Importance-Aware Scheduling for High-Dimensional Hyperparameter Optimization

Ruinan Wang, Ian Nabney, Mohammad Golbabaee
University of Bristol, Bristol, United Kingdom
{zg21696, in17746, an22148}@bristol.ac.uk

***Abstract*—Hyperparameter Optimization (HPO) is essential for building high-performing ML/DL models, yet conventional optimizers often struggle in high-dimensional spaces where evaluations are costly and progress is diluted across many low-impact variables. We propose Greedy Importance First (GIF), an importance-aware scheduling strategy that uses a small-sample warm start to estimate hyperparameter importance, forms importance-based groups, allocates trials proportionally, and retains a full-space fallback. We evaluate GIF under fixed evaluation budgets on five anisotropic analytic functions ($d \in \{5, 10, 30, 50\}$), Bayesmark, and NAS-Bench-301 (33D). On the higher-dimensional benchmarks, GIF reaches better incumbents with faster convergence than TPE, BOHB, Random Search, and Sequential Grouping. On Bayesmark, where the effective dimensionality is smaller, GIF remains competitive but the margins are smaller. Ablation studies show that importance estimation, proportional allocation, and the fallback step all contribute to the gains. We also verify that the HIA component recovers the intended anisotropy on the analytic benchmarks. These results suggest that GIF is a simple and plug-compatible way to improve sample efficiency in high-dimensional HPO.**

***Index Terms*—Hyperparameter Optimization, Hyperparameter Importance, Bayesian Optimization, AutoML, Deep Learning**

I. Introduction

Hyperparameter optimization (HPO) is a critical stage in modern ML/DL pipelines: it governs robustness, stability, and generalization. Despite a mature toolbox—Bayesian optimization (e.g., TPE [1], BOHB [2]), evolutionary [3], and bandit methods [4]—efficiency often degrades as dimensionality grows: each evaluation becomes costlier and surrogates become harder to fit and less informative [5]. Crucially, the obstacle is not dimensionality alone but the strongly uneven influence of hyperparameters [6]. In many models, a small subset of settings accounts for most performance variation, while others contribute marginally. Yet most optimizers advance all coordinates in lockstep each iteration, effectively enforcing uniform scheduling. This induces a dimensionality bottleneck: treating all hyperparameters equally dilutes the budget and delays progress, especially under tight evaluation limits.

Hyperparameter importance assessment (HIA) provides a principled foundation for addressing this bottleneck: from a small set of trials, it estimates each hyperparameter's marginal contribution to performance—and, when needed, pairwise interactions. However, despite the availability of estimators such as N-RReliefF [7], fANOVA [8], and PED-ANOVA [9], there is no widely adopted strategy that operationalizes these estimates into concrete scheduling decisions. As a result, HIA methods are underutilized in practice.

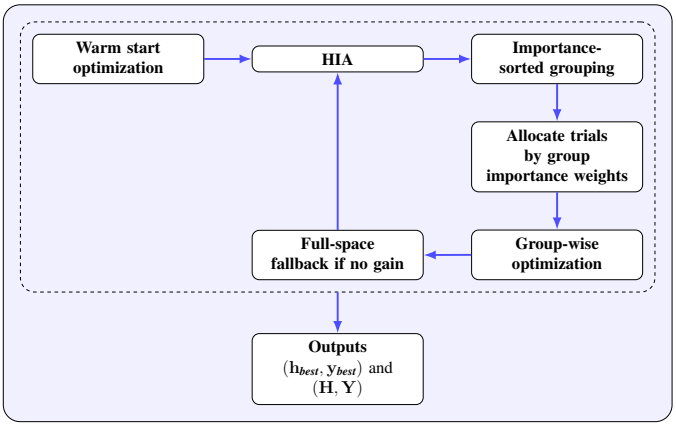

Fig. 1. **GIF Pipeline:** High-level workflow of the proposed Greedy Importance First strategy.

This paper introduces Greedy Importance-First (GIF), an importance-aware HPO strategy that turns HIA insights into an explicit, budgeted search plan. As illustrated in Fig. 1, GIF (i) performs a small-sample warm start to collect an initial history for HIA HIA algorithms; (ii) orders hyperparameters by estimated importance and groups them accordingly; (iii) allocates budgets proportionally to group importance and optimizes each group while fixing other variables at the current incumbent, warm-starting from the accumulated history; and (iv) when a round yields no improvement, falls back to joint optimization to restore global exploration. This design concentrates the budget where it matters most, while the fallback to joint optimization provides a principled escape from local stagnation. We evaluate GIF under fixed budgets on controlled anisotropic analytic functions, Bayesmark tasks [10], and NAS-Bench-301 [11]. Ablations disentangle the effect of each component, and we further verify that HIA can recover the ground-truth anisotropy on the analytic benchmarks—even with limited evaluations, it correctly highlights the few dominant coordinates while suppressing negligible ones.

Contributions:

- We propose Greedy Importance First (GIF), which converts HIA estimates into a concrete search plan. Specifically, GIF orders hyperparameters by estimated importance, partitions them into groups, allocates trials propor-

tionally, and introduces a per-round full-space fallback. It provides a simple pathway to more economical high-dimensional HPO.
- We validate that standard HIA estimators (e.g., N-RReliefF) recover ground-truth anisotropy on controlled functions, supporting their use as reliable priors when budgets are tight.
- Under fixed evaluation budgets and multiple random seeds, GIF consistently outperforms various established HPO baselines on higher-dimensional benchmarks (five anisotropic analytic functions, NAS-Bench-301), while remaining competitive on lower/mid-dimensional Bayesmark tasks (4 models $\times$ 5 datasets). In high-dimensional scenarios, GIF achieves a markedly better accuracy–time trade-off.
- Ablation studies show that the introduced components—(i) importance-driven ranking, (ii) proportional budget allocation, and (iii) the full-space joint fallback—each contributes to the overall gains; removing any single component leads to a clear drop in performance.

II. Related Work

Hyperparameter Importance Assessment (HIA). Understanding which hyperparameters "matter" has long supported post-hoc analysis and space design; for example, Weights & Biases (W&B) Sweeps provide importance plots from trial histories [12], while libraries such as Optuna and SMAC3 expose fANOVA-based importance tools [13], [14]. Methodologically, fANOVA remains a standard variance-decomposition approach [8]; PED-ANOVA generalizes it with a Pearson-divergence–based closed form that enables efficient local importance on arbitrary subspaces (e.g., top-performing regions) [9]. Complementary to fANOVA-style decompositions, N-RReliefF adapts ReliefF to continuous responses and quantifies both marginal and pairwise interaction importance from HPO histories, offering a lightweight, data-driven estimator under tight budgets [7].

