic exploration of the impact of metric selection on the effectiveness of early stopping-based HPO. Specifically, we introduce a set of metrics that incorporate uncertainty and highlight their practical significance in enhancing the reliability of early stopping decisions. Our empirical experiments on HPO and NAS benchmarks show that using training loss as an early stopping metric in the early training stages improves HPO outcomes by up to 24.76% compared to the more widely accepted validation loss. Furthermore, integrating uncertainty into the metric yields an additional improvement of up to 4% under budget constraints, translating into meaningful resource savings and scalability benefits in large-scale HPO scenarios. These findings demonstrate the critical role of metric selection while shedding light on the potential implications of integrating uncertainty as a metric. This research provides empirical insights that serve as a compass for the selection and formulation of metrics, thereby contributing to a more profound comprehension of mechanisms underpinning early stopping-based HPO.

PVLDB Reference Format:
Jiawei Guan, Feng Zhang, Jiesong Liu, Xiaoyong Du, and Xipeng Shen. A Systematic Study on Early Stopping Metrics in HPO and the Implications of Uncertainty. PVLDB, 18(6): 1551 - 1564, 2025.
doi:10.14778/3725688.3725689

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/Guan-JW/Metric_Study.

Proceedings of the VLDB Endowment, Vol. 18, No. 6 ISSN 2150-8097.
doi:10.14778/3725688.3725689

1 INTRODUCTION

Recent years have seen a trend towards integrating machine learning (ML) functionalities into data management systems [8, 18, 20–22, 28, 39, 51, 54, 56–58]. Prominent examples include Apache SystemDS [7], Snorkel [41], and HoloClean [42], which facilitate diverse aspects of data handling from integration to predictive analysis. Developing effective ML models to best meet the needs of data management systems is however challenging. Although numerous AutoML systems have emerged in recent years to facilitate the process (e.g., Google Vertex AI [19], Amazon SageMaker Autopilot [10], Microsoft's FLAML [46]), a critical aspect of ML model development, hyperparameter optimization (HPO), is yet to be better understood. The primary challenge lies in the limited understanding of how uncertainties in model predictions affect the reliability of optimization, which is vital for achieving robust HPO outcomes.

The goal of HPO is to determine the best values of some hyperparameters for a model or system. Its importance for tuning data systems has been well recognized by the data system community [30, 32, 33, 43, 47, 55, 56]. The hyperparameters for ML include learning rates, regularization schemes, neural architecture-specific configurations (e.g., types of layers, number of hidden units, etc.), and so on. Their values are critical to model performance.

HPO is time-consuming due to the extensive training of numerous candidate models in a combinatorial hyperparameter space [45, 50]. To address this challenge, various HPO strategies have been developed, employing methods such as Bayesian optimization, genetic algorithms, and rule-based searches for more efficient exploration of the hyperparameter space [6, 53].

Regardless of the used search methods, HPO schemes can be categorized into two main types: *complete evaluation-based HPO* and *early stopping-based (or multi-fidelity) HPO* [32]. Complete evaluation-based HPO involves training the model to completion for each hyperparameter configuration, ensuring thorough testing of each setting but at a high computational cost. In contrast, early stopping-based HPO can fit into an acceptable time budget through early termination of training for underperforming candidates based on specific *early stopping metrics* [2, 35, 37]. This approach can drastically reduce the computational time and resources, and hence becomes dominant in HPO systems [16, 26, 27, 31, 32].

Although previous studies have explored various early stopping-based HPO schemes [5, 13, 36, 48, 52] and highlighted their cost-effectiveness, the understanding of early stopping criteria, especially the performance metrics used for model ranking, remains incomplete. Typically, model training uses separate datasets for

training and validation to compute respective metrics, with validation loss often being the preferred metric for guiding HPO processes; some, especially in data management-centric production environments, use training metrics instead [4, 9, 17].

Early stopping metrics directly determine what models to keep and what models to discard. Despite its pivotal role in HPO, no prior studies have systematically explored it. Existing HPOs have been using either training loss or validation loss as the early stopping metric; which one to use is based on the practitioners' personal preferences, with validation loss being a more frequent choice.

Some fundamental questions on early stopping metric remain open:

(1) How reliable are commonly used performance metrics, specifically *training* and *validation losses*, for HPO? How do they compare to one another?
(2) How to explain the reasons for the different effectiveness of the metrics? More fundamentally, what are the nature of early stopping and the key factors for its effectiveness?
(3) Besides the commonly considered measures, are there any other measures worth considering for early stopping metrics? More specifically, ML models have inherent uncertainty in their predictions. How would such uncertainty impact early stopping? Would it be worthy to incorporate it into early stopping metrics for HPO? And would the combination of multiple metrics help? How to do that?

This paper aims to answer these fundamental questions and advance the principled understanding of early stopping in HPO. We do that through a four-fold exploration. (i) We first conduct an empirical study on nine HPO tasks in three widely used HPO benchmarks (Nas-Bench-201, LCBench, and HPOBench) over nine datasets, and systematically examine the effectiveness of the popular early stopping metrics for HPO. We use the concept of reliability to assess if a specific performance metric shows statistically significant superiority over others. (ii) From the data, we distill a set of insights on the relative effectiveness of the popular early stopping metrics, theoretically analyze the inherent nature of early stopping, and reveal the reasons for the pros and cons of those metrics. (iii) We study the impact of model uncertainty–the variations in model predictions–on early stopping, propose a set of metrics that integrate model uncertainty, and uncover the potential of incorporating model uncertainty into early stopping metrics for HPO. (iv) By identifying distinct stages in the model learning process, we further develop stage-adaptive metrics and augment them with uncertainty measures. This integration aligns early stopping more closely with the dynamics of model training, achieving significant improvement over traditional metrics.

To the best of our knowledge, this work is the first that gives a systematic exploration on early stopping metric for HPO. By addressing some fundamental open questions, it sheds light on how uncertainty and training stages influence metric design and selection, providing valuable insights and empirical guidelines for more effective HPO strategies.

2 BACKGROUND — HPO OVERVIEW

Hyperparameter Optimization (HPO) is an important mechanism in automating the adjustment of hyperparameters, enabling efficient model deployment for practical applications [14]. As illustrated in Figure 1, the HPO process begins with a problem definition phase, which includes identifying hyperparameters, setting tuning objectives, establishing constraints such as computational budgets, and specifying training and validation datasets. Using these definitions, the HPO Tuner initiates an iterative tuning process.

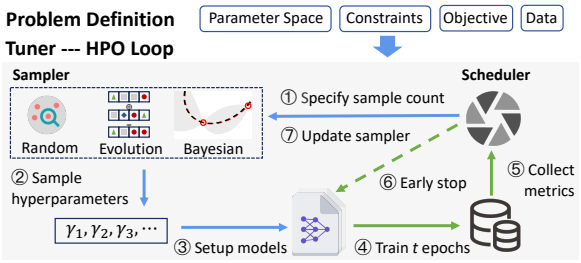


Figure 1: The General HPO Workflow

The HPO Tuner comprises a *scheduler* and a *sampler*. First, the scheduler ① determines the number of samples based on available budgets, and the sampler ② selects hyperparameters accordingly. Next, the Tuner ③ sets up candidate models with these hyperparameters and ④ trains them for a defined number of epochs as per the scheduler's instructions. The scheduler then ⑤ collects performance metrics and ranks the models. If early stopping is employed, the scheduler ⑥ halts underperforming models and directs the rest to continue training. At the end of a tuning cycle, the scheduler provides training statistics to the sampler, which then ⑦ updates its internal parameters to suggest configurations for the next round.

Different HPO strategies adopt varying scheduling policies. Strategies that incorporate early stopping integrate an internal cycle (steps ④, ⑤, and ⑥) to halt unpromising models early in each tuning round. The interaction between early stopping and sampling methods can impact the efficacy of these strategies. General-purpose samplers, such as random sampling, are fully compatible with early stopping, as seen in our Hyperband experiments in the following sections. In contrast, experience-based samplers such as Bayesian optimization may suffer disruptions from premature stops, potentially compromising their predictive accuracy and convergence guarantees. However, advanced approaches such as BOHB [15] reconcile sampling with early stopping by favoring evaluations that extend to larger budgets, while tree-structured Parzen estimators have been shown to outperform Gaussian Processes in early stopping contexts due to their flexibility and scalability [15, 53]. Budget-conscious evolutionary algorithms have also been effectively combined with early stopping [3]. In general, the success of these strategies hinges on the performance metrics collected in step ⑤, which are essential for ensuring the reliability of HPO results.

3 EMPIRICAL STUDY ON EARLY STOPPING METRICS FOR HPO

In early stopping-based HPO, performance metrics play a critical role in assessing each configuration's capability at a specific fidelity level, informing the early stopping decision for filtering. An effective metric should consistently reflect a model's present performance and its potential for improvement. Despite extensive

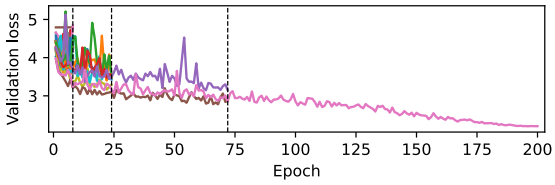


Figure 2: One iteration of SH — An example of running one round of SH on the ImageNet-16-120 benchmark, where $R = 27$, $\eta = 3$, and one unit of resource corresponds to 8 epochs.

Algorithm 1: Pseudocode for Successive Halving (SH).

Input : initial budget b_0, maximum budget R, filtering ratio η, and set of n configurations T

1 $b = b_0$
2 **while** $b \leq R$ **do**
3 $\quad L = \{\hat{f}^b(\gamma) : \gamma \in T\}$ `// early stopping metrics`
4 $\quad T = \text{top}_k(T, L, \lfloor |T| \cdot \eta^{-1} \rfloor)$ `// ranking`
5 $\quad b = \eta \cdot b$
6 **end**

Algorithm 2: Pseudocode for target HPO algorithms.

Input : budget R, filtering ratio η

1 $s_{max} = \lfloor \log_\eta(R) \rfloor$
2 **for** $s \in \{s_{max}, s_{max} - 1, \ldots, 0\}$ **do**
3 $\quad$ sample $n = \lceil \frac{s_{max}+1}{s+1} \cdot \eta^s \rceil$ configurations T run SH on T with $R \cdot \eta^{-s}$ as initial budget
4 **end**

research on early stopping-based HPO methods, there is a notable absence of consensus regarding the selection and reliability of metrics for guiding early stopping decisions. In this section, we address this gap by empirically comparing commonly used metrics to unveil underlying differences. We start by detailing the experimental setup applied across all experiments throughout this paper.

3.1 Experimental Setup

3.1.1 Target HPO Algorithm. We assess the reliability of common metrics within single-objective classification tasks using Successive Halving and Hyperband algorithms.

Successive Halving (SH) [25] employs a heuristic approach to allocate increasing resources to the most promising configurations based on their performance metrics. As outlined in Algorithm 1, SH begins with a predefined set of configurations, budget limits, and a filtering ratio. Each configuration is initially evaluated with the lowest budget (line 3), and only the top-performing fraction, defined by $1/\eta$, is retained for the next round (line 4). This iterative process of filtering and reallocating resources continues until one configuration receives the maximum budget. Figure 2 illustrates a single iteration of the SH algorithm.

Hyperband [31] extends SH by better balancing the number of configurations and budget allocation. As outlined in Algorithm 2, Hyperband takes two key parameters: 1) the maximum budget R per configuration, and 2) the filtering ratio η, which controls how many configurations advance in each SH round. These parameters

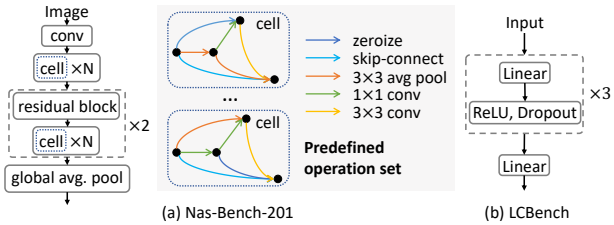


Figure 3: The Skeleton of Architectures in Nas-Bench-201 and LCBench — (a) Each cell in Nas-Bench-201 is a DAG, where each edge is associated with a candidate operation selected from a predefined operation set. (b) Shaped MLPs with ReLUs and dropouts.

together define the maximum number of iterations s_{max} for the inner loop (line 1). The outer loop starts with the most aggressive level $s = s_{max}$, exploring the widest set of configurations (line 2), ensuring at least one configuration receives the full budget R. Subsequent rounds progressively reduce the number of configurations by a factor of $1/\eta$ until $s = 0$, at which point all configurations are assigned the maximum budget R.

