

HyperSHAP: Shapley Values and Interactions for Explaining Hyperparameter Optimization

Marcel Wever¹, Maximilian Muschalik², Fabian Fumagalli², Marius Lindauer¹

¹L3S Research Center, Leibniz University Hannover, Hannover, Germany

²MCML, LMU Munich, Munich, Germany

m.wever@ai.uni-hannover.de, maximilian.muschalik@ifi.lmu.de, f.fumagalli@lmu.de, m.lindauer@ai.uni-hannover.de

Abstract

Hyperparameter optimization (HPO) is a crucial step in achieving strong predictive performance. Yet, the impact of individual hyperparameters on model generalization is highly context-dependent, prohibiting a one-size-fits-all solution and requiring opaque HPO methods to find optimal configurations. However, the black-box nature of most HPO methods undermines user trust and discourages adoption. To address this, we propose a game-theoretic explainability framework for HPO based on Shapley values and interactions. Our approach provides an additive decomposition of a performance measure across hyperparameters, enabling local and global explanations of hyperparameters' contributions and their interactions. The framework, named HyperSHAP, offers insights into ablation studies, the tunability of learning algorithms, and optimizer behavior across different hyperparameter spaces. We demonstrate HyperSHAP's capabilities on various HPO benchmarks to analyze the interaction structure of the corresponding HPO problems, demonstrating its broad applicability and actionable insights for improving HPO.

Code — <https://github.com/automl/HyperSHAP>

Appendix — <https://arxiv.org/abs/2502.01276>

1 Introduction

Hyperparameter optimization (HPO) is an important step in the design process of machine learning (ML) applications to achieve strong performance for a given dataset and performance measure (Snoek et al. 2014; Bischl et al. 2023). Especially, this is true for deep learning, where hyperparameters describe the architecture and steer the learning behavior (Zimmer, Lindauer, and Hutter 2021). Also, for generative AI and fine-tuning of foundation models, HPO is key for achieving the best results (Yin et al. 2021; Tribes et al. 2023; Wang, Liu, and Awadallah 2023).

Hyperparameters affect the generalization performance of models in varied ways, with some having a more significant impact on tuning than others (Bergstra and Bengio 2012). The impact of hyperparameters on performance is highly context-dependent, varying with the dataset characteristics (e.g., size, noise level) and the specific performance measure being optimized (e.g., accuracy, F1) (Bergstra and Bengio 2012; van Rijn and Hutter 2018).

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

This complexity makes HPO particularly challenging, requiring opaque HPO methods to find optimal configurations within large search spaces (Feurer et al. 2015). Yet, even with an optimized configuration, understanding *why it outperforms others* remains difficult due to intricate effects and interactions among hyperparameters.

Despite their potential, HPO methods remain underused by domain experts, ML practitioners, and ML researchers (Lee et al. 2019; Hasebrook et al. 2023; Simon et al. 2023). This limited adoption is partly due to their rigidity and poor adaptability to special cases, but also to a lack of interpretability (Wang et al. 2019; Drozdal et al. 2020). The latter is a key requirement among HPO users (Wang et al. 2019; Xin et al. 2021; Hasebrook et al. 2023; Sun et al. 2023), and its absence has even led to a shift to manual tuning in high-stakes applications (Xin et al. 2021). For ML researchers, explanations are crucial to understand the contribution of individual components and retain control over model behavior. HPO researchers rely on such insights to analyze method performance and behavior. Prior work on hyperparameter importance and effects (Hutter, Hoos, and Leyton-Brown 2014; Moosbauer et al. 2021; Segel et al. 2023; Watanabe, Bansal, and Hutter 2023; Theodorakopoulos, Stahl, and Lindauer 2024) highlights the need to close interpretability gaps to build trust and foster effective collaboration between HPO tools and ML practitioners (Lindauer et al. 2024). A complementary view is offered by *tunability* (Probst, Boulesteix, and Bischl 2019), measuring performance gains over defaults to guide whether and what to tune. Yet, explanation methods tailored to tunability remain scarce.

Contribution We formalize HYPERSHAP, a novel post-hoc HPO-explanation framework:

- (1) We propose HYPERSHAP as a collection of 5 explanation games and interpret them using Shapley values and interactions for specific configurations, hyperparameter spaces, and optimizers.
- (2) We showcase how HYPERSHAP can be employed for tackling various explanation tasks.
- (3) Comparing to fANOVA, we find that HYPERSHAP's explanations are more actionable to select subsets of hyperparameters for tuning.
- (4) We provide a publicly available reference implementation of HYPERSHAP via GitHub.

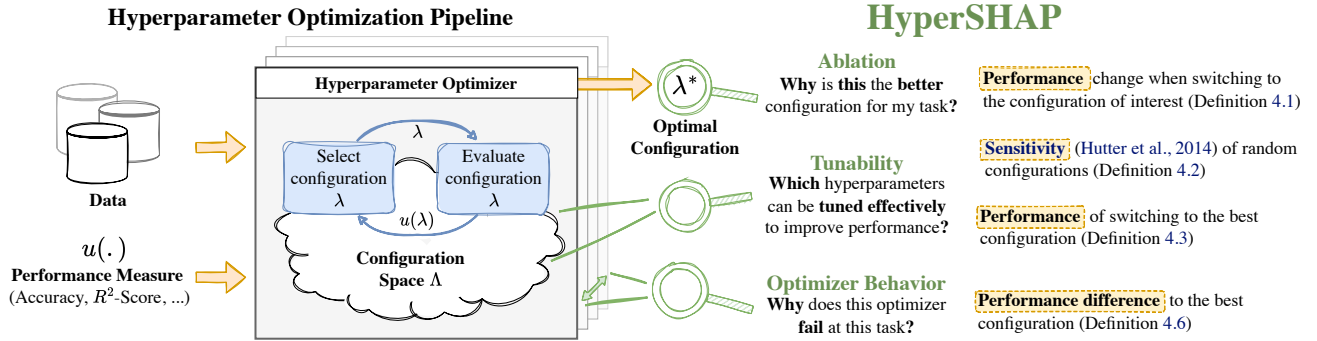


Figure 1: Game-theoretic explanations as defined with HYPERSHAP analyze hyperparameter values, hyperparameter spaces, and optimizers. HYPERSHAP can be used for data-specific explanations or across datasets. Test