Gray-box and uncertainty-aware HPO. Gray-box approaches enrich BO surrogates with intermediate training signals (e.g., learning curves, checkpoint features, or partial-fidelity measurements), and uncertainty-aware schedulers couple candidate selection with budget allocation to avoid premature discarding under early-stage noise [2], [15], [16]. While these methods exploit richer signals across candidates, GIF serves as a lightweight allocator across hyperparameters: it uses HIA from small warm-starts to reweight search effort across hyperparameters and can plug into TPE-style optimizers as the inner engine.

Resource allocation, warm starts, and scheduling. Many HPO systems exploit warm starts (e.g., transferring priors or surrogate states), parallel scheduling, or multi-fidelity allocation across candidates and tasks [2], [4], [17]–[19]. However, they typically retain uniform treatment across hyperparameters within an iteration. GIF breaks this per-iteration uniformity by fixing non-targeted hyperparameters to the current incumbent and concentrating trials on the most important groups. This increases the signal-to-noise ratio per evaluation in high-dimensional regimes. A per-round full-space fallback then provides a principled escape hatch from local plateaus.

In sum, GIF turns early HIA into a concrete search plan: importance-ordered grouping, importance-proportional allocation, warm-started subspace search, and a safeguarded full-space fallback. This yields a plug-compatible route to sample-efficient HPO in high-dimensional settings, complementary to structural high-dimensional BO (subspace/variable-selection assumptions and local/trust-region BO), gray-box surrogates, and uncertainty-aware schedulers.

III. Problem Setup

We consider HPO on a fixed dataset $\mathcal{D}$ and hyperparameter search space $\Theta = \Theta_1 \times \cdots \times \Theta_d$, where each Θ_i is the domain of hyperparameter H_i. A configuration is $\mathbf{h} = (h_1, \ldots, h_d) \in \Theta$. The black-box objective is $f_{\mathcal{D}} : \Theta \to \mathbb{R}, \mathbf{h} \mapsto f_{\mathcal{D}}(\mathbf{h})$, which returns a scalar performance (e.g., validation accuracy to maximize). The goal of HPO is $\mathbf{h}^\star \in \arg\max_{\mathbf{h}\in\Theta} f_{\mathcal{D}}(\mathbf{h}), y^\star = f_{\mathcal{D}}(\mathbf{h}^\star)$, subject to a limited evaluation (or wall-clock) budget B_{total}. After t evaluations, the history is $\mathcal{H} = \{\mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(t)}\}$ and $\mathcal{Y} = \{y^{(1)}, \ldots, y^{(t)}\}$ with $y^{(i)} = f_{\mathcal{D}}(\mathbf{h}^{(i)})$. The incumbent (best-so-far) configuration is $(h_{\text{best}}, y_{\text{best}})$ where $y_{\text{best}} = \max_{i\le t} y^{(i)}$. An optimizer $\mathcal{A}_{\text{opt}}$ proposes new candidates conditioned on $(\mathcal{H}, \mathcal{Y})$, evaluates them, and appends results until B_{total} is exhausted. Standard outputs are the final incumbent $(h_{\text{best}}, y_{\text{best}})$ and the complete trace $(\mathcal{H}, \mathcal{Y})$.

Representative baseline. TPE [1] partitions the history $(\mathcal{H}, \mathcal{Y})$ by a score threshold y_0 (e.g., the γ-quantile), and fits conditional densities $l(\mathbf{h}) = p(\mathbf{h} \mid y \ge y_0)$ and $g(\mathbf{h}) = p(\mathbf{h} \mid y < y_0)$. New candidates maximize $l(\mathbf{h})/g(\mathbf{h})$, a proxy for expected improvement. In practice, l and g are estimated via Parzen windows with the factorization $p(\mathbf{h}) \approx \prod_{j=1}^{d} p(h_j)$. The iterative loop is: fit densities $\to$ sample $\mathbf{h}^{(t+1)} \to$ evaluate $f_{\mathcal{D}}(\mathbf{h}^{(t+1)}) \to$ update the history.

Typical bottlenecks in High Dimensions. For the representative optimizer TPE, as the dimensionality d of the search space increases, several limitations arise under tight evaluation budgets B_{total}: **(i)** The independence assumption $p(\mathbf{h}) \approx \prod_j p(h_j)$ neglects coordinate interactions. In high-dimensional hyperparameter spaces, many variables only matter through their joint effects. Ignoring such dependencies causes both $l(\mathbf{h})$ and $g(\mathbf{h})$ to appear nearly uniform across most coordinates, offering little guidance for exploration. **(ii)** In higher dimensions, density estimation becomes increasingly noisy because the effective sample size per coordinate shrinks. With limited evaluations, each marginal distribution is poorly supported, so $l(\mathbf{h})$ and $g(\mathbf{h})$ fluctuate heavily, yielding unstable search guidance. **(iii)** As d increases, contributions of coordinates to f are highly imbalanced; the presence of many low-impact dimensions reduces the effective signal-to-noise in each sampled evaluation, leading to slower improvement over iterations. These effects help explain why standard BO methods such as TPE often struggle in high-dimensional settings. GIF addresses this issue by using hyperparameter

Algorithm 1 GIF Main Strategy

Require: Search space Θ, objective $f_{\mathcal{D}}$, subsample ratio α, initial budget B_{init}, step size B_{step}, total budget B_{total}, max group size k, importance evaluator $\mathcal{A}_{\text{imp}}$, optimizer $\mathcal{A}_{\text{opt}}$, fallback ratio ρ

Ensure: Incumbent $(\mathbf{h}_{\text{best}}, y_{\text{best}})$, complete evaluation trace $(\mathcal{H}, \mathcal{Y})$

$(\mathcal{H}, \mathcal{Y}), (\mathbf{h}_{\text{best}}, y_{\text{best}}) \leftarrow$ **WarmStart**$(\Theta, f_{\mathcal{D}}, \alpha, B_{\text{init}}, \mathcal{A}_{\text{opt}})$