3.1.2 Target HPO Tasks. We apply SH and Hyperband on three tuning benchmarks. Detailed specifications can be found in Table 1.

Table 1: Benchmark Specifications

	Nas-Bench-201			**LCBench**
Datasets	CIFAR-10	CIFAR-100	ImageNet -16-120	Fashion-MNIST, Volkert
Train/Valid/Test	25k/25k/10k	50k/5k/5k	151k/3k/3k	4k/2k/1k
Hyperparameter	$1 \leftarrow 0$ $2 \leftarrow \{0, 1\}^*$ $3 \leftarrow \{0, 1, 2\}^*$	Candidate OPs: none, skip_connect, nor_conv_1x1, nor_conv_3x3, avg_pool_3x3		batch size: [16, 512] learning rate: $[1e^{-4}, 1e^{-1}]$ momentum: [0.1, 0.99] weight decay: $[1e^{-5}, 1e^{-1}]$ dropout: [0.0, 1.0] #layers: [1, 5] max. #units/layer: [64, 1024]

	HPOBench — NN		**HPOBench — BNN**	
Datasets	Higgs	Adult	Boston Housing	Protein Structure
Train/Valid/Test	6k/3k/1k	30k/15k/5k	0.3k/0.1k/0.1k	3k/1k/1k
Hyperparameter	batch size: [4, 256], depth: [1, 3] alpha: $[1e^{-8}, 1]$, width: [16, 1024] learning rate init: $[1e^{-5}, 1.0]$		burn in ratio: [0, 0.8], mdecay: [0, 1] $\#units_1$: [16, 512], $\#units_2$: [16, 512] learning rate: $[1e^{-6}, 0.1]$	

Neural Architecture Search (NAS) [11] provides offline evaluations of network architectures on three datasets: CIFAR-10, CIFAR-100, and ImageNet-16-120. It features a search space represented by a densely connected DAG, as shown in Figure 3 (a). The DAG contains four nodes, labeled 0-3 in Table 1. Edges connecting these nodes are hyperparameters selected from five candidate operations listed in Table 1. All configurations are trained for 200 epochs.

Tabular Classification. We use LCBench [60], which contains offline evaluations of shaped MLP models on OpenML datasets, as shown in Figure 3 (b). The search space includes seven hyperparameters: five are standard training parameters such as regularization and learning rate, and two relate to architecture. LCBench evaluates 2k configurations under three budgets (12, 25, and 50 epochs). We choose the maximum budget of 50 epochs for our experiments.

HPOBench. We use HPOBench [13], focusing on the Neural Network (NN) and Bayesian Neural Network (BNN) benchmarks. These

"raw" benchmarks require manual training from scratch, allowing full access to training statistics across all epochs and supporting multiple trials with varying random seeds. In contrast, "tabular" or "surrogate" benchmarks that offer limited fidelity or rely on performance prediction are excluded. The NN benchmark optimizes five hyperparameters for a feedforward neural network trained on OpenML datasets. The BNN benchmark involves five hyperparameters and is applied to regression tasks from the UCI repository [12].

3.1.3 Methodology. The early stopping metrics we examine include training loss and validation loss. To validate the reliability of these metrics, we conduct experiments across various budget constraints (R) and filtering ratios (η) within the Hyperband algorithm, as detailed in Table 2. For each setting, we perform 1000 repetitions with different random seeds, each including a randomly selected subset of model configurations. We compare the outcomes of early stopping decisions guided by different metrics, and we employ the Wilcoxon signed-rank test [49] to determine the presence of significant differences among the metrics. To report results, we consider indicators such as *final performance*, *performance over time*, and *performance regret* (defined as the discrepancy between the best-found value and the best-known value) [13].

Table 2: Early Stop Settings — R denotes the maximum budget available to a single configuration. η determines the proportion of configurations that persist in every early stopping round.

	Nas-Bench-201	LCBench	HPOBench—NN	HPOBench—BNN
R	50, 81, 160, 180	10, 15, 30, 45	40, 80, 120, 160	2500, 5000, 7500, 10000
η	3, 1.33	3, 1.33	3, 1.33	3, 1.33

3.2 Observed Reliability of Common Metrics

To investigate how the choice of metrics–training loss versus validation loss–affects early stopping strategies in HPO, we first conduct a series of experiments on the Nas-Bench-201 benchmark, using random sampling combined with SH for model selection.

Figure 4 compares final test accuracies achieved when using training versus validation losses as early stopping metrics. Figure 4 (a) shows the distribution of differences in final test accuracy between models selected respectively by these two metrics across varying budgets. The differences are computed by subtracting the test accuracies achieved using validation loss from that achieved using training loss. Contrary to the conventional preference for validation metrics, our findings in the complex Nas-Bench-201 task indicate that training metrics consistently outperform validation metrics in selecting better models across all budget levels, with an average accuracy difference of 0.72% and a maximum of 24.76%.

Additionally, Figure 4 (b) explores how the disparity between early stopping decisions informed by these metrics evolves under a computational budget of 150. Performance regret, defined here as the gap between the best-performing model found by HPO and the true optimum among explored candidates, shows that relying on validation loss for early stopping frequently halts training prematurely, yielding sub-optimal final selections.

These observations underscore that the choice of metric – training versus validation – significantly impacts the outcomes of HPO.

Table 3: Analysis of Common Metrics with the Wilcoxon Signed-Rank Test — The hypothesis posits that training loss $m_{\mathcal{T}}$ is superior to validation loss $m_{\mathcal{V}}$. A p-value closer to zero indicates higher reliability of training loss, while to one suggests better performance of validation loss. d_{acc} (d_{loss}) represents the average difference in final test accuracy (loss) between two metrics.

Nas-Bench-201		CIFAR-10	CIFAR-100	ImageNet-16-120	*LCBench*	Volkert	Fashion-MNIST
$R = 50$	p	$7.6e^{-109}$	$2.2e^{-78}$	$6.6e^{-55}$	$R = 9$	$1.0e^{-9}$	$5.4e^{-6}$
$\eta = 3$	d_{acc}	0.940	1.009	0.912	$\eta = 3$	0.187	0.060
$R = 81$	p	$4.0e^{-119}$	$3.1e^{-79}$	$7.8e^{-49}$	$R = 15$	$7.0e^{-12}$	$4.0e^{-8}$
$\eta = 3$	d_{acc}	0.938	1.021	0.763	$\eta = 3$	0.206	0.067
$R = 160$	p	$1.8e^{-40}$	$8.9e^{-153}$	$2.9e^{-27}$	$R = 30$	0.999	0.999
$\eta = 1.33$	d_{acc}	0.279	2.476	0.347	$\eta = 1.33$	-0.243	-0.071
$R = 180$	p	$6.1e^{-48}$	$2.4e^{-158}$	$3.3e^{-7}$	$R = 45$	0.999	0.999
$\eta = 1.33$	d_{acc}	0.277	3.084	0.165	$\eta = 1.33$	-0.347	-0.066

HPOBench-NN		Higgs	Adult	*HPOBench-BNN*		Boston	Protein
$R = 40$	p	$5.4e^{-92}$	$7.3e^{-17}$	$R = 2500$	p	$1.5e^{-60}$	$4.1e^{-41}$
$\eta = 3$	d_{acc}	0.003	$9.5e^{-4}$	$\eta = 3$	d_{loss}	-36.762	-1.118
$R = 80$	p	$9.9e^{-21}$	0.257	$R = 5000$	p	$2.8e^{-145}$	$1.7e^{-6}$
$\eta = 3$	d_{acc}	0.001	$1.0e^{-4}$	$\eta = 3$	d_{loss}	-94.295	-1.167
$R = 120$	p	0.248	0.999	$R = 7500$	p	$1.4e^{-115}$	$3.7e^{-34}$
$\eta = 1.33$	d_{acc}	$1.1e^{-4}$	-0.001	$\eta = 1.33$	d_{loss}	-79.010	-0.015
$R = 160$	p	0.001	0.999	$R = 10000$	p	$4.0e^{-7}$	$3.2e^{-5}$
$\eta = 1.33$	d_{acc}	$7.1e^{-4}$	$9.3e^{-4}$	$\eta = 1.33$	d_{loss}	-9.381	-0.009

Interestingly, our findings challenge the widespread practice of favoring validation metrics, showing that training metrics can be more effective in guiding early stopping in HPO.

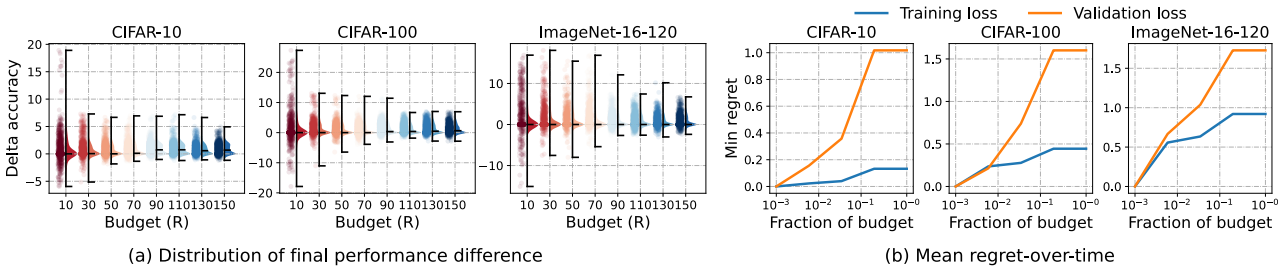
In conjunction with qualitative analysis, we employ the Wilcoxon signed-rank test to statistically evaluate the differences between training and validation metrics. Assuming that the final test results derived from training loss exceed that from validation loss, we report the resulting p-values alongside observed performance differences in Table 3. The results reveal marked disparities in HPO outcomes. Initially, training loss ($m_{\mathcal{T}}$) significantly outperforms validation loss ($m_{\mathcal{V}}$) under limited budgets, especially during early training stages. For example, in Nas-Bench-201, the results at $R = 50$ and $R = 81$ show extremely low p-values and significant accuracy discrepancies. However, as budget allocation increases, validation loss progressively becomes more indicative of final performance. With larger budgets, p-values generally rise, and final discrepancies diminish. Specifically, in LCBench and NN, validation loss shows notable superiority over training loss when $R \geq 30$ and $R \geq 120$, respectively. Conversely, in Nas-Bench-201 and BNN, training loss consistently exhibits its advantage across varying budgets, highlighted by an average accuracy gap of 3% on CIFAR-100 at $R = 180$.

We distill these observations into the following insight:

Insight 1: training loss, as opposed to the commonly favored validation loss, is a more effective metric for guiding early stopping in HPO across various budgets. However, as the available budget increases, validation loss becomes more effective.

4 THE REASONS AND THEORETICAL ANALYSIS

This section examines the reasons for the observed variances in the effectiveness of the common metrics and, more importantly, reveals



(a) Distribution of final performance difference (b) Mean regret-over-time

Figure 4: SH Early Stop Comparison — SH with $\eta = 3$ across various budget constraints. (a) illustrates the distribution of differences in test accuracy obtained when employing training loss versus validation loss as early stopping metrics. (b) demonstrates the disparities in test accuracy between the optimal models selected based on these metrics and the actual optimal models throughout the tuning process.

the underlying factors and their impact on the efficacy of early stopping metrics. Before delving into a detailed analysis, we first examine the models' performance across their training lifecycles.

4.1 Model Performance over Time

Figure 5 displays the *performance-over-time* curves for validation and training losses across all target tasks. The benchmarks show distinct characteristics; Nas-Bench-201 and BNN exhibit considerable volatility in validation losses while maintaining stable training losses. Conversely, LCBench and NN show moderate fluctuations in both metrics, with more noticeable variability in validation losses. Note that in the BNN figures, thicker lines indicate greater performance fluctuations. The learning trajectories also differ; loss values in Nas-Bench-201 and BNN decline slowly over epochs, whereas LCBench and NN stabilize more rapidly. These differences can be attributed to the varying model complexities and dataset sizes outlined in Section 3.1.2: LCBench and NN utilize shallow MLP models, while BNN models weights as probability distributions, resulting in higher computational complexity. Nas-Bench-201, on the other hand, employs more complex architectures and larger datasets.

Moreover, initial loss values in Nas-Bench-201 are closely clustered and relatively high, with significant performance divergences emerging later in training. In contrast, configurations in LCBench and HPOBench show clear performance differences from the outset. These patterns suggest that early stopping-based HPO is more effective for lightweight tasks, where clear early disparities facilitate decisive early stopping. However, in Nas-Bench-201, less distinct early performance differences might lead to the premature termination of potentially superior models. Therefore, careful selection of early stopping metrics is especially crucial in Nas-Bench-201.