2 Related Work

Hyperparameter importance (HPI) has gained significant attention in machine learning due to its crucial role in justifying the need for HPO (Pushak and Hoos 2020, 2022; Schneider et al. 2022), whereas tunability quantifies how much certain hyperparameters can be tuned for specific tasks (Probst, Boulesteix, and Bischl 2019). A variety of approaches have been developed to assess how different hyperparameters affect the performance of resulting models, ranging from simple (surrogate-based) ablations (Fawcett and Hoos 2016; Biedenkapp et al. 2017) to sensitivity analyses and eliciting interactions between hyperparameters based on fANOVA (Hutter, Hoos, and Leyton-Brown 2014; van Rijn and Hutter 2018; Bahmani et al. 2021; Watanabe, Bansal, and Hutter 2023). In this work, we propose a novel approach to quantifying HPI using Shapley values, with a particular focus on capturing interactions between hyperparameters through Shapley interaction indices. We focus on quantifying interactions since prior works (Zimmer, Lindauer, and Hutter 2021; Pushak and Hoos 2022; Novello et al. 2023) noticed that interaction is occasionally comparably low, which could serve as a foundation for a new generation of HPO methods that do not assume interactions to be omnipresent.

Beyond quantifying HPI, to better understand the impact of hyperparameters and the tuning behavior of hyperparameter optimizers, other approaches have been proposed, such as algorithm footprints (Smith-Miles and Tan 2012), partial dependence plots for hyperparameter effects (Moosbauer et al. 2021) or deriving symbolic explanations (Segel et al. 2023), providing an interpretable model for estimating the performance of a learner from its hyperparameters.

3 Hyperparameter Optimization

Hyperparameter optimization (HPO) is concerned with the problem of finding the most suitable *hyperparameter configuration* of a learner for a given task, typically consisting of some labeled dataset D and some performance measure u quantifying the usefulness (Bischl et al. 2023). To put it formally, let \mathcal{X} be an instance space and \mathcal{Y} a label space and suppose $x \in \mathcal{X}$ are (non-deterministically) associated with labels $y \in \mathcal{Y}$ via a joint probability distribution \mathbb{P} .

Then, a dataset $D = \{(x^{(k)}, y^{(k)})\}_{k=1}^N \subset \mathcal{X} \times \mathcal{Y}$ is a sample from that probability distribution. Furthermore, a predictive performance measure $u : \mathcal{Y} \times P(\mathcal{Y}) \rightarrow \mathbb{R}$ is a function mapping tuples consisting of a label and a probability distribution over the label space to the reals. Given a configuration $\lambda \in \Lambda$, a learner parameterized with λ maps datasets D from the dataset space \mathbb{D} to a corresponding hypothesis $h_{\lambda, D} \in \mathcal{H} := \{h \mid h : \mathcal{X} \rightarrow P(\mathcal{Y})\}$.

As a configuration $\lambda \in \Lambda$ typically affects the hypothesis space \mathcal{H} and the learning behavior, it needs to be tuned to the given dataset and performance measure. The task of HPO is then to find a configuration yielding a hypothesis that generalizes well beyond the data used for training. For a dataset $D \in \mathbb{D}$, the following optimization problem needs to be solved: $\lambda^* \in \arg \max_{\lambda \in \Lambda} \int_{(x, y) \sim \mathbb{P}} u(y, h_{\lambda, D}(x))$.

As the true generalization performance is intractable, it is estimated by splitting the given dataset D into training D_T and validation data D_V . Accordingly, we obtain $\lambda^* \in \arg \max_{\lambda \in \Lambda} \text{VAL}_u(\lambda, D)$, with

$$\text{VAL}_u(\lambda, D) := \mathbb{E}_{(D_T, D_V) \sim D} \sum_{(x, y) \in D_V} \frac{u(y, h_{\lambda, D_T}(x))}{|D_V|}.$$

Naïvely, HPO can be approached by discretizing the domains of hyperparameters and conducting a grid search or by a random search (Bergstra and Bengio 2012). More commonly, state-of-the-art methods often leverage Bayesian optimization and multi-fidelity optimization for higher efficiency and effectiveness (Bischl et al. 2023).

4 Explainable AI and Game Theory

Within the field of eXplainable AI (XAI), cooperative game theory has been widely applied to assign contributions to entities, such as features or data points for a given task (Rozenberczki et al. 2022). Most prominently, it is used to interpret predictions of black-box models using feature attributions (Lundberg and Lee 2017) and the Shapley Value (SV) (Shapley 1953). Shapley Interactions (SIs) (Grabisch and Roubens 1999) extend the SV by additionally assigning contributions to groups of entities, which reveal *synergies and redundancies*.

Such feature interactions reveal additive structures essential for understanding complex predictions (Sundararajan, Dhamdhere, and Agarwal 2020). Explanations consist of two components (Fumagalli et al. 2024): (1) an *explanation game* $\nu : 2^{\mathcal{N}} \rightarrow \mathbb{R}$, a set function over feature subsets of the n features of interest indexed by $\mathcal{N} = \{1, \dots, n\}$ that evaluates properties such as prediction or performance; (2) *interpretable* main and interaction effects derived from the SV and SIs. Analogously, the next section defines explanation games over hyperparameter ablations in VAL_u , using the SV and SIs to quantify tunability.