$T_{\text{used}} \leftarrow B_{\text{init}}, \quad T_{\text{full used}} \leftarrow 0, \quad B_{\text{full total}} \leftarrow \rho\, B_{\text{total}}$

while $T_{\text{used}} < B_{\text{total}}$ **do**

  $I \leftarrow \mathcal{A}_{\text{imp}}(\mathcal{H}, \mathcal{Y})$ ▷ importance weights $\{I_i\}_{i=1}^{d}$

  FormGroups: sort indices by I (desc.), then partition into groups $\mathcal{G} = \{\mathcal{G}_j\}$ with $|\mathcal{G}_j| \le k$

  $B_{\text{cur}} \leftarrow \min\big(B_{\text{step}},\, B_{\text{total}} - T_{\text{used}}\big)$

  $\mathbf{b} \leftarrow$ **AllocateBudget**$(\mathcal{G}, I, B_{\text{cur}})$

  $(\mathcal{H}, \mathcal{Y}, \mathbf{h}_{\text{best}}, y_{\text{best}}, T_{\text{used}}, \mathit{improved}) \leftarrow$ **GroupOpt**$(\mathcal{G}, \mathbf{b}, \mathcal{H}, \mathcal{Y}, \mathbf{h}_{\text{best}}, y_{\text{best}}, T_{\text{used}}, \mathcal{A}_{\text{opt}})$

  if not $\mathit{improved}$ **and** $T_{\text{full used}} < B_{\text{full total}}$ **and** $T_{\text{used}} < B_{\text{total}}$ **then**

    $R \leftarrow \left\lfloor \frac{B_{\text{total}} - T_{\text{used}}}{B_{\text{step}}} \right\rfloor + 1$

    $B_{\text{full}} \leftarrow \min\Big(\big\lfloor \frac{B_{\text{full total}} - T_{\text{full used}}}{R} \big\rfloor,\, B_{\text{total}} - T_{\text{used}}\Big)$

    $(\mathcal{H}, \mathcal{Y}, \mathbf{h}_{\text{best}}, y_{\text{best}}, T_{\text{used}}, T_{\text{full used}}) \leftarrow$ **FullSpaceOpt**$(\Theta, f_{\mathcal{D}}, B_{\text{full}}, \mathcal{H}, \mathcal{Y}, \mathbf{h}_{\text{best}}, y_{\text{best}}, T_{\text{used}}, T_{\text{full used}}, \mathcal{A}_{\text{opt}})$

  end if

end while

return $(\mathbf{h}_{\text{best}}, y_{\text{best}})$ and $(\mathcal{H}, \mathcal{Y})$

importance estimates to order variables, form groups, and allocate evaluations more selectively.

IV. The GIF Algorithm

A. Pipeline Overview

Algorithm 1 formalizes how the key components of GIF are orchestrated into a single scheduling strategy.

B. Warm Start

Inputs: Search space Θ, dataset $\mathcal{D}$ (size $|\mathcal{D}|$), objective $f_{\mathcal{D}} : \Theta \to \mathbb{R}$, subsample ratio $\alpha \in (0, 1]$, warm-start budget B_{init}, inner optimizer $\mathcal{A}_{\text{opt}}$. **Outputs:** Initial history $(\mathcal{H}, \mathcal{Y})$ with $|\mathcal{H}| = |\mathcal{Y}| = B_{\text{init}}$, and incumbent $(\mathbf{h}_{\text{best}}, y_{\text{best}})$. We randomly subsample the dataset to obtain $\mathcal{D}_{\text{init}}$ of size $\alpha|\mathcal{D}|$ and run $\mathcal{A}_{\text{opt}}$ for B_{init} evaluations on Θ (using $\mathcal{D}_{\text{init}}$), producing $(\mathcal{H}, \mathcal{Y})$ and initializing $(\mathbf{h}_{\text{best}}, y_{\text{best}})$ as the best in this history. The warm-start stage reduces early evaluation cost while providing a more informative history for subsequent importance estimation than purely random initialization.

C. Hyperparameter Importance Assessment (HIA)

Inputs: Optimization history $(\mathcal{H}, \mathcal{Y})$; evaluator $\mathcal{A}_{\text{imp}}$. **Outputs:** Normalized importance profile $\{I_i\}_{i=1}^{d}$ assigning a non-negative weight to each hyperparameter H_i.

In general, HIA methods assign weights $I_1, \ldots, I_d$ estimating each hyperparameter's marginal contribution to performance, providing interpretable insights about "what matters" and informing downstream scheduling or search-space design. Representative techniques include *fANOVA* [8], *PED-ANOVA* [9], and *N-RReliefF* [7]. In GIF, we employ *N-RReliefF* as our default importance evaluator. Given history $(\mathcal{H}, \mathcal{Y})$, N-RReliefF treats each configuration as a reference point, compares it with its nearest neighbors in configuration space, and accumulates per-dimension covariation weighted by the performance difference between neighbors. This produces raw scores $\widehat{I}_i$, which are then mapped into positive, comparable importances via a softplus normalization and rescaled so that $\sum_i I_i = 1$. In this way, dimensions where small input changes consistently lead to large performance shifts are assigned higher weights, which in turn underpin ordering and grouping in GIF.

D. Grouping and Allocation

Key Inputs: Importance weights $\{I_i\}_{i=1}^{d}$; maximum group size k; per-round step size B_{step}; total budget B_{total}; used trials T_{used}. **Outputs:** A partition of hyperparameter indices into groups $\mathcal{G} = \{\mathcal{G}_j\}$ with $|\mathcal{G}_j| \le k$, and per-group trials allocations $\mathbf{b} = [b_1, \ldots, b_{|\mathcal{G}|}]$. We first set the current round budget $B_{\text{cur}} = \min\big(B_{\text{step}},\, B_{\text{total}} - T_{\text{used}}\big)$. Based on $\{I_i\}$, we sort hyperparameters by descending weight and partition them into groups of size at most k. For each group $\mathcal{G}_j$, we compute its total weight $I_j = \sum_{i \in \mathcal{G}_j} I_i$ and allocate trials proportionally: $b_j = \max\Big(1,\, \Big\lfloor \frac{I_j}{\sum_g I_g}\, B_{\text{cur}} \Big\rfloor\Big)$. Enforcing $b_j \ge 1$ guarantees at least one trial per group; the final allocations $\mathbf{b}$ are then passed to the group-wise optimization stage. If rounding leaves unassigned trials, we distribute the remaining trials one by one to the groups with the largest fractional remainders, ensuring that the final integer allocations exactly match the per-round budget.