4.2 Theoretical Analysis

We conduct a theoretical analysis to investigate the causes of suboptimal decisions in early stopping strategies before convergence. Consider an HPO task under a supervised learning context where a model M is trained on data points $\mathcal{D}_T = \{(x_i, y_i)\}_{i=1}^n$ sampled i.i.d. from some unknown data distribution $\mathcal{U}$. Let there be K hyperparameter candidates $\gamma_1, \gamma_2, \ldots, \gamma_K \in \Gamma$. We denote the model trained with hyperparameter γ for t epochs as M_γ^t and the converged model as M_γ^*. Given loss function $\ell(\cdot,\cdot) \in [lb, ub]$ ($lb \geq 0$), the *expected risks* of models M_γ^t and M_γ^* with respect to $\mathcal{U}$ are defined as:

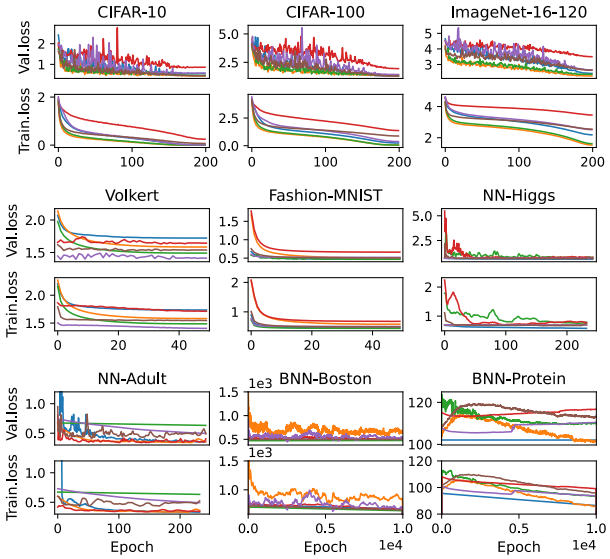


Figure 5: Performance over Time — Randomly select 6 configurations to show how losses change over time in each benchmark. Validation losses show higher fluctuation.

$$f^t(\gamma) = \mathbb{E}_{\mathcal{U}}\left[\ell\left(\mathrm{y}, M_\gamma^t(\mathbf{x})\right)\right], \text{ and } f^*(\gamma) = \mathbb{E}_{\mathcal{U}}\left[\ell\left(\mathrm{y}, M_\gamma^*(\mathbf{x})\right)\right]. \quad (1)$$

The objective of HPO is to identify hyperparameters γ_o that minimize the expected risk of converged models, expressed as $\gamma_o = \arg\min_{\gamma\in\Gamma} f^*(\gamma)$. However, the expected risk cannot be directly computed as $\mathcal{U}$ is unknown. Instead, HPO relies on estimating this risk using a finite set $\mathcal{D}$ drawn i.i.d. from the distribution $\mathcal{U}$. Thus, practical HPO centers around minimizing the *empirical estimate*:

$$\hat{f}^*(\gamma) = \frac{1}{|\mathcal{D}|}\sum_{x_i,y_i\in\mathcal{D}} \ell\big(y_i, M_\gamma^*(x_i)\big). \quad (2)$$

Early stopping (ES) actions at epoch t involve filtering models using a ranking function π on a list of empirical estimates:

$$ES\Big\{\pi\big(\hat{f}^t(\gamma_1),\ldots,\hat{f}^t(\gamma_k)\,|\,\mathcal{D}\big)\Big\} \to \{\gamma_{n_1},\ldots,\gamma_{n_t}\}, \quad (3)$$

where n_t denotes the number of configurations retained after screening. The ranking function serves as the early stopping criterion,

utilizing performance metrics at epoch t to estimate the models' true capability and guide resource allocation within the HPO process. Research into early stopping criteria often employs early performance metrics as proxies for ultimate model capability. Regardless of the specific early stopping criterion, its reliability hinges on how closely the metrics reflect the models' actual performance. We next explore factors influencing the reliability of early stopping metrics.

Proposition 4.1. *Consider an early stopping-based HPO that uses model's loss function as its early stopping metric. Let f^t and $\hat{f}^t$ denote the expected and empirical losses at any epoch t before convergence, and f^* and $\hat{f}^*$ denote the expected and empirical losses at convergence, as defined in Eqs. 1 and 2. Assume $f^t(\gamma) \geq f^*(\gamma)$ and $\hat{f}^t(\gamma) \geq \hat{f}^*(\gamma)$ hold for all $\gamma \in \Gamma$. Let $\gamma_o = \arg\min_{\gamma\in\Gamma} f^*(\gamma)$ be the optimal hyperparameter, and let γ_{so} denote a sub-optimal candidate. Then, the probability of making an incorrect early stopping decision at epoch t can be bounded according to Markov's inequality:*

$$P\big(\hat{f}^t(\gamma_o) - \hat{f}^t(\gamma_{so}) \geq 0\big) \leq \frac{1}{ub - lb}\big(f^t(\gamma_o) - f^t(\gamma_{so}) + ub - lb\big). \quad (4)$$

Using Hoeffding's inequality [23], we derive tighter bounds:

$$\begin{aligned} P\big(\hat{f}^t(\gamma_o) - \hat{f}^t(\gamma_{so}) \geq 0\big) \leq & e^{-\frac{2|D|\left(f^t(\gamma_{so}) - f^t(\gamma_o)\right)^2}{(ub-lb)^2}}, \\ & if\, f^t(\gamma_{so}) > f^t(\gamma_o) \\ P\big(\hat{f}^t(\gamma_o) - \hat{f}^t(\gamma_{so}) \geq 0\big) \geq 1 - & e^{-\frac{2|D|\left(f^t(\gamma_{so}) - f^t(\gamma_o)\right)^2}{(ub-lb)^2}}, \\ & if\, f^t(\gamma_{so}) \leq f^t(\gamma_o). \end{aligned} \quad (5)$$

Proof Sketch. Define $F = \hat{f}^t(\gamma_o) - \hat{f}^t(\gamma_{so}) + M$, where M is chosen to ensure $F \geq 0$. Assuming loss values are bounded within $[lb, ub]$ ($lb \geq 0$), the maximum offset required for F is $ub - lb$. Thus, choosing $M = ub - lb$ guarantees $F \geq 0$. Using Markov's inequality, we obtain:

$$\begin{aligned} P\big(\hat{f}^t(\gamma_o) - \hat{f}^t(\gamma_{so}) \geq 0\big) = P(F \geq M) &\leq \frac{\mathbb{E}[F]}{M} \\ &= \frac{\mathbb{E}[\hat{f}^t(\gamma_o) - \hat{f}^t(\gamma_{so}) + M]}{M}. \end{aligned}$$

Given $\mathbb{E}[\hat{f}^t(\gamma_o) - \hat{f}^t(\gamma_{so})] = \mathbb{E}[\hat{f}^t(\gamma_o)] - \mathbb{E}[\hat{f}^t(\gamma_{so})] = f^t(\gamma_o) - f^t(\gamma_{so})$, we can establish a general upper bound as Eq. 4. To formulate a tighter bound, we employ Hoeffding's inequality [23]. Let $F = \hat{f}^t(\gamma_o) - \hat{f}^t(\gamma_{so})$, we have:

$$P(F \geq 0) = P(F - \mathbb{E}[F] \geq -\mathbb{E}[F]).$$

When $\mathbb{E}[F] \leq 0$, we obtain:

$$P(F - \mathbb{E}[F] \geq -\mathbb{E}[F]) \leq e^{-\frac{2|\mathcal{D}|\mathbb{E}[F]^2}{(ub-lb)^2}} = e^{-\frac{2|\mathcal{D}|\left(f^t(\gamma_o) - f^t(\gamma_{so})\right)^2}{(ub-lb)^2}}.$$

When $\mathbb{E}[F] > 0$, we obtain:

$$\begin{aligned} P(F - \mathbb{E}[F] \geq -\mathbb{E}[F]) &= 1 - P(F - \mathbb{E}[F] < -\mathbb{E}[F]) \\ &\geq 1 - e^{-\frac{2|\mathcal{D}|\mathbb{E}[F]^2}{(ub-lb)^2}} = 1 - e^{-\frac{2|\mathcal{D}|\left(f^t(\gamma_o) - f^t(\gamma_{so})\right)^2}{(ub-lb)^2}}. \end{aligned}$$

□

This proposition identifies three key factors that impact the effectiveness of early stopping strategies in HPO: 1) the dataset size $|\mathcal{D}|$, 2) the discriminative power of metrics, quantified by $\left(f^t(\gamma_o) - f^t(\gamma_{so})\right)^2$, and 3) the range of metric values $(ub - lb)^2$.

First, a larger dataset size $|\mathcal{D}|$ tightens risk bounds, enhancing the reliability of the metrics derived. This is intuitive, as larger datasets typically reduce the variance between estimated and true quantities, which in turn increases the accuracy of the metrics in reflecting model capabilities. Second, stronger discriminative power in the metrics, quantified by $\left(f^t(\gamma_{so}) - f^t(\gamma_o)\right)^2$, also narrows risk bounds; thus, designing more discriminative metrics during training is beneficial. Third, narrowing the loss range $ub - lb$ further reduces risk bounds. Although Eqs. 4 and 5 incorporate these bounds, refining the loss value ranges relevant to specific training stages can further reduce the probability of early stopping errors. This underscores the importance of variations in loss values for metric efficacy. Notably, the ratio $\left(f^t(\gamma_o) - f^t(\gamma_{so})\right)^2/(ub - lb)^2$ in Eq. 5 acts as a form of regularization, suggesting that the effectiveness of early stopping is more significantly determined by the actual variance in metric values rather than the preset range.

These observations elucidate why, in the context of Nas-Bench-201, training loss proves to be a more reliable metric for early stopping than validation loss. Training loss accurately captures model expressiveness, particularly in the early learning phase, leading to more stable and consistent model rankings. In contrast, validation loss emphasizes generalization and can favor lower-capacity models that converge prematurely. Therefore, recognizing the different stages and the inherent uncertainties in model training is crucial, as each stage carries distinct assumptions and implications that affect model evaluation and selection.

We further demonstrate these key factors with a specific Gaussian distribution example.

Example 4.2 (Gaussian Assumption). Given the HPO task as defined in Proposition 4.1. Suppose the losses at epoch t for hyperparameters γ_o and γ_{so} follow $\mathcal{N}(\mu_o, \sigma_o^2)$ and $\mathcal{N}(\mu_{so}, \sigma_{so}^2)$, respectively. The estimates $\hat{f}^t(\gamma_o)$ and $\hat{f}^t(\gamma_{so})$ are averaged over the loss values and thus follow $\mathcal{N}(\mu_o, \frac{\sigma_o^2}{|\mathcal{D}|})$ and $\mathcal{N}(\mu_{so}, \frac{\sigma_{so}^2}{|\mathcal{D}|})$. Assume that $\hat{f}^t(\gamma_o)$ and $\hat{f}^t(\gamma_{so})$ are independent. Then, the probability of an incorrect early stopping decision at epoch t can be obtained using the Gaussian cumulative distribution function (CDF) Φ:

$$P\big(\hat{f}^t(\gamma_o) \geq \hat{f}^t(\gamma_{so})\big) = 1 - \Phi\left(\frac{\mu_o - \mu_{so}}{\sqrt{\frac{1}{|\mathcal{D}|}(\sigma_o^2 + \sigma_{so}^2)}}\right). \quad (6)$$

Proof Sketch. Let $F = \hat{f}^t(\gamma_o) - \hat{f}^t(\gamma_{so})$. Since the difference of two independent normal variables is also normally distributed, F follows $\mathcal{N}(\mu_o - \mu_{so}, \frac{\sigma_o^2}{|\mathcal{D}|} + \frac{\sigma_{so}^2}{|\mathcal{D}|})$. We want to find $P(F \geq 0)$, which is given by:

$$P(F \geq 0) = P\left(\frac{F - (\mu_o - \mu_{so})}{\sqrt{\frac{1}{|\mathcal{D}|}(\sigma_o^2 + \sigma_{so}^2)}} \geq \frac{\mu_{so} - \mu_o}{\sqrt{\frac{1}{|\mathcal{D}|}(\sigma_o^2 + \sigma_{so}^2)}}\right).$$

Now, variable $\frac{F - (\mu_o - \mu_{so})}{\sqrt{\frac{1}{|\mathcal{D}|}(\sigma_o^2 + \sigma_{so}^2)}}$ follows $\mathcal{N}(0, 1)$. According to the CDF of standard normal distribution, we obtain:

$$P(F \geq 0) = 1 - \Phi\left(\frac{\mu_{so} - \mu_o}{\sqrt{\frac{1}{|\mathcal{D}|}(\sigma_o^2 + \sigma_{so}^2)}}\right).$$

□

The monotonically increasing pattern of the CDF curve for a standard normal distribution highlights the benefits of using a larger dataset ($|\mathcal{D}|$) and more discriminative metrics ($\mu_{so} - \mu_o$). It also illustrates how variations in the metrics ($\sigma_o^2 + \sigma_{so}^2$) increase the risk of incorrect early stopping decisions.