Explanation Games via Feature Imputations. Given the prediction of a black box model $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and an instance $\mathbf{x} \in \mathbb{R}^n$, *baseline imputation* with $\mathbf{b} \in \mathbb{R}^n$ for a coalition $S \subseteq \mathcal{N}$ is given by $\oplus_S : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ as

$$\nu_{\mathbf{x}}^{(\mathbf{b})}(S) := f(\mathbf{x} \oplus_S \mathbf{b}) \text{ with } \mathbf{x} \oplus_S \mathbf{b} := \begin{cases} x_i, & \text{if } i \in S, \\ b_i, & \text{if } i \notin S. \end{cases}$$

Baseline imputation is highly sensitive to the chosen baseline (Sturmfels, Lundberg, and Lee 2020). *Marginal* and *conditional* imputation extend this by averaging over randomized baselines (Sundararajan and Najmi 2020): $\nu_{\mathbf{x}}^{(p)}(S) := \mathbb{E}_{\mathbf{b} \sim p(\mathbf{b})}[f(\mathbf{x} \oplus_S \mathbf{b})]$, where $p(\mathbf{b})$ is the marginal or conditional feature distribution. The imputed predictions define *local* games for explaining individual predictions, while *global* games capture aggregate properties, e.g., variance or performance. As such, explanations increasingly reflect the underlying distribution p (Fumagalli et al. 2024).

Shapley Value (SV) and Shapley Interaction (SI). An explanation game is additively decomposed by the Möbius Interactions (MIs) $m : 2^{\mathcal{N}} \rightarrow \mathbb{R}$ (Muschalik et al. 2024), i.e. the Möbius transform (Rota 1964), for $T \subseteq \mathcal{N}$ as

$$\nu(T) = \sum_{S \subseteq T} m(S) \text{ with } m(S) := \sum_{L \subseteq S} (-1)^{|S|-|L|} \nu(L).$$

The MIs capture *pure main and interaction effects* but contain 2^n non-trivial components, too many for practical interpretation in ML applications (Muschalik et al. 2024). To reduce this complexity, the SV and SIs summarize the MIs into *interpretable* effects. The SV assigns contributions to individuals, is uniquely characterized, and satisfies four axioms: linearity, symmetry, dummy, and efficiency. The SV summarizes the MIs distributing each MI among the involved players via $\phi^{\text{SV}}(i) = \sum_{S \subseteq \mathcal{N}: i \in S} \frac{1}{|S|} m(S)$ for all $i \in \mathcal{N}$. Yet, the SV does not uncover interactions. Given an *explanation order* $k \in \{1, \dots, n\}$, the SIs Φ_k extend the SV to assign contributions to subsets of players up to size k . For $k = 1$ the SIs yield the SV and the MIs for $k = n$. Various forms of SIs exist, where positive values indicate synergy and negative values signal redundancy among the involved features. For instance, the Faithful Shapley Interaction Index (FSII) (Tsai, Yeh, and Ravikumar 2023) defines the best k -additive approximation $\hat{\nu}_k(S) := \sum_{L \subseteq S: |L| \leq k} \Phi_k(L)$ weighted by the Shapley kernel, enabling quantification of interaction strength. SIs thus offer a flexible trade-off between expressivity and complexity, a framework we now adapt to HPO.

5 Explaining Hyperparameter Optimization

Explanations in HPO are needed at multiple levels, from individual configurations to qualitative comparisons of HPO tools. Here, we consider four areas, dubbed Ablation, Sensitivity, Tunability, and Optimizer Bias. We begin with Ablation as the foundation of HYPERSHAP, extend it to Sensitivity (showing links to fANOVA by Hutter, Hoos, and Leyton-Brown (2014)), and compare it theoretically to Tunability. Tunability then serves to uncover Optimizer Bias. We conclude with practical considerations of HYPERSHAP. Let \mathcal{N} denote the set of hyperparameters; we quantify main and interaction effects based on the SV and SIs of the explanation games. Proofs are deferred to the appendix.

5.1 Ablation of Hyperparameter Configurations

One common approach to explaining HPO results is to compare a configuration of interest, λ^* , to a reference configuration λ^0 , typically a library default or a tuned default that has performed well on prior tasks. The configuration λ^* may stem from HPO or be manually selected. The key question is how changes in λ^* impact performance relative to λ^0 . To investigate this, we can incrementally modify λ^0 by replacing its hyperparameter values with those from λ^* , one at a time; a process known as *ablation*, widely used in empirical ML research (Cohen and Howe 1988; Rendsburg, Heidrich, and von Luxburg 2020; Herrmann et al. 2024).

HPO-ablation studies were proposed by Fawcett and Hoos (2016) but limited to sequential single-hyperparameter ablation paths, ignoring interactions. Instead, we form an explanation game for ablation using *all possible subsets*, which allows us to capture interactions.

Definition 1 (Ablation Game). The Ablation explanation game $\nu_{G_A} : 2^{\mathcal{N}} \rightarrow \mathbb{R}$ is defined as a tuple $G_A := (\lambda^0, \lambda^*, D, u)$, consisting of a *baseline (default)* configuration λ^0 , a target configuration λ^* , a dataset D , and a measure u . Given a coalition $S \subseteq \mathcal{N}$, we construct an intermediate configuration with $\oplus_S : \Lambda \times \Lambda \rightarrow \Lambda$ as

$$\lambda^* \oplus_S \lambda^0 := \begin{cases} \lambda_i^*, & \text{if } i \in S, \\ \lambda_i^0, & \text{else,} \end{cases}$$

and evaluate its value via $\nu_{G_A}(S) := \text{VAL}_u(\lambda^* \oplus_S \lambda^0, D)$.

The Ablation game quantifies the worth of a coalition based on the comparison with a baseline configuration λ^0 . In XAI terminology, this approach is known as *baseline imputation*. Natural extensions of the Ablation game capture these ablations with respect to a distribution $\lambda^0 \sim p^0(\lambda^0)$ over configuration space Λ as $\mathbb{E}_{\lambda^0 \sim p^0(\lambda^0)}[\text{VAL}_u(\lambda^* \oplus_S \lambda^0, D)]$, which relates to the *marginal performance* (Hutter, Hoos, and Leyton-Brown 2014). In XAI terminology, it is further distinguished between distributions $p(\lambda^0)$ that either depend (conditional) or do not depend (marginal) on the target configuration λ^* . Baseline imputation is often chosen for efficiency and is also argued to have desirable properties (Sundararajan and Najmi 2020). Still, the choice of baseline strongly influences the explanation (Sturmfels, Lundberg, and Lee 2020). We typically use a *default* configuration (Anastacio and Hoos 2020) here, though our methodology readily extends to probabilistic baselines.