E. Group-wise Optimization

Key Inputs: Groups $\mathcal{G} = \{\mathcal{G}_j\}$, per-group allocations $\mathbf{b} = [b_1, \ldots, b_{|\mathcal{G}|}]$, current history $(\mathcal{H}, \mathcal{Y})$, incumbent $(\mathbf{h}_{\text{best}}, y_{\text{best}})$, and inner optimizer $\mathcal{A}_{\text{opt}}$. **Outputs:** Updated history $(\mathcal{H}, \mathcal{Y})$, updated incumbent $(\mathbf{h}_{\text{best}}, y_{\text{best}})$, and updated trial counter T_{used}.

For each group $\mathcal{G}_j$, we fix all hyperparameters outside $\mathcal{G}_j$ to their values in the current incumbent $\mathbf{h}_{\text{best}}$. We then invoke the inner optimizer $\mathcal{A}_{\text{opt}}$ for b_j evaluations restricted to $\mathcal{G}_j$, with `warm-start` from the existing history $(\mathcal{H}, \mathcal{Y})$. The resulting evaluations $(\mathcal{H}_j, \mathcal{Y}_j)$ are appended to the history, and T_{used} is incremented by b_j. After each group is optimized, we update the incumbent if a better configuration is discovered. If all groups fail to improve the incumbent, the round is considered *unsuccessful*, potentially triggering the full-space fallback. Otherwise, the algorithm proceeds with the next round using the updated history and incumbent.

F. Full-Space Fallback

Key Inputs: Remaining trials $T_{\text{left}} = B_{\text{total}} - T_{\text{used}}$; full-space reserved quota $B_{\text{full total}} = \rho\, B_{\text{total}}$; cumulative full-space trials used $T_{\text{full used}}$ (i.e., trials already spent on full-space fallback); step size B_{step}. **Outputs:** updated evaluation history $(\mathcal{H}, \mathcal{Y})$ and updated incumbent $(\mathbf{h}_{\text{best}}, y_{\text{best}})$. To guard against subspace stagnation while balancing exploration–exploitation, GIF triggers a full-space step *only* when an entire group-wise round yields no improvement. Given T_{left}, define the remaining full-space quota $T_{\text{full left}} = \max\big(0,\ B_{\text{full total}} - T_{\text{full used}}\big)$, and the estimated number of future rounds $n_{\text{round}} = \left\lfloor \frac{T_{\text{left}}}{B_{\text{step}}} \right\rfloor + 1$. Allocate a per-round fallback budget $B_{\text{full}} = \min\Big(\lfloor T_{\text{full left}}/n_{\text{round}} \rfloor,\ T_{\text{left}} \Big)$. Run the inner optimizer on the full space Θ for B_{full} evaluations with warm-start $(\mathcal{H}, \mathcal{Y})$, obtain $(\mathcal{H}_{\text{full}}, \mathcal{Y}_{\text{full}})$, and update $(\mathbf{h}_{\text{best}}, y_{\text{best}})$, $T_{\text{used}} \leftarrow T_{\text{used}} + B_{\text{full}}$, $T_{\text{full used}} \leftarrow T_{\text{full used}} + B_{\text{full}}$. If group-wise optimization keeps improving, the fallback is never activated; the algorithm continues with the standard per-round budget until $T_{\text{used}} = B_{\text{total}}$, and any unused full-space quota remains unspent.

Implementation Note The inner routine $\mathcal{A}_{\text{opt}}$ can be any standard HPO method (e.g., TPE, BOHB) that supports warm starts. All calls reuse the cumulative history $(\mathcal{H}, \mathcal{Y})$, enabling consistent importance estimation and avoiding redundant random initialization. In our experiments, we focus on the scheduling strategy itself and therefore adopt TPE as the default $\mathcal{A}_{\text{opt}}$ unless otherwise specified.

V. Experiments

We evaluate GIF on three types of benchmarks: (1) anisotropic analytic functions designed to stress high-dimensional search, (2) Bayesmark tabular tasks, and (3) NAS-Bench-301 (33D) for neural architecture optimization. Unless noted otherwise, each run uses a total budget of 500 evaluations over 5 independent seeds, with the warm-start budget B_{init}=100 counted as part of the total. The inner optimizer $\mathcal{A}_{\text{opt}}$ is TPE [13], and the importance estimator $\mathcal{A}_{\text{imp}}$ is N-RReliefF [7]. For GIF, we use the following default configuration across benchmarks: subsample ratio α=0.6, per-round step size B_{step}=d, maximum group size k=$\lfloor d/3 \rfloor$, and fallback ratio ρ=0.2.

A. Anisotropic Analytic Function Benchmarks

We selected five classic black-box optimization functions widely used in HPO benchmarking: Sphere, Rosenbrock, Ackley [20], Griewank [21], and Rastrigin [22]. Each function was instantiated at dimensions $d \in \{5, 10, 30, 50\}$.

Anisotropic Variable Transformation. To induce anisotropy, we applied a diagonal scaling $\mathbf{w} = (w_1, \ldots, w_d)$ with $w_i = \exp\big(-\alpha\,(i-1)\big), \alpha = \frac{-\log(10^{-3})}{d-1}$, so that $w_d/w_1 = \exp\big(-\alpha(d-1)\big) = 10^{-3}$. This stylized construction creates a known, non-uniform sensitivity profile across coordinates. It is intended to provide a clean and controlled testbed for evaluating importance-aware schedulers under heterogeneous influence.

TABLE I
Weighted analytic benchmark functions with anisotropic scaling.