Remark. We assume a Gaussian distribution for the model's performance metrics for several reasons. First, the equivalence between infinitely wide DNNs and Gaussian processes ensures that the model outputs follow a Gaussian distribution [29]. Second, Performance metrics—typically computed as aggregates (e.g., sums or averages) of these outputs—converge to a Gaussian distribution via the Central Limit Theorem when the individual terms are independent or weakly correlated. This assumption is widely used in probabilistic modeling and provides a mathematically tractable framework for uncertainty estimation [1, 34, 40, 59].

From the discussion, we distill the following insight:

Insight 2: metrics derived from large datasets and those highlighting significant differences between model configurations improve early stopping, while variability in these metrics increases error risks. Understanding distinct training stages and uncertainties is crucial for optimal metric selection.

5 MODEL UNCERTAINTY AND EARLY STOPPING

This section presents the first known exploration of model uncertainty's effects on early stopping and its incorporation into early stopping in HPO. Model uncertainty, which embodies inherent prediction variability, can destabilize the model and obscure its true capacity while also offering valuable insights into its latent potential. We next delve into the manifestations and impacts of model uncertainty and demonstrate how leveraging uncertainty can guide the formulation of more reliable early stopping decisions.

5.1 Manifestations of Uncertainty

Uncertainty in ML arises from two primary sources: intrinsic noise within the data and variability in model predictions stemming from limited knowledge [1, 24]. Since data uncertainty remains a constant, it is the variability in model predictions, referred to as model uncertainty, that predominantly affects early stopping decisions.

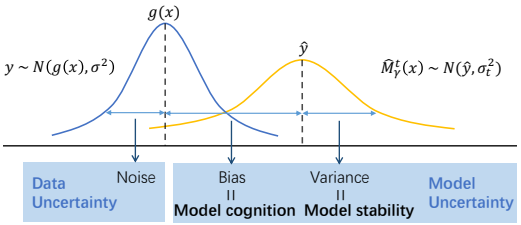


Figure 6: Decomposition of uncertainty.

For clarity and in alignment with prior research on uncertainty in machine learning [59], we assume that model predictions follow a normal distribution, $\hat{M}_y^t(\mathbf{x}) \sim \mathcal{N}(\hat{\mathbf{y}}, \sigma_t^2)$, where $\hat{\mathbf{y}} = \mathbb{E}\hat{M}_y^t(\mathbf{x})$. Let $\mathbf{y} = g(\mathbf{x}) + \varepsilon$ be an observation corresponding to a given input $\mathbf{x} \in \mathbb{R}^d$, perturbed by noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. Therefore, $\mathbf{y} \sim \mathcal{N}(g(\mathbf{x}), \sigma^2)$, where $g(\mathbf{x})$ represents the ground truth, and σ represents the noise arising from data, which is fundamentally irreducible.

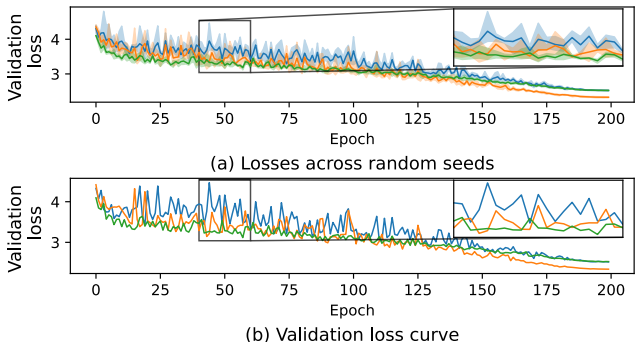


Figure 7: Manifestation of model uncertainty — (a) and (b) display inter-seed and inter-epoch uncertainty in validation losses for three randomly selected models in ImageNet-16-120, respectively.

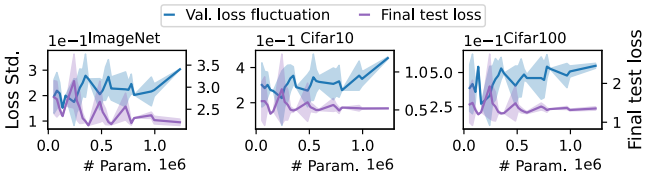


Figure 8: Early-stage fluctuation and final test loss vs. model capacity. — Correlation between the std. of inter-epoch validation loss from the first five epochs and final test loss, with model capacity quantified by the number of model parameters involved in NAS.

Figure 6 shows the distributions of observations and model predictions for a specific $\mathbf{x}$ at epoch t, illustrating uncertainty decomposition. The "*Bias*", quantifying the gap between $\hat{\mathbf{y}}$ and $g(\mathbf{x})$, reflects the model's learning capacity under various training configurations (e.g., hyperparameters, fidelity, learning algorithms). The "*Variance*" represents the model's sensitivity to training samples and encompasses both the model's current learning level and the uncertainty associated with stability. Good early stopping decisions should primarily rely on the model's current learning capability.

The impact of training variation manifests in two aspects: First, different training settings, such as initialization and data batch loading order, introduce variability to the learning process, enabling models to capture diverse facets of the data and leading to different bias and error patterns. Figure 7 (a) shows the mean and variance of validation losses across three random seeds, highlighting this effect. Leveraging this variability can offer deeper insight into the model's learning capabilities. Second, model uncertainty is evident through substantial fluctuations in consecutive epochs, especially in the validation set before the model fully adapts to underlying data patterns. As demonstrated in Figure 7 (b), significant oscillations in validation losses occur during the first 100 epochs. These fluctuations diminish as the model approaches convergence.

Furthermore, Figure 8 unveils a noteworthy pattern: models with higher expressive capacity (i.e., larger model sizes and lower final

test losses) tend to exhibit greater early-stage fluctuations. This stems from the fact that stronger models carry more uncertainty in their initial stages when their knowledge has not yet aligned with their expressive potential. In contrast, models with limited expressiveness converge sooner and display less uncertainty. Therefore, making early stopping decisions at peaks of uncertainty may lead to the premature termination of more capable models.

5.2 Implications of Uncertainty Integration

Next, we explore strategies for integrating uncertainty into early stopping metric to aid in making informed early stopping decisions.

5.2.1 Benefits of Integrating Uncertainty. Observations from Figure 7 reveal considerable fluctuations in model predictions throughout training. Building upon the insights from Section 4, improving the stability of these metrics is posited to enhance the precision of early stopping decisions. Therefore, we delve into the effect of integrating uncertainty into performance metrics on HPO outcomes. In the absence of an exact mathematical model for model's performance distribution, we opt for the simplifying assumption of a Gaussian distribution following Example 4.2.

Example 5.1 (Uncertainty Across Random Seeds). Given the HPO context defined in Example 4.2, we consider a scenario where each model configuration is trained R times with distinct random seeds. Let $\hat{f}_r^t(\gamma_o)$ and $\hat{f}_r^t(\gamma_{so})$ represent the empirical risk at epoch t for each independent seed r, which follow $N(\mu_o, \frac{\sigma_o^2}{|\mathcal{D}|})$ and $N(\mu_{so}, \frac{\sigma_{so}^2}{|\mathcal{D}|})$, respectively. To exploit the inter-seed uncertainty, we introduce a new early stopping metric $m_e^t(\gamma) = \frac{1}{R}\sum_{r=1}^{R} \hat{f}_r^t(\gamma)$ for each hyperparameter γ, as an alternative to $\hat{f}^t(\gamma)$ for model ranking in Eq. 3. $m_e^t(\gamma_o)$ and $m_e^t(\gamma_{so})$ follow $\mathcal{N}(\mu_o, \frac{\sigma_o^2}{R|\mathcal{D}|})$ and $\mathcal{N}(\mu_{so}, \frac{\sigma_{so}^2}{R|\mathcal{D}|})$. Then, the probability of an incorrect early stopping decision is:

$$P\big(\hat{m}_e^t(\gamma_o) \geq \hat{m}_e^t(\gamma_{so})\big) = 1 - \Phi\left(\frac{\mu_{so} - \mu_o}{\sqrt{\frac{1}{R|\mathcal{D}|}(\sigma_o^2 + \sigma_{so}^2)}}\right). \tag{7}$$

Remark. When models are trained under identical conditions but with different seeds, each trial remains independent. In this context, assuming that outcomes from different seeds follow the same normal distribution is statistically justified by the Central Limit Theorem. Averaging results across independent trials reduces variances in performance metrics, thus lowering the risk of premature early stopping errors. However, the computational costs of multiple training trials constrain the feasibility of this approach in practice.

Example 5.2 (Uncertainty Across Adjacent Epochs). Given the HPO context defined in Example 4.2, consider a model with hyperparameter γ, where the empirical risks $\hat{f}^t(\gamma)$ over a small window of W consecutive epochs follow normal distributions with a constant mean μ but distinct standard deviations $\sigma^{t_1}, \ldots, \sigma^{t_W}$. To exploit the inter-epoch uncertainty, we introduce a new early stopping metric, $\hat{m}_w^t(\gamma) = \frac{1}{W}\sum_{w=t_1}^{t_W} \hat{f}^w(\gamma)$, $t = \lfloor (t_1 + t_W)/2 \rfloor$, which calculates the average empirical risk over the window W. Accordingly, $\hat{m}_w^t(\gamma_o)$ and $\hat{m}_w^t(\gamma_{so})$ are distributed as $\mathcal{N}(\mu_o, \frac{1}{W^2|\mathcal{D}|}\sum_{w=t_1}^{t_W} \sigma_o^{w2})$ and $\mathcal{N}(\mu_{so}, \frac{1}{|W^2\mathcal{D}|}\sum_{w=t_1}^{t_W} \sigma_{so}^{w2})$, respectively. The metric $\hat{m}_e^t(\gamma)$ replaces $\hat{f}^t(\gamma)$ for model ranking, as specified in Eq. 3. Then, the probability of an incorrect early stopping decision is:

$$P\big(\hat{m}_w^t(\gamma_o) \geq \hat{m}_w^t(\gamma_{so})\big) = 1 - \Phi\left(\frac{\mu_{so} - \mu_o}{\frac{1}{W^2|\mathcal{D}|}\sum_{w=t_1}^{t_W}(\sigma_o^{w2} + \sigma_{so}^{w2})}\right). \tag{8}$$

Remark. Assuming a constant mean loss over a short epoch window while allowing standard deviations to vary provides a practical approach to analyzing complex models. This simplification does not compromise the detection of significant shifts in learning behavior.

5.2.2 Uncertainty-Integrated Metrics. Building on Examples 5.1 and 5.2, we propose two metrics that integrate uncertainty to enhance early stopping decisions: the *ensemble averaging* metric m_e and the *window smoothing* metric m_w. The m_e metric leverages uncertainty across multiple training trials to improve early stopping reliability. It adopts an ensemble learning strategy in which each model is independently trained on a resampled dataset with identical configurations but with different initialization seeds [38]. By averaging performance at the same epoch, m_e provides a robust measure of model stability under varying initial conditions. We denote the metrics computed on the training and validation datasets as $m_{e_\mathcal{T}}$ and $m_{e_\mathcal{V}}$, respectively.

The m_w metric is designed to smooth out random fluctuations to refine early stopping decisions. It operates on the premise that performance fluctuations across neighboring epochs primarily reflect uncertainty rather than substantial changes in model performance. m_w computes the average performance over a predefined small window of consecutive epochs, yielding a more consistent and stable representation of the model's performance by averaging out short-term stochastic variations. Similarly, we define $m_{w_\mathcal{T}}$ and $m_{w_\mathcal{V}}$ for the training and validation datasets, respectively.