5.2 Sensitivity and Tunability of Learners

Zooming out from a specific configuration, we can ask to what extent it is worthwhile to tune hyperparameters. In the literature, this question has been connected to the term of *tunability* (Probst, Boulesteix, and Bischl 2019). Tunability aims to quantify how much performance improvements can be obtained by tuning a learner, comparing against a baseline configuration, e.g., a configuration that is known to work well across various datasets (Pushak and Hoos 2020). In this context, we are interested in the importance of tuning specific hyperparameters. A classical tool to quantify variable importance is *sensitivity analysis* (Owen 2013), measuring the variance induced by the variables and decomposing their contributions into main and interaction effects.

Definition 2 (Sensitivity Game). The Sensitivity game $\nu_{G_V} : 2^{\mathcal{N}} \rightarrow \mathbb{R}$ is defined as a tuple $G_V := (\lambda^0, \Lambda, p^*, D, u)$, consisting of a *baseline* configuration λ^0 , a configuration space of interest Λ equipped with a probability distribution p^* , a dataset D , and a performance measure u . The value function is given by

$$\nu_{G_V}(S) := \mathbb{V}_{\lambda \sim p^*(\lambda)}[\text{VAL}_u(\lambda \oplus_S \lambda^0, D)].$$

A large value of a coalition $S \subseteq \mathcal{N}$ in the Sensitivity game indicates that these hyperparameters are important to be set to the right value. Hutter, Hoos, and Leyton-Brown (2014) implicitly rely on the Sensitivity game and compute the fANOVA decomposition, quantifying *pure* main and interaction effects. In game theory, this corresponds to the MIs of the Sensitivity game, which can be summarized using the SV and SIs (Fumagalli et al. 2024).

While sensitivity analysis is a suitable tool in XAI, it has some drawbacks for measuring tunability (Probst, Boulesteix, and Bischl 2019). First, as illustrated below, the total variance being decomposed $\nu_{G_V}(\mathcal{N})$ highly depends on the chosen probability distribution p^* and the configuration space Λ . Moreover, it does not reflect the performance increase expected when tuning all hyperparameters, but variations (in any direction). Second, for a coalition of hyperparameters $S \subseteq \mathcal{N}$, we expect that the coalition’s worth (performance) increases when tuning additional hyperparameters, i.e., $\nu(S) \leq \nu(T)$, if $S \subseteq T$. This property is known as *monotonicity* (Fujimoto, Kojadinovic, and Marichal 2006), but *does not* hold in general for the Sensitivity game ν_{G_V} . For a simple example, we refer to the appendix. Based on Probst, Boulesteix, and Bischl (2019), we define an explanation game for tunability that exhibits monotonicity:

Definition 3 (Tunability Game). The Tunability game is defined by a tuple $G_T = (\lambda^0, \Lambda, D, u)$, consisting of a baseline configuration $\lambda^0 \in \Lambda$, a configuration space Λ , a dataset D , and a measure u . The value function is given by

$$\nu_{G_T}(S) := \max_{\lambda \in \Lambda} \text{VAL}_u(\lambda \oplus_S \lambda^0, D).$$

The Tunability game directly measures the performance obtained from tuning the hyperparameters of a coalition S while leaving the remaining hyperparameters at the default value λ^0 . The Tunability game is monotone, which yields the following proposition.

		Sensitivity		Tunability	
		$(0, 0)$	λ^*	$(0, 0)$	λ^*
Score	λ_1	1/4	1/4	1	0
	λ_2	$\frac{m}{(m+1)^2}$	$\frac{m}{(m+1)^2}$	1	0
	$\lambda_1 \times \lambda_2$	0	0	0	0

Table 1: Importance scores for a 2D HPO problem under the Sensitivity and Tunability games, with baseline set to $(0, 0)$ and optimum λ^* . Sensitivity assigns lower scores to hyperparameters with larger domains (λ_2). Setting $\lambda^0 = \lambda^*$ reduces the Tunability scores to 0; Sensitivity is unaffected.

Proposition 1. *The Tunability game yields non-negative SVs and non-negative pure individual (main) effects obtained from functional ANOVA via the MIs.*

While the *main effects* obtained from the Tunability game are non-negative, interactions can still be negative, indicating redundancies of the involved hyperparameters.

Comparing Tunability vs. Sensitivity. We now showcase the different results of the Tunability game vs. the Sensitivity game using an educational example. We consider a two-dimensional configuration space $\Lambda := \Lambda_1 \times \Lambda_2$ with discrete configurations $\Lambda_1 := \{0, 1\}$ and $\Lambda_2 := \{0, \dots, m\}$ for $m > 1$. The optimal configuration is defined as $\lambda^* := (1, m)$, and the performance is quantified by $\text{VAL}_u(\lambda, D) := \mathbf{1}_{\lambda_1=\lambda_1^*} + \mathbf{1}_{\lambda_2=\lambda_2^*}$, where $\mathbf{1}$ is the indicator function. That is, we observe an increase of performance of 1 for each of the hyperparameters set to the optimal configuration λ^* . Lastly, we set the configuration baseline to $\lambda^0 := (0, 0)$ or $\lambda^0 := \lambda^*$. Intuitively, we expect that both hyperparameters obtain similar importance scores, since they both contribute equally to the optimal performance $\text{VAL}_u(\lambda^*, D) = 2$. Moreover, if the baseline is set to the optimal configuration λ^* , we expect the score to reflect that there is no benefit of tuning. Since the hyperparameters affect the performance independently, we do not expect any interactions.