Function	Formula
Anisotropic Sphere	$f(\mathbf{x}) = -\sum_{i=1}^{d} (w_i x_i)^2$
Anisotropic Rosenbrock	$f(\mathbf{x}) = -\sum_{i=1}^{d-1} \Big[100\big(w_{i+1}x_{i+1} - (w_i x_i)^2\big)^2 + (1 - w_i x_i)^2\Big]$
Anisotropic Ackley	$f(\mathbf{x}) = -\Big(-20\,\exp\big(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}(w_i x_i)^2}\big) - \exp\big(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi w_i x_i)\big) + 20 + \mathrm{e}\Big)$
Anisotropic Griewank	$f(\mathbf{x}) = -\Big(1 + \frac{1}{4000}\sum_{i=1}^{d}(w_i x_i)^2 - \prod_{i=1}^{d}\cos\big(\frac{w_i x_i}{\sqrt{i}}\big)\Big)$
Anisotropic Rastrigin	$f(\mathbf{x}) = -\sum_{i=1}^{d}\Big[(w_i x_i)^2 - 10\cos(2\pi w_i x_i) + 10\Big]$

In Table I, the domains are $[-5, 5]^d$ for Sphere, Rosenbrock, Ackley, and Griewank, and $[-5.12, 5.12]^d$ for Rastrigin. We negate the standard minimization forms to adopt a maximization convention. For Sphere, Ackley, Griewank, and Rastrigin, the global maximizer is $\mathbf{x} = \mathbf{0}$ with maximum 0. For Rosenbrock, the *unconstrained* maximizer under our scaling satisfies $w_i x_i = 1$ for all i, yielding value 0.

Baselines. We compared GIF against Sequential Grouping (SG) [19], Bayesian Optimization based on Tree-structured Parzen Estimator (TPE), Bayesian Optimization based on Gaussian Process (GP), Bayesian Optimization with Hyperband (BOHB) [2], and Random Search. All competitors used the identical box domains in Table I, the same total evaluation budget (500) and seeds (5), and — where appropriate — the same warm-start history.

Verification of Importance Estimation. We verified that N-RReliefF could serve as an importance analyzer by testing whether it recovered the coordinate-wise anisotropy of each benchmark function. For every function and each $d \in \{5, 10, 30, 50\}$, we drew 500 i.i.d. samples $\mathbf{x} \sim \mathcal{U}([-1, 1]^d)$, evaluated $y = f(\mathbf{x})$, estimated per-coordinate importances $\{I_i\}$, and compared them with the ground-truth weights $\{w_i\}$ after max-normalization. Recovery was quantified by the Pearson correlation between $\{w_i\}$ and $\{I_i\}$.

B. Ablation Studies

To isolate the contribution of each design component in GIF, we conducted ablations on the same anisotropic analytic benchmarks as Table I, using the identical protocol and budgets as in the previous subsection. **Variant A — Randomized Importance (`RandImp`):** We replaced the importance evaluator with random per–coordinate weights to test whether gains arose from meaningful importance estimation rather than staged optimization alone; **Variant B — Uniform Allocation (`UniAlloc`):** We retained true importances for grouping but allocated an equal number of trials to each group (no importance weighting) to probe the necessity of importance-

weighted budgeting; **Variant C — No Full-Space Fallback (`NoFB`):** We disabled the joint full-space optimization step to evaluate the fallback's role in escaping local plateaus and maintaining robustness.

For all the experiments on the anisotropic analytic benchmarks, we aggregated across all five functions and $d \in \{5, 10, 30, 50\}$, and reported: (i) normalized regret AUC (Table III, the detailed description of the metric is in Sec. VI-A); and (ii) final best values at 500 trials summarized in a per-function heatmap (mean $\pm$ std across 5 seeds; Fig. 3);

C. Bayesmark

Bayesmark is an open-source benchmark for comparing Bayesian optimization methods via a unified API, standardized search spaces, and consistent evaluation [10]. We ran the official Bayesmark benchmark on four datasets (`breast`, `digits`, `iris`, `wine`) using Bayesmark's default configurations and search spaces for four models—Decision Tree (`DT`, d=6), Random Forest (`RF`, d=6), Multi-Layer Perceptron trained with the Adam (`MLP-adam`, d=9), and Multi-Layer Perceptron trained with stochastic gradient descent (`MLP-sgd`, d=8). For consistency across tasks, we used a single primary metric per task type: accuracy for classification and mean squared error (MSE) for regression. Train/validation splits and hyperparameter ranges followed Bayesmark defaults for all optimizers. We summarized performance via task-wise normalized final best scores (Perf. Norm), Avg. Rank, Time Rank, and Win Rate in Table IV.

Baselines. We included Bayesmark's default optimizers: *HyperOpt* (HOpt; TPE-based Bayesian optimization), *OpenTuner-BanditA* (OT-B; bandit-coordinated portfolio search), *OpenTuner-GA* (OT-GA; genetic algorithm), *OpenTuner-GA-DE* (OT-GD; GA + differential evolution hybrid), *PySOT* (PySOT; surrogate-based global optimization), *RandomSearch* (RS; uniform random sampling), *Scikit-GBRT-Hedge* (GBRT; GBRT surrogate with Hedge acquisition mixing), *Scikit-GP-Hedge* (GP-H; GP surrogate with Hedge acquisition mixing), and *Scikit-GP-LCB* (GP-LCB; GP surrogate with LCB acquisition) [1], [10], [23]–[25]. To ensure fairness, all baselines and *GIF* ran with identical search spaces, budgets, seeds, and splits; when applicable, we reused the same warm-start history.

D. NAS-Bench-301

Unlike the fully tabular NAS benchmarks 101 [26] and 201 [27], NAS-Bench-301 (NB301) [11] is a surrogate benchmark that emulates the Differentiable Architecture Search (DARTS) [28] search space and yields fast, approximate evaluations in a realistic high-dimensional regime. Concretely, NB301 is built on the DARTS cell space trained on CIFAR-10 and provides learned regressors that map an architecture encoding to predicted validation accuracy (and a separate regressor for runtime), enabling faithful anytime comparisons without re-training each architecture. In this work, we used the official `SNB-DARTS-XGB-v1.0` release [11]: an XGBoost-based surrogate trained on DARTS+CIFAR-10 with stratified train/val/test splits over data gathered from multiple NAS optimizers. We kept the benchmark's 33-dimensional architecture encoding and queried the surrogate-predicted validation accuracy as the objective; for wall-clock plots we used the benchmark's runtime surrogate to accumulate simulated time. We evaluated GIF against TPE, BOHB, Random, and SG on `darts-xgb-v1.0`. We reported: (i) best validation score vs. evaluations; (ii) best validation score vs. simulated wall-clock time; and (iii) a Pareto view (score vs. time) that summarizes the quality–time trade-off (Fig. 4).