We first evaluate the ensemble averaging metric using three random seeds and apply the Wilcoxon signed-rank test to compare $m_{e_\mathcal{T}}$ with $m_\mathcal{T}$ and $m_{e_\mathcal{V}}$ with $m_\mathcal{V}$. Results from experiments on Nas-Bench-201 and HPOBench are presented in Table 4. LCBench is excluded due to a lack of training results from multiple seeds. Our findings demonstrate the efficacy of the ensemble averaging metric. First, the consistently low p-values across various budgets and filtering ratios for both training and validation losses suggest that the ensemble averaging approach significantly enhances the accuracy of early stopping decisions. This confirms that incorporating ensemble-based uncertainty quantification can substantially enhance HPO performance when training resources are sufficient. Second, the benefits of the ensemble averaging metric tend to increase with larger budgets. This indicates that the diverse training trials complement each other in reflecting model capability and uncertainty, thus enhancing assessments of the model's learning and generalization abilities.

We next evaluate the effect of the window smoothing metric. We set the window size to 5 for Nas-Bench-201 and BNN, and 3 for LCBench and NN. Again, we use the Wilcoxon signed-rank test to compare $m_{w_\mathcal{T}}$ with $m_\mathcal{T}$ and $m_{w_\mathcal{V}}$ with $m_\mathcal{V}$. Table 5 reveals significant disparities across benchmarks. In Nas-Bench-201 and BNN, the window smoothing metric substantially improves validation performance, evidenced by the low p-values of $m_{w_\mathcal{V}}$ compared to $m_\mathcal{V}$; however, its benefit diminishes with increasing budget, aligning with our observation that model loss volatility decreases over time. In contrast, $m_{w_\mathcal{T}}$ shows no significant difference from $m_\mathcal{T}$. For

Table 4: Comparison of Ensemble Averaging Metrics and Common Metrics Using the Wilcoxon Signed-Rank Test — The hypotheses are shown in the "Assumption" row. A p-value closer to zero indicates a stronger possibility that the assumption holds. d_{acc} (d_{loss}) represents the average difference in final test accuracy (loss) between the two metrics. $m_{e_\mathcal{T}}$ and $m_{e_\mathcal{V}}$ are computed using three random seeds.

		CIFAR-10		CIFAR-100		ImageNet-16-120		BNN-Protein
Assumption		$m_{e_\mathcal{T}} > m_\mathcal{T}$	$m_{e_\mathcal{V}} > m_\mathcal{V}$	$m_{e_\mathcal{T}} > m_\mathcal{T}$	$m_{e_\mathcal{V}} > m_\mathcal{V}$	$m_{e_\mathcal{T}} > m_\mathcal{T}$	$m_{e_\mathcal{V}} > m_\mathcal{V}$	$m_{e_\mathcal{T}} > m_\mathcal{T}$
$R = 50/2500$	p	$6.4e^{-8}$	$1.3e^{-9}$	$5.5e^{-16}$	$3.1e^{-17}$	$1.7e^{-13}$	$8.8e^{-14}$	$5.4e^{-9}$
$\eta = 3$	$d_{acc/loss}$	0.031	0.228	0.103	0.391	0.135	0.330	-1.289
$R = 81/5000$	p	$1.4e^{-7}$	$7.7e^{-13}$	$3.0e^{-16}$	$3.0e^{-17}$	$5.0e^{-17}$	$2.5e^{-23}$	$6.7e^{-148}$
$\eta = 3$	$d_{acc/loss}$	0.040	0.268	0.092	0.419	0.157	0.496	-1.463
$R = 160/7500$	p	$1.5e^{-21}$	$9.7e^{-13}$	$1.3e^{-16}$	$2.2e^{-20}$	$5.3e^{-23}$	$3.7e^{-58}$	$6.0e^{-48}$
$\eta = 1.33$	$d_{acc/loss}$	0.103	0.125	0.120	0.455	0.197	0.494	-1.248
$R = 180/10000$	p	$4.9e^{-40}$	$2.2e^{-14}$	$4.8e^{-13}$	0.094	$4.0e^{-26}$	$1.4e^{-41}$	$3.9e^{-120}$
$\eta = 1.33$	$d_{acc/loss}$	0.157	0.123	0.116	0.046	0.224	0.370	-3.704
		NN-Higgs		NN-Adult		BNN-Boston		BNN-Protein
Assumption		$m_{e_\mathcal{T}} > m_\mathcal{T}$	$m_{e_\mathcal{V}} > m_\mathcal{V}$	$m_{e_\mathcal{T}} > m_\mathcal{T}$	$m_{e_\mathcal{V}} > m_\mathcal{V}$	$m_{e_\mathcal{T}} > m_\mathcal{T}$	$m_{e_\mathcal{V}} > m_\mathcal{V}$	$m_{e_\mathcal{V}} > m_\mathcal{V}$
$R = 40/2500$	p	$2.1e^{-26}$	$3.3e^{-24}$	$3.0e^{-45}$	$1.3e^{-45}$	$1.2e^{-120}$	$1.2e^{-156}$	$1.0e^{-34}$
$\eta = 3$	$d_{acc/loss}$	0.002	0.002	0.002	0.001	-2.402	-3.774	-1.159
$R = 80/5000$	p	$1.3e^{-26}$	$1.0e^{-41}$	$1.0e^{-54}$	$1.3e^{-50}$	$9.7e^{-88}$	$1.0e^{-133}$	$5.0e^{-137}$
$\eta = 3$	$d_{acc/loss}$	0.002	0.003	0.002	0.002	-0.284	-3.197	-3.060
$R = 120/7500$	p	$7.5e^{-28}$	$4.9e^{-41}$	$3.8e^{-11}$	$3.3e^{-119}$	$1.8e^{-70}$	$3.5e^{-155}$	$9.8e^{-125}$
$\eta = 1.33$	$d_{acc/loss}$	0.001	0.002	0.001	0.003	-1.458	-4.730	-3.049
$R = 160/10000$	p	$2.1e^{-42}$	$1.7e^{-118}$	$4.4e^{-8}$	$7.6e^{-107}$	$1.3e^{-86}$	$2.1e^{-154}$	$1.1e^{-127}$
$\eta = 1.33$	$d_{acc/loss}$	0.001	0.004	0.002	$1.9e^{-4}$	-2.729	-4.028	-3.414

LCBench and NN, however, window smoothing metrics generally perform worse than the conventional loss metrics, likely due to rapid convergence and significant loss reductions between epochs that deviate from the assumptions outlined in Example 5.2. These findings motivate further exploration into the variability of model performance throughout training to refine early stopping metrics.

The analysis leads to the following insight:

Insight 3: leveraging uncertainty across different seeds and consecutive epochs can enhance the reliability of early stopping metrics for HPO. Specifically, harnessing inter-seed uncertainty consistently yields superior outcomes, while exploiting inter-epoch uncertainty demands more nuanced strategies and a deeper comprehension of the model's learning trajectory.

6 TRAINING STAGES AND EARLY STOPPING

The above analysis shows that employing different early stopping metrics at various training stages can lead to distinct outcomes. Figure 4 compares the effects of using training versus validation losses under different computational budgets, while Figure 7 sheds light on performance fluctuations throughout training. These observations underscore the necessity of a comprehensive understanding of model behavior in selecting effective early stopping metrics.

6.1 Evolution of Model Performance

In Section 4.1, we presented performance trends of various HPO tasks across their training cycles, offering initial insights into their complexity and effectiveness. We now extend this analysis through a statistical examination of their distinct training stages.

6.1.1 Derivative of Losses. We calculate the derivatives of losses for all model configurations, as shown in Figure 9. The derivative quantifies performance changes between adjacent epochs.

Table 5: Comparison of Window Smoothing Metrics and Common Metrics Using the Wilcoxon Signed-Rank Test — The hypotheses are shown in the "Assumption" row. A p-value closer to zero indicates a stronger possibility that the assumption holds. d_{acc} (d_{loss}) represents the average difference in final test accuracy (loss) between two metrics. The window size is set to 5 for Nas-Bench-201 and BNN, and 3 for LCBench and NN.

Nas-Bench-201		CIFAR-10		CIFAR-100		ImageNet-16-120	
Assumption		$m_{w_\mathcal{T}} > m_\mathcal{T}$	$m_{w_\mathcal{V}} > m_\mathcal{V}$	$m_{w_\mathcal{T}} > m_\mathcal{T}$	$m_{w_\mathcal{V}} > m_\mathcal{V}$	$m_{w_\mathcal{T}} > m_\mathcal{T}$	$m_{w_\mathcal{V}} > m_\mathcal{V}$
$R = 50$	p	0.482	$1.1e^{-16}$	0.499	$3.2e^{-15}$	0.417	$1.3e^{-14}$
$\eta = 3$	d_{acc}	0.0	0.331	0.0	0.398	0.001	0.386
$R = 81$	p	0.540	$1.1e^{-23}$	0.043	$7.6e^{-17}$	0.517	$9.6e^{-27}$
$\eta = 3$	d_{acc}	$8.6e^{-4}$	0.381	0.012	0.447	$1.3e^{-4}$	0.532
$R = 160$	p	0.311	0.357	$3.9e^{-4}$	$5.7e^{-26}$	0.187	$2.7e^{-28}$
$\eta = 1.33$	d_{acc}	0.004	0.043	0.033	0.548	0.004	0.326
$R = 180$	p	$1.4e^{-17}$	$5.1e^{-6}$	$5.0e^{-3}$	$3.5e^{-7}$	0.968	$7.1e^{-17}$
$\eta = 1.33$	d_{acc}	0.098	0.077	0.031	0.214	-0.011	0.209
LCBench / NN		Fashion-MNIST		Volkert		NN-Higgs	
Assumption		$m_{w_\mathcal{T}} > m_\mathcal{T}$	$m_{w_\mathcal{V}} > m_\mathcal{V}$	$m_{w_\mathcal{T}} > m_\mathcal{T}$	$m_{w_\mathcal{V}} > m_\mathcal{V}$	$m_{w_\mathcal{T}} > m_\mathcal{T}$	$m_{w_\mathcal{V}} > m_\mathcal{V}$
$R = 9/40$	p	0.999	1.0	1.0	0.998	1.0	1.0
$\eta = 3$	d_{acc}	$-5.0e^{-4}$	$-6.8e^{-4}$	-0.173	-0.076	$-3.0e^{-3}$	$-3.1e^{-3}$
$R = 15/80$	p	1.0	0.998	0.958	0.504	1.0	0.999
$\eta = 3$	d_{acc}	$-3.4e^{-4}$	$-3.3e^{-4}$	-0.059	-0.001	$-2.2e^{-3}$	$-1.4e^{-3}$
$R = 30/120$	p	0.196	0.942	0.005	0.863	0.210	0.815
$\eta = 1.33$	d_{acc}	$9.6e^{-5}$	$-7.4e^{-5}$	0.0074	-0.015	$-2.5e^{-3}$	$-4.4e^{-3}$
$R = 45/160$	p	0.973	0.800	$6.2e^{-12}$	$7.1e^{-5}$	0.624	0.918
$\eta = 1.33$	d_{acc}	$-5.9e^{-5}$	$-2.6e^{-5}$	0.205	0.092	$-2.0e^{-3}$	$-1.5e^{-3}$
NN / BNN		NN-Adult		BNN-Boston		BNN-Protein	
Assumption		$m_{w_\mathcal{T}} > m_\mathcal{T}$	$m_{w_\mathcal{V}} > m_\mathcal{V}$	$m_{w_\mathcal{T}} > m_\mathcal{T}$	$m_{w_\mathcal{V}} > m_\mathcal{V}$	$m_{w_\mathcal{T}} > m_\mathcal{T}$	$m_{w_\mathcal{V}} > m_\mathcal{V}$
$R = 40/2500$	p	0.999	0.999	$1.2e^{-5}$	$3.7e^{-10}$	0.004	$4.0e^{-4}$
$\eta = 3$	$d_{acc/loss}$	$-5.8e^{-4}$	$-5.9e^{-4}$	-0.709	-0.326	-0.103	-0.220
$R = 80/5000$	p	0.992	0.998	0.425	$2.6e^{-4}$	0.449	0.042
$\eta = 3$	$d_{acc/loss}$	$-2.3e^{-4}$	$-3.5e^{-4}$	0.015	-0.601	-0.017	-1.534
$R = 120/7500$	p	0.999	$9.1e^{-27}$	0.356	$3.6e^{-4}$	0.552	0.033
$\eta = 1.33$	$d_{acc/loss}$	$-2.0e^{-4}$	$1.2e^{-3}$	2.090	-3.063	0.014	-0.020
$R = 160/10000$	p	$1.9e^{-10}$	$5.0e^{-5}$	0.099	$2.2e^{-7}$	0.683	$3.2e^{-5}$
$\eta = 1.33$	$d_{acc/loss}$	$5.0e^{-4}$	$3.4e^{-4}$	-0.053	-4.182	0.009	-0.088

Key observations from Figure 9 include: First, except for BNN, the mean derivatives for all tasks are consistently negative. The volatility observed in BNN stems from its special training technique of modeling probability distributions; nonetheless, its loss is gradually stabilizing. This indicates an overall improvement in performance, suggesting that training processes are effective. Second, in the later training stages, both training and validation loss derivatives converge toward zero, with few significant decreases in training loss or increases in validation loss. This suggests minimal overfitting and supports the reliability of final test performance as robust HPO objectives. Third, all benchmarks exhibit notable fluctuations in validation loss derivatives, with larger shaded areas indicating higher instability. This instability undermines the reliability of early stopping decisions based on validation loss.