The HPI scores of the Sensitivity and Tunability game for the example are given by Table 1. Both approaches, Sensitivity and Tunability, correctly quantify the absence of interaction $\lambda_1 \times \lambda_2$. As opposed to the Tunability game, the Sensitivity game assigns smaller scores to the hyperparameter λ_2 due to the larger domain Λ_2 . In fact, the Sensitivity score of λ_2 roughly decreases with order m^{-1} . Moreover, the Tunability scores reflect the performance increase and, as expected, distribute the difference between the optimal and the baseline performance properly among the hyperparameters. In contrast, the Sensitivity scores decompose the overall variance, which depends on Λ and p^* . Lastly, setting the baseline configuration λ^0 to λ^* decreases the Tunability scores to zero, whereas the Sensitivity scores remain unaffected. In summary, Sensitivity reflects the variability in performance when changing the hyperparameter values, whereas Tunability reflects the benefit of tuning hyperparameters over the baseline.

5.3 Optimizer Bias

The Tunability game aims to explain the importance of hyperparameters being tuned, which can also be used to gain insights into the capabilities of a hyperparameter optimizer. In particular, by comparing the optimal performance with the empirical performance of a single optimizer, we can uncover biases and pinpoint specific hyperparameters that the optimizer of interest fails to exploit. We define a hyperparameter optimizer as a function $\mathcal{O} : \mathbb{D} \times 2^\Lambda \rightarrow \Lambda$, mapping from the space of datasets and a configuration space to a specific configuration.

Definition 4 (Optimizer Bias Game). The Optimizer Bias HPI game is defined as a tuple $G_O = (\Lambda, \lambda^0, \mathcal{O}, D, u)$, consisting of a configuration space Λ , a baseline λ^0 , the hyperparameter optimizer of interest \mathcal{O} , a dataset D and a measure u . For $S \subseteq \mathcal{N}$, we define $\Lambda^S := \{\lambda \oplus_S \lambda^0 : \lambda \in \Lambda\}$ and

$$\nu_{G_O}(S) := \text{VAL}_u(\mathcal{O}(D, \Lambda^S), D) - \nu_{G_T}(S).$$

Intuitively, the value function captures how much performance is lost relative to the best known configuration. In other words, with the help of Definition 4, we can pinpoint where the hyperparameter optimizer \mathcal{O} falls short, revealing, for example, whether it struggles to optimize certain hyperparameters or types thereof. The analysis can be conducted via inexpensive surrogate-based HPO benchmarks.

5.4 Practical Aspects of HYPERSHAP

This section addresses practical aspects of HYPERSHAP to efficiently approximate the proposed games and generalize them to multiple datasets.

Efficient Approximation. Naïvely, to evaluate a single coalition in Definition 3 of the Tunability game, we need to conduct one HPO run. While this can be costly, we argue that using surrogate models that are, e.g., obtained through Bayesian optimization, can be used to calculate the maximum efficiently. Surrogate models are commonly used in explainability methods for HPO, including fANOVA and related approaches (Hutter, Hoos, and Leyton-Brown 2014; Biedenkapp et al. 2017; Moosbauer et al. 2021; Segel et al. 2023). For HYPERSHAP, we can bound the approximation error for the explanations as follows:

Theorem 1. *For a surrogate model with approximation error ϵ , the approximation error of Shapley values and interactions in HYPERSHAP is bounded by 2ϵ .*

Optimizer Bias Analysis To analyze Optimizer Bias, we propose to approximate ν_{G_T} using a diverse ensemble of optimizers $\mathbb{O} := \{\mathcal{O}_i\}$, and choose the best result for Λ^S obtained through any optimizer from \mathbb{O} , forming a virtual optimizer, always returning the best-known value. This virtual best hyperparameter optimizer approximates

$$\nu_{G_T}(S) \approx \max_{\lambda^i \in \mathcal{O}_i(D, \Lambda^S)} \text{VAL}_u(\lambda^i, D).$$

Worst Case Analysis. In order to identify hyperparameters that should not be mistuned, we can conduct a worst-case analysis with HYPERSHAP by replacing the max by a min operator in Definition 3.

Game Extensions Across Multiple Datasets. In a more general setting, we are interested in explanations across multiple datasets, for which we can extend the previous games naturally as follows:

Definition 5 (Multi-Dataset Games). Given a collection of datasets $\mathcal{D} := \{D_1, \dots, D_M\}$, the corresponding games $\nu_G^{D_i}$ for $1 \leq i \leq M$ with $G \in \{G_A, G_V, G_T, G_O\}$, we define its multi-dataset variant with the value function $\nu_G^{\mathcal{D}}(S) := \bigoplus_{i=1}^M \nu_G^{D_i}(S)$, where \bigoplus denotes an aggregation operator, e.g., the mean or a quantile of the game values obtained for the datasets D_i .

Considering explanations across datasets enables a broader view of the impact of how individual hyperparameters and their interactions affect generalization performance. Aggregating coalition values reveals which hyperparameters are generally worth tuning, rather than just data-specific importance, justifying tuning recommendations or uncovering systematic optimizer biases beyond data-specific effects.

6 Experiments

We evaluate the applicability of HYPERSHAP across various explanation tasks and benchmarks. To this end, we rely on four HPO benchmarks: `lcbench` (Zimmer, Lindauer, and Hutter 2021), `rbv2_ranger` (Pfisterer et al. 2022), `PD1` (Wang et al. 2024), and `JAHs-Bench-201` (Bansal et al. 2022). The implementation is based on `shapiq` (Muschalik et al. 2024) and (will be) publicly available on GitHub¹. We provide details regarding the setup, interpreting plots, and more results in the appendix. Generally, positive interactions are colored in red and negative in blue.

6.1 Insights from Ablation and Tunability

First, we compare the results of the Ablation and the Tunability game in terms of hyperparameter importance and interactions (cf. Figure 3). We retrieve an optimized configuration of PD1’s `lmlb_transformer` scenario and explain it with the Ablation game. HYPERSHAP’s explanation shows that the majority of the performance increase is attributed to the initial learning rate (L-I), which is not surprising since it is also intuitively the most important one. However, using HYPERSHAP to create Tunability explanations reveals that both hyperparameters, L-I and optimizer momentum (O-M), are of equal importance with a negative interaction. Thus, the optimizer chose to tune L-I over O-M for the configuration in question, even though a similar performance improvement could have been achieved by tuning O-M instead. Hence, HYPERSHAP can reveal which hyperparameters were subject to optimization via the Ablation game, while the Tunability game emphasizes the potential contributions of hyperparameters and their interactions.