TABLE II
PEARSON CORRELATION BETWEEN GROUND-TRUTH WEIGHTS w_i AND HIA SCORE ESTIMATES I_i.

	$d = 5$	$d = 10$	$d = 30$	$d = 50$
Weighted Ackley	0.995	0.986	0.959	0.941
Weighted Griewank	0.985	0.896	0.670	0.547
Weighted Rastrigin	0.990	0.854	0.696	0.717
Weighted Rosenbrock	0.993	0.982	0.831	0.791
Weighted Sphere	0.987	0.917	0.819	0.805
Mean $\pm$ Std	0.990 ± 0.004	0.927 ± 0.057	0.795 ± 0.116	0.760 ± 0.144

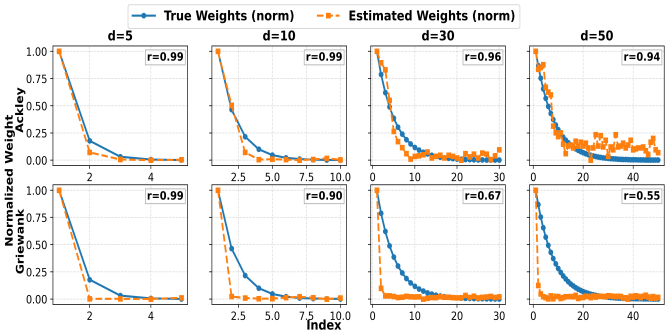

Fig. 2. Anisotropy verification on weighted Ackley and Griewank: normalized ground-truth weights vs. estimated weights across dimensions ($d \in \{5, 10, 30, 50\}$), with Pearson correlation r shown in each subplot.

VI. RESULTS

A. Verification of Importance Estimation

Before applying GIF to real HPO tasks, we first test whether the importance estimator (N-RReliefF) can recover the intended anisotropy on the analytic benchmarks. Experimental details are given in Sec. V-A. Table II reports the Pearson correlation between the estimated scores $\{I_i\}$ and the ground-truth weights $\{w_i\}$ for $d \in \{5, 10, 30, 50\}$. As dimension increases under a fixed sampling budget, the estimates become noisier and the correlations decrease. Figure 2 shows this trend for two representative functions. On weighted Ackley, the estimated importance profile follows the true decay closely across all dimensions, with only mild degradation as d grows. Weighted Griewank is noticeably harder: the match deteriorates more quickly, especially in higher dimensions. This is consistent with its oscillatory cosine-product structure, which introduces stronger interactions and makes marginal importance harder to estimate from limited samples. Overall, the results show that the intended anisotropy can be recovered

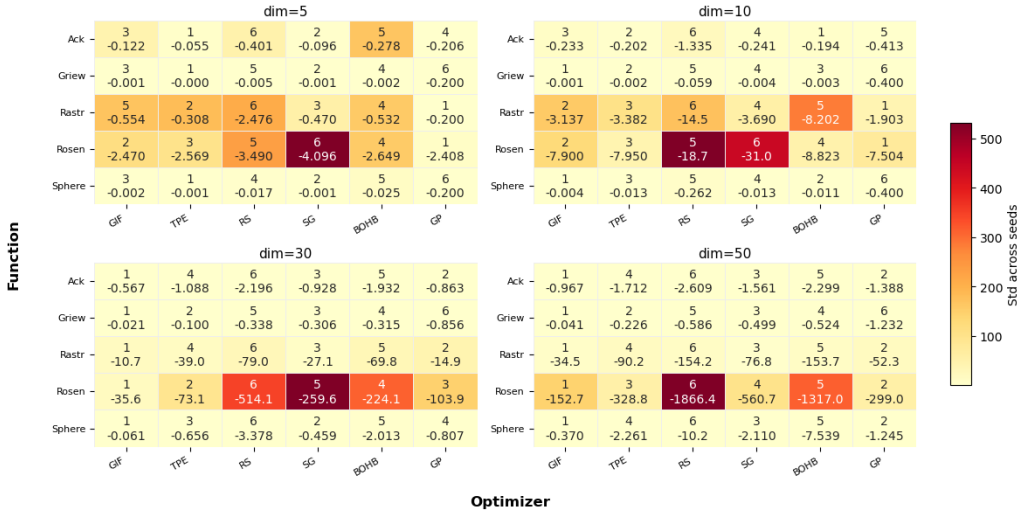

Fig. 3. Performance summary of GIF and baselines on weighted analytic benchmarks.

TABLE III
NORMALIZED REGRET AUC (LOWER IS BETTER) FOR ANISOTROPIC ANALYTIC BENCHMARKS. TOP: BASELINES VS. GIF. BOTTOM: ABLATIONS. GIF-WIN = FRACTION OF SEEDS WITH THE BEST AUC.

Dim	BOHB	GP	RS	SG	TPE	GIF-win
5	4.90 ± 2.07	10.40 ± 3.27	7.01 ± 2.55	9.57 ± 1.36	2.71 ± 0.64	20%
10	11.36 ± 3.16	20.35 ± 6.23	21.41 ± 2.72	25.36 ± 5.01	8.06 ± 2.00	60%
30	35.09 ± 3.42	30.89 ± 4.52	39.75 ± 6.01	40.14 ± 5.42	21.34 ± 2.84	100%
50	44.23 ± 3.57	42.88 ± 2.26	48.36 ± 4.28	47.08 ± 6.15	29.35 ± 2.54	100%

Dim	GIF	RandImp	UniAlloc	NoFB	GIF-win
5	3.20 ± 0.81	3.24 ± 2.65	3.28 ± 2.63	3.26 ± 2.68	0%
10	**7.36 ± 0.96**	6.96 ± 2.55	7.95 ± 2.68	7.19 ± 2.51	60%
30	**14.75 ± 2.55**	18.46 ± 3.98	21.01 ± 5.71	17.62 ± 4.39	100%
50	**19.18 ± 1.56**	24.07 ± 3.07	24.70 ± 2.97	23.33 ± 2.54	100%

TABLE IV
BAYESMARK SUMMARY. OPTIMIZER ABBREVIATIONS ARE DEFINED IN SEC. V-C.