We employ the ruptures toolkit [44], which specializes in change-point detection, to analyze model convergence. ruptures is adept at handling non-stationary signals and is particularly effective for identifying phase transitions in model losses. We utilize its RBF kernel-based cost function for robust detection. The identified change points are marked with red lines in Figure 9.

Limiting the detection to two change points, ruptures successfully pinpoints critical transitions in model performance. The first change point marks a shift from rapid to slower learning, and the second indicates stabilization as loss derivatives approach zero. In most HPO tasks–except Volkert–training losses stabilize before

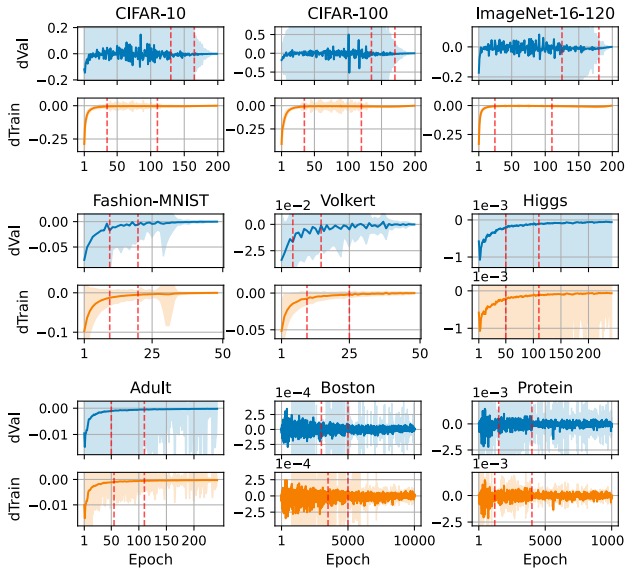


Figure 9: Derivative of Losses — The derivatives of training and validation losses over epochs. The solid line represents the mean loss across model configurations, while the shaded area denotes the range of one std. Negative derivatives suggest model improvement. Red lines mark transition points identified using ruptures.

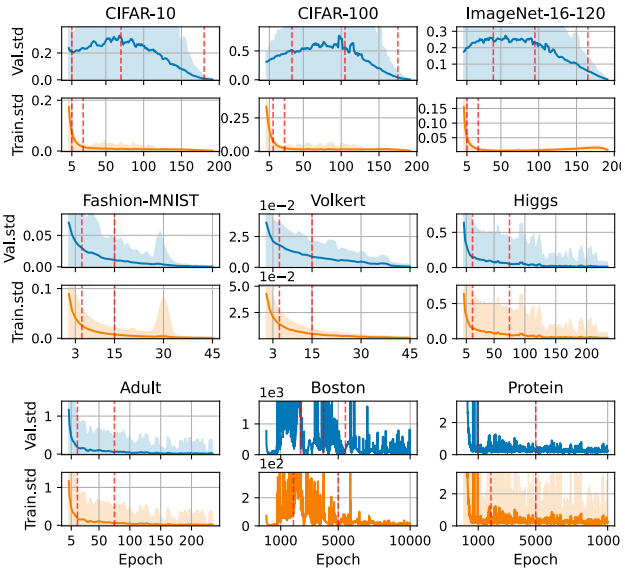


Figure 10: Standard Deviation of Losses across Consecutive Epochs — A window size of 5 is used for Nas-Bench-201 and BNN, while a size of 3 is used for LCBench and NN. The solid line represents the average std. of loss across all model configurations. The shaded area denotes the range of max. and min. values observed. Red lines mark transition points identified using ruptures.

validation losses. Specifically, in Nas-Bench-201, convergence occurs around epochs 110–120 for training losses and 160–185 for validation losses. In contrast, for LCBench, NN, and BNN, the convergence points for both losses are closely aligned (15–30, 100–120, and 4500–5000 epochs, respectively), indicating more synchronized stabilization compared to Nas-Bench-201.

6.1.2 Uncertainty in Losses. The derivative of loss serves as an indicator of model learning progress. To further assess volatility changes, we calculate the standard deviations (std.) of losses across specific epoch windows, as shown in Figure 10. A std. value approaching zero suggests that model performance is stabilizing.

Figure 10 reveals several observations: First, the training loss std. is generally smoother than validation loss std., indicating less variability. Validation losses exhibit pronounced fluctuations, as highlighted by both solid lines and shaded areas, particularly in Nas-Bench-201. Although both training and validation std. values in BNN are high, the training std. remains consistently lower. Second, as training progresses, std. values gradually decrease, reflecting reduced loss volatility. Third, in Nas-Bench-201 and Boston from HPOBench, the validation loss std. initially increases before declining. This pattern reflects models' adaptation to complex tasks. Early in training, the model adjusts from a basic state, resulting in increased fluctuations in validation losses. As training progresses, the model stabilizes and converges to optimal parameters, leading to more consistent performance and reduced std. values.

To further delineate these trends, we apply ruptures with a linear cost function to identify change points in the std. curves. For Nas-Bench-201, these change points effectively segment the learning trajectory into phases of rising, falling, and stabilizing std., offering structured insights into the model's performance evolution.

6.1.3 Learning Stage Division. Our analysis of the derivatives and stds. of losses reveals four distinct stages in model training:

(1) ***Initial Exploration.*** In this stage, the std. of validation loss increases, reflecting the model's exploratory adjustments and significant instability in generalization performance.
(2) ***Optimization.*** The second stage shows a decreasing trend in both validation and training losses and their stds. This suggests that models are refining their parameters, leading to reduced uncertainty and more stable, reliable behavior.
(3) ***Convergence.*** Training and validation losses, along with their stds, stabilize at a plateau, indicating that the models' learning and generalization capabilities have reached optimal levels.
(4) ***Potential Overfitting.*** This stage is marked by a decline in training loss coupled with an increase in validation loss. In our study, however, benchmarks are configured with a maximum number of epochs to prevent overfitting.

Table 6 details the epoch ranges for the benchmarks as determined by change-point detection. In our early stopping setups for Nas-Bench-201 and BNN-Boston, when $R \leq 81$ and $R \leq 5000$ with $\eta = 3$, the early stopping points fall within stage 1. For $R \geq 160$ and $R \geq 7500$ with $\eta = 1.33$, the early stopping points extend into stages 2 and 3. In LCBench, NN, and BNN-Protein, early stopping points for $R \leq 15$, $R \leq 80$, and $R \leq 5000$ with $\eta = 3$ are confined to stage 2, while for $R \geq 30$, $R \geq 120$, and $R \geq 7500$ with $\eta = 1.33$, they reach stage 3. This stage division elucidates why, as shown in Table 3, the benefit of the training metric diminishes under larger

budget settings. Notably, for LCBench and NN, the validation metrics at $R \geq 30$ and $R \geq 120$ significantly outperform training metrics. This systematic stage division provides a structured framework for analysis and guides the design of subsequent experiments.

Table 6: Epoch Range of Training Stages — Nas-Bench-201 and Boston encompasses all three stages, while others start from stage 2. Stage 4 is omitted due to the absence of overfitting in these tasks.

	CIFAR-10	CIFAR-100	ImageNet-16-120	BNN-Boston
Stage 1	1-70	1-95	1-95	1-2385
Stage 2	70-180	95-175	95-165	2385-5500
Stage 3	180-200	175-200	165-200	5500-10000

	Fashion-MNIST	Volkert	NN-Higgs	NN-Adult	BNN-Protein
Stage 2	1-15	1-15	1-75	1-75	1-5000
Stage 3	15-50	15-50	75-243	75-243	5000-10000

6.2 Stage-Adaptive Early Stopping

We next investigate the integration of stage information into early stopping metrics to improve decision-making. To achieve this, we develop and evaluate a set of stage-adaptive metrics.

6.2.1 Stage-Adaptive Metrics. We introduce a *stage-adaptive* metric $m_{\mathcal{T}/\mathcal{V}}$ that refines the tuning process by selectively using training or validation loss at different early stopping points. Due to the high uncertainty associated with validation loss in stage 1, $m_{\mathcal{T}/\mathcal{V}}$ initially relies on training loss. As training progresses into later stages, it transitions to validation loss. The metrics allowing this transition from stages 2 and 3 are denoted as $m^{(2-3)}_{\mathcal{T}/\mathcal{V}}$ and $m^{(3)}_{\mathcal{T}/\mathcal{V}}$, respectively. Furthermore, we explore how the integration of uncertainty with stage adaptiveness can improve early stopping decisions. We also introduce a *stage-adaptive ensemble averaging* metric $m_{e_{\mathcal{T}/\mathcal{V}}}$ and a *stage-adaptive window smoothing* metric $m_{w_{\mathcal{T}/\mathcal{V}}}$ to demonstrate their efficacy in this context.

6.2.2 Applying Stage-Adaptive Metrics in Early Stop. We begin with a comparison of the stage-adaptive metrics against conventional metrics, as presented in Table 7, excluding setups where early stopping points occur solely in stage 1. For benchmarks that do not include stage 1, we focus on comparing $m^{(3)}_{\mathcal{T}/\mathcal{V}}$, which starts using validation loss from stage 3, with $m_{\mathcal{V}}$, which starts from stage 2.

The results lead to a key conclusion: initiating early stopping with validation loss from stage 2 is less effective than consistently using training loss, while switching to validation loss in stage 3 offers some advantages. Specifically, the near 1 p-values when comparing $m^{(2-3)}_{\mathcal{T}/\mathcal{V}}$ and $m_{\mathcal{T}}$ in Nas-Bench-201 and BNN-Boston suggest that transitioning to validation loss at stage 2 adversely affects early stopping accuracy, particularly in low-budget settings where most early stopping points fall in stage 2. Nonetheless, $m^{(2-3)}_{\mathcal{T}/\mathcal{V}}$ significantly outperforms $m_{\mathcal{V}}$, confirming the benefit of retaining training loss in stage 1. Further examination shows that $m^{(3)}_{\mathcal{T}/\mathcal{V}}$ generally exceeds the performance of $m_{\mathcal{T}}$. In Nas-Bench-201, the similarity between $m^{(3)}_{\mathcal{T}/\mathcal{V}}$ and $m_{\mathcal{T}}$ is attributed to a late triggering of stage 3, where most early stopping points are concentrated in stages 1 and 2, leaving only a small subset of configurations for stage 3 evaluation. However, $m^{(3)}_{\mathcal{T}/\mathcal{V}}$ significantly outperforms $m_{\mathcal{V}}$ and $m^{(2-3)}_{\mathcal{T}/\mathcal{V}}$, reinforcing the advantages of using training loss in stage 2. In LCBench and NN, the benefits of stage adaptiveness are less pronounced but still observable. As shown in Table 3, the impact of validation loss on guiding early stopping becomes more evident at larger budgets. This is because models in LCBench and NN converge quickly, revealing their generalization ability earlier. Consequently, metrics that switch to validation loss from stage 3 onward exhibit improved performance over those relying solely on training loss, thus enhancing the reliability of final HPO outcomes.

Next, we integrate the uncertainty metrics discussed in Section 5.2 with stage-adaptive strategies. First, we consider the stage-adaptive ensemble averaging metric $m_{e_{\mathcal{T}/\mathcal{V}}}$. Previous experiments have demonstrated the advantages of the ensemble averaging metric m_e. Table 8 further shows that switching to validation loss from stage 3 significantly enhances performance compared to m_e. Second, we explore the stage-adaptive window smoothing metric $m_{w_{\mathcal{T}/\mathcal{V}}}$. Table 9 demonstrates that applying window smoothing to the stage-adaptive metrics consistently yields significant benefits across various benchmarks and budget settings. By comparing Tables 5, 7, and 9, we find that the combined approach is more effective in optimizing early stopping outcomes than either technique used separately. Overall, when metrics accurately reflect the characteristics of the training stages, introducing uncertainty, such as ensemble averaging and window smoothing techniques, significantly enhances the performance of early stopping.