6.2 Higher-Order Interactions in HPO

Second, we investigate the interaction structure of HPO problems for individual and across datasets. In Figure 2, left (MI), and further in the appendix, we observe the presence of many higher-order interactions, which are difficult to interpret. The SIs (order 2) and SV in HYPERSHAP summarize the MI into *interpretable* explanations.

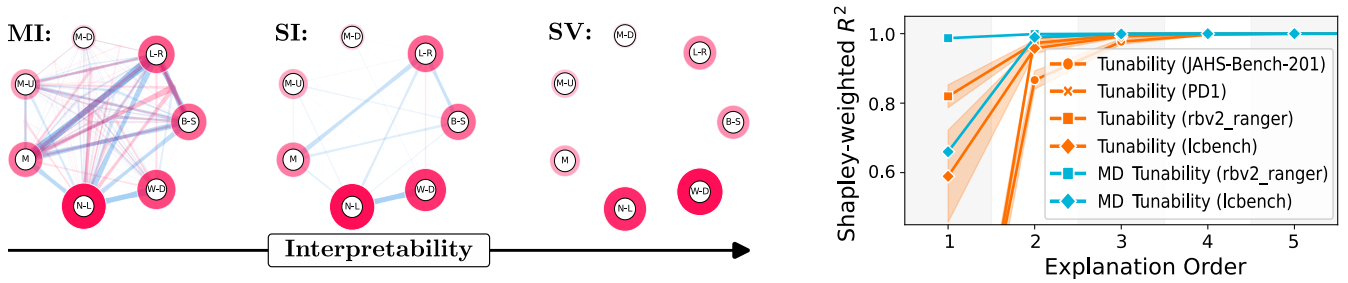


Figure 2: **Left:** Interaction graphs showing Möbius interactions (MI), second-order Shapley interactions (SI), and Shapley values (SV) where MIs terms are aggregated for interoperability. **Right:** Faithfulness of lower-order explanations approximating higher-order effects (Muschalik et al. 2024). An explanation order of 3 already approximates the full game ($R^2 \approx 1$) well.

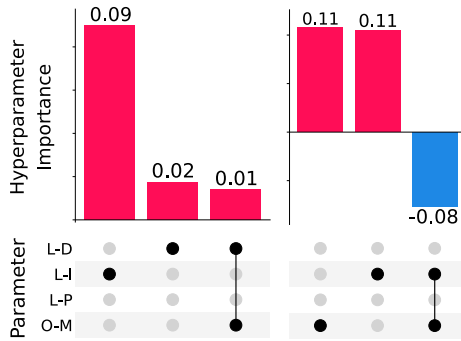


Figure 3: Upset plots for Ablation (left) and Tunability (right) of `lm1b_transformer` (Wang et al. 2024).

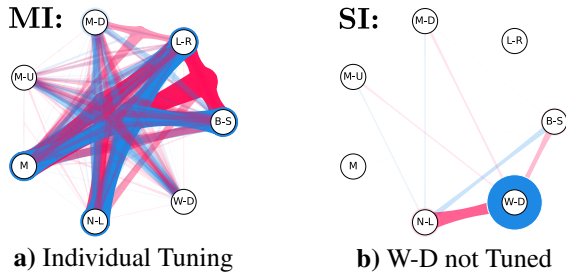


Figure 4: Interaction graphs showing results for the Optimizer Bias game via Moebius interactions (MI) and Shapley interactions (SI) on dataset ID 3945 of `lcbench`.

Figure 2, right, shows that SIs still *faithfully* capture the overall game behavior, which we measure with a Shapley-weighted loss (Muschalik et al. 2024) and varying explanation order. We find that most of the explanatory power is captured by interactions up to the third order, confirming prior research that suggests hyperparameter interactions are typically of lower order (Pushak and Hoos 2020). Interactions beyond the third order contribute little to the overall understanding of the game. Thanks to the convenient properties of the SV and SIs, HYPERSHAP provides a reliable way to capture and fairly summarize higher-order interactions into more interpretable explanations.

6.3 Detecting Optimizer Bias

The third experiment uses the Optimizer Bias game to investigate biases in black-box hyperparameter optimizers. To this end, we create two artificially biased hyperparameter optimizers. The first optimizer tunes each hyperparameter *separately*, ignoring interactions between them, while the second is *not allowed* to tune the most important hyperparameter. The virtual best hyperparameter optimizer is an ensemble of the investigated optimizer and five random search optimizers with a budget of 10,000 samples each. Ideally, a perfect optimizer would show no interactions and no main effects in HYPERSHAP’s Optimizer Bias explanations as the differences for every coalition would be 0.

Fig. 4 shows the Optimizer Bias explanations, i.e., the difference between two Tunability games, using the optimizer’s returned value and the (approximated) maximum, respectively. Note that main effects in the Optimizer Bias game can only be negative and show the optimizer’s inability to properly tune certain hyperparameters. In Fig. 4a, small main effects, in turn, suggest that the optimizer can effectively tune hyperparameters individually. The presence of both negative and positive interactions, which result from missing out on positive and negative interactions, respectively, shows that it fails to capture interactions. This confirms that the optimizer, which tunes hyperparameters independently, fails to capture their joint synergies. On the other hand, the second optimizer, ignoring the weight decay (W-D) hyperparameter for this particular dataset, clearly demonstrates bias in the interaction graph in Fig. 4b. The blue main effect for W-D and interactions involving W-D reveal this bias, showing how HYPERSHAP can help identify such flaws and contribute to the development of more effective HPO methods.

6.4 Explaining Bayesian Optimization

Inspired by Rodemann et al. (2024), we use HYPERSHAP to explain SMAC (Lindauer et al. 2022) – a state-of-the-art hyperparameter optimizer based on Bayesian optimization – by analyzing its surrogate model throughout the optimization process. We run SMAC with a total budget of 6 000 evaluations and inspect the surrogate model at 1%, 5%, 25%, and 100% of the budget. As shown in Fig. 5, we observe how the model’s belief about the main effects of and interactions between hyperparameters evolves over time.