Opt.	Avg. Rank ↓	Time Rank ↓	Perf. Norm ↑	Win Rate ↑
GIF	**2.72**	5.13	**0.811**	**0.750**
RS	6.28	**1.66**	0.189	0.063
HOpt	4.56	5.84	0.354	0.094
PySOT	4.47	6.03	0.371	0.156
GP-H	3.47	10.0	0.401	0.125
OT-B	5.28	3.78	0.274	0.094
GBRT	4.78	7.94	0.314	0.125
OT-GD	5.50	2.94	0.234	0.063
GP-LCB	4.09	9.00	0.370	0.219
OT-GA	5.75	2.69	0.220	0.063

reliably in low and moderate dimensions, and that even in harder high-dimensional cases the estimator still captures the broad decay pattern.

B. Analytic Benchmarks and Ablations

We evaluate GIF on five anisotropic analytic functions against five baselines and three ablation variants. The setup is described in Sec. V-A and Sec. V-B. Figure 3 shows that at low dimension (d=5), GIF does not consistently dominate strong baselines such as TPE. As dimension increases, however, GIF becomes more reliable across functions, while several baselines degrade or become unstable.

Table III reports normalized regret AUC (lower is better), following [29]. For a maximization objective $f(\cdot)$, the regret at trial t is $r_t = f^\star - \max_{s\le t} f\big(h^{(s)}\big)$, where $f^\star$ is the known optimum. We summarize the trajectory over T trials by Regret-AUC $= \frac{1}{r_0 T}\int_0^T r_t\,dt$, where r_0 is computed from a shared initialization so that results from different functions are on a comparable scale.

At d=5, TPE achieves the best mean score, while GIF is slightly worse and wins only 20% of seeds. This is not surprising: in small spaces, strong baselines can already model the landscape well, so the benefit of importance-aware scheduling is limited. From d=10 onward, GIF consistently performs better than all baselines. Its win rate also increases sharply with dimension, suggesting that the gains are not only larger on average but also more stable across seeds.

The ablation results show that each component matters. Replacing learned importance with random weights (`RandImp`) or removing proportional allocation (`UniAlloc`) leads to worse AUC, indicating that the importance signal itself is useful. Removing the fallback step (`NoFB`) is especially harmful in higher dimensions, where periodic global exploration helps recover from poor subspace choices.

C. Bayesmark (Mid-Dimensional Evaluation)

Having verified the importance estimator and tested GIF on controlled analytic functions, we next evaluate it on Bayesmark. Table IV summarizes performance across tasks using four metrics: mean normalized final score (Perf. Norm), average rank by performance, average rank by runtime, and win rate.

GIF is the strongest method overall in this benchmark: it achieves the best Perf. Norm, the best average rank, and

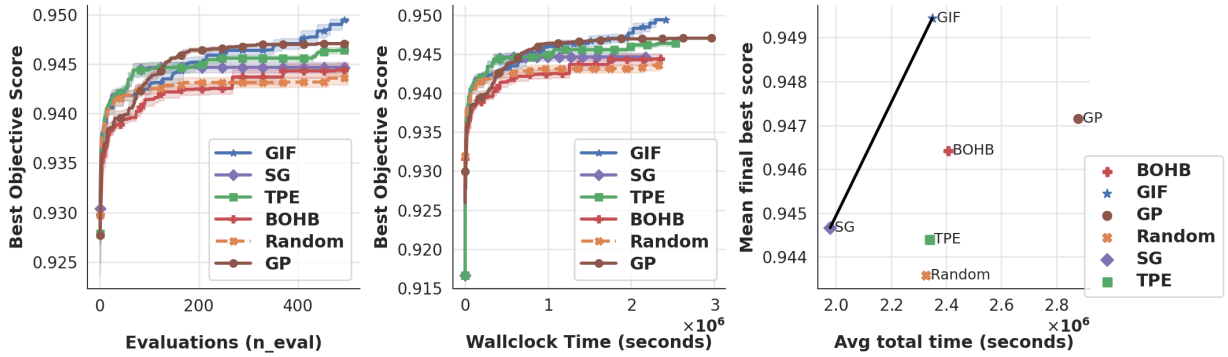

Fig. 4. Convergence and Pareto analysis on NAS-Bench-301 (DARTS-XGB surrogate, 33D)

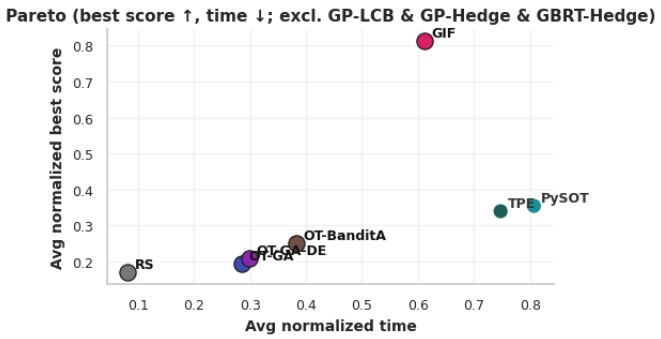

Fig. 5. Pareto trade-off between final score and time (lightweight methods).

the highest win rate. Random Search remains attractive when runtime is the main concern, but its final performance is much weaker. Other surrogate-based methods, such as HOpt and PySOT, can perform well on some tasks, but they are less consistent across the full benchmark.

This result is in line with the role of GIF. Bayesmark contains tasks with different datasets, models, and response surfaces, so performance on one subset of tasks does not necessarily transfer to the others. GIF is not the fastest method because importance estimation, grouping, and fallback steps introduce extra overhead. Still, that extra coordination improves how evaluations are spent, which leads to better final solutions overall. The Pareto plot in Fig. 5 reflects this trade-off: GIF lies on the frontier, offering stronger final quality at a moderate time cost.

D. NasBench 301 (High-Dimensional Evaluation)

We now shift to a genuinely high-dimensional setting: NAS-Bench-301 (33D, DARTS-XGB surrogate). Figure 4 summarizes convergence in evaluations (left), wall-clock time (middle), and the score–time Pareto view (right). In evaluations, GIF keeps improving after other methods flatten out; around 340 evaluations, it overtakes all baselines. This indicates that the importance-guided scheduler continues to discover productive subspaces late in the run, and the warm-started full-space fallback helps it escape plateaus. In wall-clock time, GIF uses the budget efficiently: it reaches the top accuracy without being the slowest; SG is faster but stalls at a lower ceiling, and GP is slowest at the same budget. The Pareto panel makes the trade-off explicit: GIF and SG define the frontier—SG at the "faster but lower score" end, GIF at the "higher score at similar time" end—while GP, TPE, BOHB, and Random are dominated (either slower for similar accuracy or less accurate at similar time). As a result, in high dimensions, focusing trials on the most important groups while retaining periodic full-space search yields stronger final incumbents and higher accuracy for the time spent.