6.2.3 Analysis of the Most Effective Budget Setting. We further extend our analysis to explore the most effective budget settings for the stage-adaptive window smoothing metric, which has thus far been identified as the most effective metric. Figure 11 shows representative results across four benchmarks, tracking the performance regret (i.e., the difference between HPO outcomes and optimal candidates) as the budget increases. Each budget setting was evaluated over 1,000 random repetitions. The weaker vertical dashed lines in Figure 11 mark the minimum budgets required for increasing numbers of early stopping points to appear at different stages; for example, the second green dashed line marks the smallest budget where two early stopping points occur in stage 2.

We draw several conclusions from Figure 11. First, HPO performance improves with increased budget, as evidenced by reduced mean regrets, narrower shaded areas, and higher probabilities of identifying optimal solutions. Second, the probability of finding optimal solutions in ImageNet is notably lower than in other benchmarks, likely due to the greater complexity of Nas-Bench-201, where performance differences manifest later; thus, more advanced HPO strategies are required. Third, HPO performance stabilizes once early stopping points appear in stage 3, particularly in Nas-Bench-201 and BNN, but further budget increases yield diminishing returns. This suggests a single early stopping point in stage 3 is adequate. If a performance gap of less than 4% is acceptable (or 7% for Nas-Bench-201), ensuring early stopping points occur in stage 2 is sufficient.

The analysis leads to the following insight:

Insight 4: leveraging distinct stages of model training can enhance early stopping effectiveness. Adapting to the stages

Table 7: Comparison of Stage-Adaptive Metrics and Common Metrics Using the Wilcoxon Signed-Rank Test — d_{acc} (d_{loss}) denotes the mean disparity in final test accuracy (loss) between two metrics, with "-" indicating no difference in outcomes.

Nas-Bench-201 / HPOBench		CIFAR-10				CIFAR-100				ImageNet-16-120				BNN-Protein
Assumption		$m^{(2-3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$	$m^{(2-3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$	$m^{(2-3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$	$m^{(2-3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$	$m^{(2-3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$	$m^{(2-3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$
$R = 160/7500$	p	1.0	$9.5e^{-7}$	-	$1.8e^{-40}$	1.0	$1.5e^{-15}$	-	$8.9e^{-153}$	0.999	$5.6e^{-10}$	-	$2.9e^{-27}$	1.0
$\eta = 1.33$	$d_{acc/loss}$	-0.192	0.088	0.0	0.279	-2.161	0.315	0.0	2.476	-0.208	0.139	0.0	0.348	0.014
$R = 180/10000$	p	1.0	$9.2e^{-14}$	0.683	$8.0e^{-24}$	1.0	$3.2e-5$	0.339	$2.8e^{-11}$	0.991	$3.1e^{-5}$	0.710	$4.1e^{-5}$	0.999
$\eta = 1.33$	$d_{acc/loss}$	-0.164	0.113	-0.015	0.162	-2.955	0.128	-0.029	0.283	-0.078	0.087	-0.073	0.092	0.010
LCBench / HPOBench		Fashion-MNIST		Volkert		NN-Higgs		NN-Adult		BNN-Boston				BNN-Protein
Assumption		$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$	$m^{(2-3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$	$m^{(2-3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{T}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$	$m^{(3)}_{\mathcal{T}/\mathcal{V}} > m_{\mathcal{V}}$
$R = 30/120/7500$	p	$3.8e^{-5}$	0.495	$1.7e^{-5}$	0.308	0.248	0.086	$6.4e^{-6}$	$1.6e^{-6}$	1.0	$6.3e^{-18}$	-	$1.3e^{-37}$	$5.6e^{-4}$
$\eta = 1.33$	$d_{acc/loss}$	$5.7e^{-4}$	$1.4e^{-4}$	0.189	$5.4e^{-5}$	0.001	4.549	$3.4e^{-4}$	$2.6e^{-4}$	63.128	-15.882	-0.311	-32,638	$-3.5e^{-4}$
$R = 45/160/10000$	p	$1.4e^{-5}$	0.795	0.502	0.279	$9.3e^{-5}$	0.002	$1.2e^{-44}$	0.011	$1.2e^{-12}$	$5.1e^{-19}$	$1.4e^{-14}$	$1.8e^{-18}$	0.005
$\eta = 1.33$	$d_{acc/loss}$	$5.6e^{-4}$	$9.9e^{-5}$	$2.5e^{-4}$	$3.7e^{-5}$	0.031	$1.9e^{-4}$	0.001	$2.0e^{-4}$	-12.871	-22.252	-12.909	-22.290	-0.389

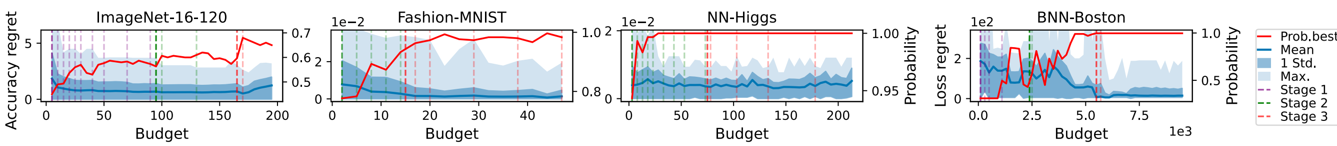


Figure 11: Performance Regret across Budgets. —The solid blue line shows mean regret. The dark and light blue areas indicate one std. and max. values, respectively. Strong vertical dashed lines mark the stage division points. Weaker lines mark the min. budgets required for increasing numbers of early stopping points in each stage. The solid red line denotes the probability of HPO identifying optimal solutions.

Table 8: Effect of Combining Stage-Adaptive Strategy with Ensemble Averaging — Using Wilcoxon Signed-Rank Test. Hypotheses are shown in the "Assumption" row. "-" indicates no difference in outcomes. Metrics are calculated with three random seeds.

		CIFAR-10		CIFAR-100		ImageNet-16-120		BNN-Protein
Assumption		$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{T}}}$	$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{V}}}$	$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{T}}}$	$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{V}}}$	$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{T}}}$	$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{V}}}$	$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{T}}}$
$R = 160/7500$	p	-	$3.0e^{-26}$	-	$1.4e^{-133}$	-	0.012	$7.8e^{-4}$
$\eta = 1.33$	$d_{acc/loss}$	0.0	0.158	0.0	1.839	0.0	0.044	-1.693
$R = 180/10000$	p	$3.8e^{-39}$	$6.0e^{-59}$	$1.6e^{-151}$	$7.3e^{-157}$	0.001	0.002	$8.9e^{-3}$
$\eta = 1.33$	$d_{acc/loss}$	0.181	0.253	2.775	2.938	0.072	0.068	-0.289
		NN-Higgs		NN-Adult		BNN-Boston		BNN-Protein
Assumption		$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{T}}}$	$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{V}}}$	$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{T}}}$	$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{V}}}$	$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{T}}}$	$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{V}}}$	$m^{(3)}_{e_{\mathcal{T}/\mathcal{V}}} > m_{e_{\mathcal{V}}}$
$R = 120/7500$	p	$6.0e^{-7}$	0.058	$7.5e^{-3}$	0.014	$1.6e^{-36}$	$1.9e^{-3}$	$2.8e^{-3}$
$\eta = 1.33$	$d_{acc/loss}$	$5.5e^{-4}$	$4.0e^{-5}$	0.002	$9.4e^{-5}$	-0.629	-0.054	-0.493
$R = 160/10000$	p	$8.3e^{-3}$	0.337	-	0.184	0.065	0.150	$2.1e^{-4}$
$\eta = 1.33$	$d_{acc/loss}$	$4.0e^{-4}$	$4.5e^{-6}$	0.0	$2.4e^{-5}$	-0.595	-0.384	-0.345

Table 9: Effect of Combining Stage-Adaptive Strategy with Window Smoothing — Using Wilcoxon Signed-Rank Test. Hypotheses are shown in the "Assumption" row. The window size is set to 5 for Nas-Bench-201 and BNN, and 3 for LCBench and NN.

		CIFAR-10		CIFAR-100		ImageNet-16-120	
Assumption		$m^{(2-3)}_{w_{\mathcal{T}/\mathcal{V}}} > m^{(2-3)}_{\mathcal{T}/\mathcal{V}}$	$m^{(3)}_{w_{\mathcal{T}/\mathcal{V}}} > m^{(3)}_{\mathcal{T}/\mathcal{V}}$	$m^{(2-3)}_{w_{\mathcal{T}/\mathcal{V}}} > m^{(2-3)}_{\mathcal{T}/\mathcal{V}}$	$m^{(3)}_{w_{\mathcal{T}/\mathcal{V}}} > m^{(3)}_{\mathcal{T}/\mathcal{V}}$	$m^{(2-3)}_{w_{\mathcal{T}/\mathcal{V}}} > m^{(2-3)}_{\mathcal{T}/\mathcal{V}}$	$m^{(3)}_{w_{\mathcal{T}/\mathcal{V}}} > m^{(3)}_{\mathcal{T}/\mathcal{V}}$
$R = 160$	p	0.239	0.311	$1.7e^{-12}$	$3.9e^{-4}$	$8.3e^{-23}$	0.187
$\eta = 1.33$	d_{acc}	0.031	0.004	0.379	0.033	0.233	0.004
$R = 180$	p	0.041	0.001	$4.3e^{-10}$	$2.7e^{-8}$	$1.0e^{-13}$	$5.2e^{-14}$
$\eta = 1.33$	d_{acc}	0.023	0.024	0.269	0.226	0.171	0.164
		Fashion-MNIST	Volkert	NN-Higgs	NN-Adult	BNN-Boston	BNN-Protein
Assumption		$m^{(3)}_{w_{\mathcal{T}/\mathcal{V}}} > m^{(3)}_{\mathcal{T}/\mathcal{V}}$	$m^{(3)}_{w_{\mathcal{T}/\mathcal{V}}} > m^{(3)}_{\mathcal{T}/\mathcal{V}}$	$m^{(3)}_{w_{\mathcal{T}/\mathcal{V}}} > m^{(3)}_{\mathcal{T}/\mathcal{V}}$	$m^{(3)}_{w_{\mathcal{T}/\mathcal{V}}} > m^{(3)}_{\mathcal{T}/\mathcal{V}}$	$m^{(3)}_{w_{\mathcal{T}/\mathcal{V}}} > m^{(3)}_{\mathcal{T}/\mathcal{V}}$	$m^{(3)}_{w_{\mathcal{T}/\mathcal{V}}} > m^{(3)}_{\mathcal{T}/\mathcal{V}}$
$R = 30/120/7500$	p	0.729	0.229	0.823	$3.8e^{-13}$	0.046	$3.8e^{-3}$
$\eta = 1.33$	$d_{acc/loss}$	$7.0e^{-6}$	0.024	$6.3e^{-3}$	$8.8e^{-4}$	-1.087	-0.103
$R = 45/160/10000$	p	0.189	$2.0e^{-21}$	0.624	0.018	$4.3e^{-4}$	$7.8e^{-4}$
$\eta = 1.33$	$d_{acc/loss}$	$1e^{-4}$	0.351	$1.4e^{-3}$	$9.8e^{-4}$	-1.048	$-4.2e^{-5}$

and integrating uncertainty into early stopping metrics can amplify the benefits. Ensuring early stopping in stage 3 is optimal, while additional budget leads to diminishing returns.

7 DISCUSSION AND FUTURE WORK

This paper presents the first known systematic study on early stopping metrics in HPO, and introduces model uncertainty into this context. Our study yields several guidelines for metric selection that benefit both users and HPO tools: (i) utilize training loss for tasks with slow convergence; (ii) capture uncertainty across different training trials and neighboring epochs; (iii) identify model training stages and select metrics based on uncertainty levels; and (iv) integrate stage adaptation and uncertainty into metric design.

Our findings offer practical guidance for future HPO research. Potential directions include developing precise techniques for characterizing model uncertainty and training stages, exploring dynamic metric-switching frameworks, and integrating uncertainty as a secondary optimization objective. We also advocate for task-specific early stopping metrics tailored to various datasets and model architectures, which necessitates dedicated, in-depth studies. Such efforts are expected to yield deeper insights into model selection and budget allocation in HPO.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No. 62461146205 and 62322213), Beijing Nova Program (No. 20230484397 and 20220484137), CCF - ApsaraDB Research Fund (CCF-Aliyun2024003), and the Outstanding Innovative Talents Cultivation Funded Programs 2023 of Renmin University of China. Xipeng Shen's work is supported by the United States Department of Agriculture (USDA) under Grant No. P24-001771. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of USDA. Jiawei Guan and Feng Zhang are co-first authors. Both authors contributed equally to this research. Xiaoyong Du is the corresponding author.