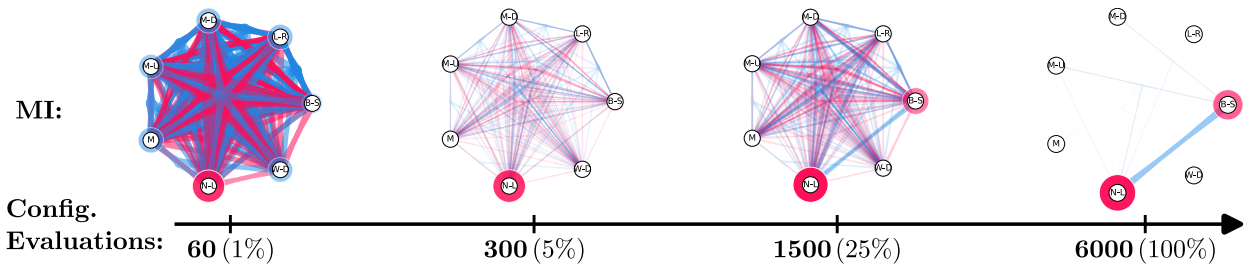


Figure 5: Explaining the surrogate model in SMAC’s Bayesian optimization with MIs at 1%, 5%, 25%, and 100% of the budget. Over time, SMAC notices first the importance of N-L and later B-S.

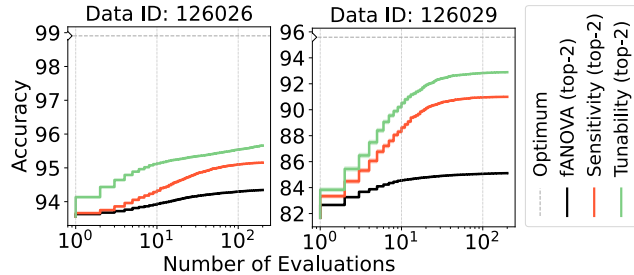


Figure 6: Anytime performance plots of HPO runs involving only the top-2 important hyperparameters for two datasets of lcbench (Zimmer, Lindauer, and Hutter 2021).

Early in the process, the surrogate model exhibits numerous large higher-order interactions, reflecting high uncertainty and a broad range of plausible interactions. Despite this, it already identifies N-L as an important hyperparameter. As optimization progresses, the model’s uncertainty decreases, leading to lower interaction values. By 25% of the budget, the optimizer uncovers B-S as another important hyperparameter, and the surrogate model briefly broadens its hypothesis about the performance landscape, resulting in increased interaction values. Eventually, as SMAC converges, the model refines its understanding of the performance landscape and interactions that are considered plausible are reduced. This analysis illustrates how HYPERSHAP can provide insights into the evolving dynamics of HPO processes.

6.5 Comparison with fANOVA

Lastly, considering the experimental setting proposed with fANOVA by Hutter, Hoos, and Leyton-Brown (2014), we compare HYPERSHAP and fANOVA on the task of selecting a subset of important hyperparameters to tune. To this end, for a given HPO task, we run fANOVA and HYPERSHAP using the SV with the Tunability and Sensitivity game to obtain explanations of order 1. Selecting the two most important hyperparameters, we conduct a subsequent HPO run optimizing only these two hyperparameters. As Tunability quantifies performance *gains*, we expect those explanations to be more suitable than explanations from Sensitivity, quantifying performance *variance* when tuning hyperparameters. Also, general hyperparameter importance, as in fANOVA, could yield less actionable explanations.

The results in Fig. 6 confirm that the anytime performance of the runs informed by HYPERSHAP is superior to that informed by fANOVA, and that Tunability outperforms Sensitivity here. For this specific task, the explanations from the Tunability game are more suitable.

6.6 Runtime Analysis

We found HYPERSHAP to be efficient across all evaluated settings. An Ablation game took 5s to 2m, while Tunability ranged from 6m to 15m for 7 and 4 hyperparameters, respectively, showing high dependency on the surrogate’s efficiency, and up to 8.5h for 10 hyperparameters on the JAHS-Bench-201 benchmark. As all coalition evaluations are independent, the method is highly parallelizable, enabling substantial wall-clock reductions. Overall, this adds modest overhead to typical multi-hour HPO runs.

7 Conclusion

In this paper, we proposed HYPERSHAP, a post-hoc explanation framework for consistently and uniformly explaining hyperparameter optimization using the SV and SIs across three levels: hyperparameter values, sensitivity and tunability of learners, and optimizer capabilities. Unlike previous methods that quantify variance (Hutter, Hoos, and Leyton-Brown 2014; Watanabe, Bansal, and Hutter 2023), HYPERSHAP attributes performance contributions. We demonstrated that HYPERSHAP not only enhances understanding of the impact of hyperparameter values or tunability of learners but also provides actionable insights for optimizing hyperparameters on related tasks.

Limitations & Future Work The computational bottleneck is the approximation of the max over $\lambda \in \Lambda$ via simulated HPO, requiring research on more efficient yet unbiased methods, e.g., via Bayesian algorithm execution (Moosbauer et al. 2022). Furthermore, extensions of HYPERSHAP to the analysis of optimizing machine learning pipelines are important future work (Heffetz et al. 2020; Feurer et al. 2022; Wever et al. 2020, 2021). Additionally, we plan to develop HPO methods that use tunability and hyperparameter importance to learn across datasets for improving their efficiency. This may allow warm-starting HPO in an interpretable way, complementing recent work on prior-guided HPO (Hvarfner, Hutter, and Nardi 2024; Fehring et al. 2025) and human-centered automated machine learning (Lindauer et al. 2024).