VII. Conclusion

Our study introduces Greedy Importance First (GIF), an importance-aware strategy that translates early hyperparameter-importance estimates into concrete scheduling decisions—grouping by importance, proportional allocation, and a safeguarded full-space fallback. Across diverse benchmarks, a consistent pattern emerges: GIF is most effective in high-dimensional regimes. On weighted analytic functions and NAS-Bench-301, it achieves both faster convergence and stronger final incumbents than strong baselines. On Bayesmark, where the effective dimensionality is smaller, GIF remains competitive, but its margins are limited on simpler models and become most pronounced on the MLPs—reflecting that importance-guided scheduling yields the biggest gains when many low-impact variables dilute progress and the landscape exhibits stronger anisotropy. In summary, GIF provides a simple, plug-compatible pathway to sample-efficient HPO in high dimensions; by reweighting effort toward important subspaces while maintaining a robust fallback, it offers practical utility for deep learning model tuning, and lays a foundation for future research on importance-aware AutoML systems.

Acknowledgment

RW acknowledges support from the China Scholarship Council. MG acknowledges support from the EPSRC project EP/X001091/1.

References

[1] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.

[2] S. Falkner, A. Klein, and F. Hutter, "Bohb: Robust and efficient hyperparameter optimization at scale," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1437–1446.

[3] I. Loshchilov and F. Hutter, "Cma-es for hyperparameter optimization of deep neural networks," *arXiv preprint arXiv:1604.07269*, 2016.

[4] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, pp. 1–52, 2018.

[5] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A.-L. Boulesteix *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 13, no. 2, p. e1484, 2023.

[6] P. Probst, A.-L. Boulesteix, and B. Bischl, "Tunability: Importance of hyperparameters of machine learning algorithms," *Journal of Machine Learning Research*, vol. 20, no. 53, pp. 1–32, 2019.

[7] R. Wang, I. Nabney, and M. Golbabaee, "Efficient hyperparameter importance assessment for cnns," in *International Conference on Neural Information Processing*. Springer, 2024, pp. 16–31.

[8] F. Hutter, H. Hoos, and K. Leyton-Brown, "An efficient approach for assessing hyperparameter importance," in *International conference on machine learning*. PMLR, 2014, pp. 754–762.

[9] S. Watanabe, A. Bansal, and F. Hutter, "Ped-anova: efficiently quantifying hyperparameter importance in arbitrary subspaces," *arXiv preprint arXiv:2304.10255*, 2023.

[10] R. D. Turner and D. Eriksson, "Bayesmark: Benchmark framework to compare bayesian optimization methods on real ml tasks," https://github.com/uber/bayesmark, 2019, accessed 2025-09-14.

[11] A. Zela, J. Siems, L. Zimmer, J. Lukasik, M. Keuper, and F. Hutter, "Surrogate nas benchmarks: Going beyond the limited search spaces of tabular nas benchmarks," *arXiv preprint arXiv:2008.09777*, 2020.

[12] Weights & Biases, "Visualize sweep results," https://docs.wandb.ai/guides/sweeps/visualize-sweep-results/, 2025, accessed: 2025-09-18.

[13] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[14] M. Lindauer, K. Eggensperger, M. Feurer, A. Biedenkapp, D. Deng, C. Benjamins, T. Ruhkopf, R. Sass, and F. Hutter, "Smac3: A versatile bayesian optimization package for hyperparameter optimization," *Journal of Machine Learning Research*, vol. 23, no. 54, pp. 1–9, 2022.

[15] J. Liu, F. Zhang, J. Guan, and X. Shen, "Uq-guided hyperparameter optimization for iterative learners," *Advances in Neural Information Processing Systems*, vol. 37, pp. 386–415, 2024.

[16] N. Mehta, J. Lorraine, S. Masson, R. Arunachalam, Z. P. Bhat, J. Lucas, and A. G. Zachariah, "Improving hyperparameter optimization with checkpointed model weights," in *European Conference on Computer Vision*. Springer, 2024, pp. 75–96.

[17] M. Wistuba, N. Schilling, and L. Schmidt-Thieme, "Scalable gaussian process-based transfer surrogates for hyperparameter optimization," *Machine Learning*, vol. 107, no. 1, pp. 43–78, 2018.

[18] K. Swersky, J. Snoek, and R. P. Adams, "Multi-task bayesian optimization," *Advances in neural information processing systems*, vol. 26, 2013.

[19] R. Wang, I. Nabney, and M. Golbabaee, "Grouped sequential optimization strategy–the application of hyperparameter importance assessment in deep learning," *arXiv preprint arXiv:2503.05106*, 2025.

[20] D. Ackley, *A connectionist machine for genetic hillclimbing*. Springer science & business media, 2012, vol. 28.

[21] A. Griewank, "Generalized descent for global optimization," *JOTA*, vol. 34, p. 15, 1985.

[22] L. A. Rastrigin, "Systems of extremal control," *Nauka*, 1974.

[23] J. Ansel *et al.*, "Opentuner: An extensible framework for program autotuning," in *International Conference on Parallel Architectures and Compilation Techniques*. IEEE, 2014, pp. 303–315.

[24] D. Eriksson, D. Bindel, and C. A. Shoemaker, "Scalable global optimization via local bayesian optimization," in *Advances in Neural Information Processing Systems*, 2019.

[25] T. Head, MechCoder, G. Louppe, I. Shcherbatyi, E. Fokoue *et al.*, "Scikit-optimize/scikit-optimize," https://scikit-optimize.github.io/, 2018.

[26] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, and F. Hutter, "Nas-bench-101: Towards reproducible neural architecture search," in *International conference on machine learning*. PMLR, 2019, pp. 7105–7114.

[27] X. Dong and Y. Yang, "Nas-bench-201: Extending the scope of reproducible neural architecture search," *arXiv preprint arXiv:2001.00326*, 2020.

[28] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[29] A. Klein, S. Falkner, J. T. Springenberg, and F. Hutter, "Fast bayesian optimization of machine learning hyperparameters on large datasets," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.