REFERENCES

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion* 76 (2021), 243–297.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2623–2631.

[3] Noor Awad, Neeratyoy Mallik, and Frank Hutter. 2021. Dehb: Evolutionary hyperband for scalable, robust and efficient hyperparameter optimization. *arXiv preprint arXiv:2105.09821* (2021).

[4] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. 2002. Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 1–16.

[5] Archit Bansal, Danny Stoll, Maciej Janowski, Arber Zela, and Frank Hutter. 2022. Jahs-bench-201: A foundation for research on joint architecture and hyperparameter search. *Advances in Neural Information Processing Systems* 35 (2022), 38788–38802.

[6] Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, et al. 2023. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 13, 2 (2023), e1484.

[7] Matthias Boehm, Iulian Antonov, Sebastian Baunsgaard, Mark Dokter, Robert Ginthör, Kevin Innerebner, Florijan Klezin, Stefanie Lindstaedt, Arnab Phani, Benjamin Rath, et al. 2019. SystemDS: A declarative machine learning system for the end-to-end data science lifecycle. *arXiv preprint arXiv:1909.02976* (2019).

[8] Baoqing Cai, Yu Liu, Ce Zhang, Guangyu Zhang, Ke Zhou, Li Liu, Chunhua Li, Bin Cheng, Jie Yang, and Jiashu Xing. 2022. HUNTER: an online cloud database hybrid tuning system for personalized requirements. In *Proceedings of the 2022 International Conference on Management of Data*. 646–659.

[9] Surajit Chaudhuri and Vivek Narasayya. 2007. Self-tuning database systems: a decade of progress. In *Proceedings of the 33rd international conference on Very large data bases*. 3–14.

[10] Piali Das, Nikita Ivkin, Tanya Bansal, Laurence Rouesnel, Philip Gautier, Zohar Karnin, Leo Dirac, Lakshmi Ramakrishnan, Andre Perunicic, Iaroslav Shcherbatyi, et al. 2020. Amazon SageMaker Autopilot: a white box AutoML solution at scale. In *Proceedings of the fourth international workshop on data management for end-to-end machine learning*. 1–7.

[11] Xuanyi Dong and Yi Yang. 2020. Nas-bench-201: Extending the scope of reproducible neural architecture search. *arXiv preprint arXiv:2001.00326* (2020).

[12] D. Dua and C. Graff. [n.d.]. UCI machine learning repository. https://archive.ics.uci.edu/. Accessed: 2024-09-12.

[13] Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, René Sass, Aaron Klein, Noor Awad, Marius Lindauer, and Frank Hutter. 2021. HPOBench: A collection of reproducible multi-fidelity benchmark problems for HPO. *arXiv preprint arXiv:2109.06716* (2021).

[14] Radwa Elshawi, Mohamed Maher, and Sherif Sakr. 2019. Automated machine learning: State-of-the-art and open challenges. *arXiv preprint arXiv:1906.02287* (2019).

[15] Stefan Falkner, Aaron Klein, and Frank Hutter. 2018. BOHB: Robust and efficient hyperparameter optimization at scale. In *International conference on machine learning*. PMLR, 1437–1446.

[16] Alexander IJ Forrester, András Sóbester, and Andy J Keane. 2007. Multi-fidelity optimization via surrogate modelling. *Proceedings of the royal society a: mathematical, physical and engineering sciences* 463, 2088 (2007), 3251–3269.

[17] Omid Gheibi, Danny Weyns, and Federico Quin. 2021. Applying machine learning in self-adaptive systems: A systematic literature review. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 15, 3 (2021), 1–37.

[18] Victor Giannakouris and Immanuel Trummer. 2024. Demonstrating λ-Tune: Exploiting Large Language Models for Workload-Adaptive Database System Tuning. In *Companion of the 2024 International Conference on Management of Data*. 508–511.

[19] Google. [n.d.]. *Google Vertex AI*. https://cloud.google.com/vertex-ai

[20] Jiawei Guan, Feng Zhang, Jiesong Liu, Hsin-Hsuan Sung, Ruofan Wu, Xiaoyong Du, and Xipeng Shen. 2022. Trec: Transient redundancy elimination-based convolution. *Advances in Neural Information Processing Systems* 35 (2022), 26578–26589.

[21] Jiawei Guan, Feng Zhang, Siqi Ma, Kuangyu Chen, Yihua Hu, Yuxing Chen, Anqun Pan, and Xiaoyong Du. 2023. Homomorphic Compression: Making Text Processing on Compression Unlimited. *Proc. ACM Manag. Data* 1, 4, Article 271 (Dec. 2023), 28 pages. https://doi.org/10.1145/3626765

[22] Wei Guo, Fuzhen Zhuang, Xiao Zhang, Yiqi Tong, and Jin Dong. 2024. A comprehensive survey of federated transfer learning: challenges, methods and applications. *Frontiers of Computer Science* 18, 6 (2024), 186356.

[23] Wassily Hoeffding. 1994. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding* (1994), 409–426.

[24] Eyke Hüllermeier and Willem Waegeman. 2021. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning* 110 (2021), 457–506.

[25] Kevin Jamieson and Ameet Talwalkar. 2016. Non-stochastic best arm identification and hyperparameter optimization. In *Artificial intelligence and statistics*. PMLR, 240–248.

[26] Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. 2017. Multi-fidelity bayesian optimisation with continuous approximations. In *International Conference on Machine Learning*. PMLR, 1799–1808.

[27] Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. 2017. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial intelligence and statistics*. PMLR, 528–536.

[28] Jiale Lao, Yibo Wang, Yufei Li, Jianping Wang, Yunjia Zhang, Zhiyuan Cheng, Wanghu Chen, Yuanchun Zhou, Mingjie Tang, and Jianguo Wang. 2024. A Demonstration of GPTuner: A GPT-Based Manual-Reading Database Tuning System. In *Companion of the 2024 International Conference on Management of Data (Santiago, Chile)(SIGMOD'24). Association for Computing Machinery, New York, NY, USA*, Vol. 4.

[29] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. 2017. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165* (2017).

[30] Guoliang Li, Xuanhe Zhou, Shifu Li, and Bo Gao. 2019. Qtune: A query-aware database tuning system with deep reinforcement learning. *Proceedings of the VLDB Endowment* 12, 12 (2019), 2118–2130.

[31] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2017. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The journal of machine learning research* 18, 1 (2017), 6765–6816.

[32] Yang Li, Yu Shen, Huaijun Jiang, Wentao Zhang, Jixiang Li, Ji Liu, Ce Zhang, and Bin Cui. 2022. Hyper-tune: Towards efficient hyper-parameter tuning at scale. *arXiv preprint arXiv:2201.06834* (2022).

[33] Yang Li, Yu Shen, Wentao Zhang, Ce Zhang, and Bin Cui. 2023. VolcanoML: speeding up end-to-end AutoML via scalable search space decomposition. *The VLDB Journal* 32, 2 (2023), 389–413.

[34] Jiesong Liu, Feng Zhang, Jiawei Guan, and Xipeng Shen. 2024. UQ-Guided Hyperparameter Optimization for Iterative Learners. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=k9uZfaeerK

[35] Anastasia Makarova, Huibin Shen, Valerio Perrone, Aaron Klein, Jean Baptiste Faddoul, Andreas Krause, Matthias Seeger, and Cedric Archambeau. 2022. Automatic termination for hyperparameter optimization. In *International Conference on Automated Machine Learning*. PMLR, 7–1.

[36] Xuhui Meng, Hessam Babaee, and George Em Karniadakis. 2021. Multi-fidelity Bayesian neural networks: Algorithms and applications. *J. Comput. Phys.* 438 (2021), 110361.

[37] Vu Nguyen, Sebastian Schulze, and Michael Osborne. 2020. Bayesian optimization for iterative learning. *Advances in Neural Information Processing Systems* 33 (2020), 9361–9371.

[38] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems* 32 (2019).

[39] Zhicheng Pan, Yihang Wang, Yingying Zhang, Sean Bin Yang, Yunyao Cheng, Peng Chen, Chenjuan Guo, Qingsong Wen, Xiduo Tian, Yunliang Dou, et al. 2023. Magicscaler: Uncertainty-aware, predictive autoscaling. *Proceedings of the VLDB Endowment* 16, 12 (2023), 3808–3821.

[40] Apostolos F Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. 2023. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *J. Comput. Phys.* 477 (2023), 111902.

[41] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB endowment. International conference on very large data bases*, Vol. 11. NIH Public Access, 269.

[42] Theodoros Rekatsinas, Xu Chu, Ihab F Ilyas, and Christopher Ré. 2017. Holoclean: Holistic data repairs with probabilistic inference. *arXiv preprint arXiv:1702.00820* (2017).

[43] Zeyuan Shang, Emanuel Zgraggen, Benedetto Buratti, Ferdinand Kossmann, Philipp Eichmann, Yeounoh Chung, Carsten Binnig, Eli Upfal, and Tim Kraska. 2019. Democratizing data science through interactive curation of ml pipelines. In *Proceedings of the 2019 international conference on management of data*. 1171–1188.

[44] Charles Truong, Laurent Oudre, and Nicolas Vayatis. 2020. Selective review of offline change point detection methods. *Signal Processing* 167 (2020), 107299.

[45] Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. 2021. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*. PMLR, 3–26.

[46] Chi Wang, Qingyun Wu, Markus Weimer, and Erkang Zhu. 2021. Flaml: A fast and lightweight automl library. *Proceedings of Machine Learning and Systems* 3 (2021), 434–447.

[47] Wei Wang, Sheng Wang, Jinyang Gao, Meihui Zhang, Gang Chen, Teck Khim Ng, and Beng Chin Ooi. 2018. Rafiki: Machine learning as an analytics service system. *arXiv preprint arXiv:1804.06087* (2018).

[48] Martin Wistuba, Arlind Kadra, and Josif Grabocka. 2022. Supervising the multi-fidelity race of hyperparameter configurations. *Advances in Neural Information Processing Systems* 35 (2022), 13470–13484.

[49] Robert F Woolson. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials* (2007), 1–3.

[50] Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. 2019. Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology* 17, 1 (2019), 26–40.

[51] Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *Frontiers of Computer Science* 18, 6 (2024), 186357.

[52] Shen Yan, Colin White, Yash Savani, and Frank Hutter. 2021. Nas-bench-x11 and the power of learning curves. *Advances in Neural Information Processing Systems* 34 (2021), 22534–22549.

[53] Li Yang and Abdallah Shami. 2020. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* 415 (2020), 295–316.

[54] Yuchen Yuan, Xiaoyue Feng, Bo Zhang, Pengyi Zhang, and Jie Song. 2024. JAPO: learning join and pushdown order for cloud-native join optimization. *Frontiers of Computer Science* 18, 6, Article 186614 (2024), 0 pages. https://doi.org/10.1007/s11704-024-3937-z

[55] Feng Zhang, Jidong Zhai, Xipeng Shen, Onur Mutlu, and Xiaoyong Du. 2021. POCLib: A high-performance framework for enabling near orthogonal processing on compression. *IEEE transactions on Parallel and Distributed Systems* 33, 2 (2021), 459–475.

[56] Xinyi Zhang, Zhuo Chang, Yang Li, Hong Wu, Jian Tan, Feifei Li, and Bin Cui. 2021. Facilitating database tuning with hyper-parameter optimization: a comprehensive experimental evaluation. *arXiv preprint arXiv:2110.12654* (2021).

[57] Xinyi Zhang, Hong Wu, Zhuo Chang, Shuowei Jin, Jian Tan, Feifei Li, Tieying Zhang, and Bin Cui. 2021. Restune: Resource oriented tuning boosted by meta-learning for cloud databases. In *Proceedings of the 2021 international conference on management of data*. 2102–2114.

[58] Xinyi Zhang, Hong Wu, Yang Li, Jian Tan, Feifei Li, and Bin Cui. 2022. Towards dynamic and safe configuration tuning for cloud databases. In *Proceedings of the 2022 International Conference on Management of Data*. 631–645.

[59] Xinlei Zhou, Han Liu, Farhad Pourpanah, Tieyong Zeng, and Xizhao Wang. 2022. A survey on epistemic (model) uncertainty in supervised learning: Recent advances and applications. *Neurocomputing* 489 (2022), 449–465.

[60] Lucas Zimmer, Marius Lindauer, and Frank Hutter. 2021. Auto-PyTorch Tabular: Multi-Fidelity MetaLearning for Efficient and Robust AutoDL. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 9 (2021), 3079 – 3090.