Acknowledgements

The authors acknowledge the computing time provided to them on the high-performance computers Noctua2 at the NHR Center PC2. These are funded by the Federal Ministry of Education and Research and the state governments participating on the basis of the resolutions of the GWK for the national highperformance computing at universities (www.nhr-verein.de/unsere-partner). Marcel Wever and Marius Lindauer acknowledge funding by the European Union (ERC, “ixAutoML”, grant no.101041029). Maximilian Muschalik acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): TRR 318/1 2021 – 438445824.

References

- Anastacio, M.; and Hoos, H. H. 2020. Model-Based Algorithm Configuration with Default-Guided Probabilistic Sampling. In *Proc. of Parallel Problem Solving from Nature*.
- Bahmani, M.; Shawi, R. E.; Potikyan, N.; and Sakr, S. 2021. To Tune or not to Tune? An Approach for Recommending Important Hyperparameters. arXiv:2108.13066.
- Bansal, A.; Stoll, D.; Janowski, M.; Zela, A.; and Hutter, F. 2022. JAHS-Bench-201: A Foundation For Research On Joint Architecture And Hyperparameter Search. In *Proc. of NeurIPS*.
- Bergstra, J.; and Bengio, Y. 2012. Random Search for Hyper-parameter Optimization. *J. Mach. Learn. Res.*, 13.
- Biedenkapp, A.; Lindauer, M.; Eggenberger, K.; Hutter, F.; Fawcett, C.; and Hoos, H. 2017. Efficient Parameter Importance Analysis via Ablation With Surrogates. In *Proc. of AAAI Conf. Artificial Intelligence*.
- Bischl, B.; Binder, M.; Lang, M.; Pielok, T.; Richter, J.; Coors, S.; Thomas, J.; Ullmann, T.; Becker, M.; Boulesteix, A.; Deng, D.; and Lindauer, M. 2023. Hyperparameter Optimization: Foundations, Algorithms, Best Practices, and Open Challenges. *WIREs Data. Mining. Knowl. Discov.*, 13.
- Cohen, P. R.; and Howe, A. E. 1988. How Evaluation Guides AI Research: The Message Still Counts More than the Medium. *AI Mag.*, 9(4).
- Drozdal, J.; Weisz, J.; Wang, D.; Dass, G.; Yao, B.; Zhao, C.; Muller, M.; Ju, L.; and Su, H. 2020. Trust in AutoML: Exploring Information Needs for Establishing Trust in Automated Machine Learning Systems. In *Proc. of Intern. Conf. Intelligent User Interfaces*.
- Fawcett, C.; and Hoos, H. 2016. Analysing Differences Between Algorithm Configurations Through Ablation. *J. Heuristics*, 22(4).
- Fehring, L.; Wever, M.; Spliethöver, M.; Hennig, L.; Wachsmuth, H.; and Lindauer, M. 2025. Dynamic Priors in Bayesian Optimization for Hyperparameter Optimization. arXiv:2511.02570.
- Feurer, M.; Eggenberger, K.; Falkner, S.; Lindauer, M.; and Hutter, F. 2022. Auto-sklearn 2.0: Hands-free AutoML via Meta-learning. *J. Mach. Learn. Res.*, 23.
- Feurer, M.; Klein, A.; Eggenberger, K.; Springenberg, J. T.; Blum, M.; and Hutter, F. 2015. Efficient and Robust Automated Machine Learning. In *Proc. of NeurIPS*.
- Fujimoto, K.; Kojadinovic, I.; and Marichal, J. 2006. Axiomatic Characterizations of Probabilistic and Cardinal-probabilistic Interaction indices. *Games and Economic Behavior*, 55(1).
- Fumagalli, F.; Muschalik, M.; Hüllermeier, E.; Hammer, B.; and Herbinger, J. 2024. Unifying Feature-based Explanations With Functional ANOVA and Cooperative Game Theory. In *Proc. of AISTATS*.
- Grabisch, M.; and Roubens, M. 1999. An Axiomatic Approach to the Concept of Interaction Among Players in Cooperative games. *Intern. Journal of Game Theory*, 28(4).
- Hasebrook, N.; Morsbach, F.; Kannengießler, N.; Zöller, M.; Franke, J.; Lindauer, M.; Hutter, F.; and Sunyaev, A. 2023. Practitioner Motives to Select Hyperparameter Optimization Methods. arXiv:2203.01717.
- Heffetz, Y.; Vainshtein, R.; Katz, G.; and Rokach, L. 2020. DeepLine: AutoML Tool for Pipelines Generation Using Deep Reinforcement Learning and Hierarchical Actions Filtering. In *Proc. of Intern. Conf. Knowledge Discovery & Data Mining*.
- Herrmann, M.; Lange, F. J. D.; Eggenberger, K.; Casalicchio, G.; Wever, M.; Feurer, M.; Rügamer, D.; Hüllermeier, E.; Boulesteix, A.; and Bischl, B. 2024. Position: Why We Must Rethink Empirical Research in Machine Learning. In *Proc. of ICML*.
- Hutter, F.; Hoos, H.; and Leyton-Brown, K. 2014. An Efficient Approach for Assessing Hyperparameter Importance. In *Proc. of ICML*.
- Hvarfner, C.; Hutter, F.; and Nardi, L. 2024. A General Framework for User-guided Bayesian Optimization. In *Proc. of Intern. Conf. Learning Representations*.
- Lee, D. J. L.; Macke, S.; Xin, D.; Lee, A.; Huang, S.; and Parameswaran, A. 2019. A Human-in-the-loop Perspective on AutoML: Milestones and the Road Ahead. *IEEE Data Eng. Bull.*, 42(2).
- Lindauer, M.; Eggenberger, K.; Feurer, M.; Biedenkapp, A.; Deng, D.; Benjamins, C.; Ruhkopf, T.; Sass, R.; and Hutter, F. 2022. SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization. *J. Mach. Learn. Res.*, 23(54).
- Lindauer, M.; Karl, F.; Klier, A.; Moosbauer, J.; Tornede, A.; Müller, A.; Hutter, F.; Feurer, M.; and Bischl, B. 2024. Position: A Call to Action for a Human-centered AutoML Paradigm. In *Proc. of ICML*.
- Lundberg, S.; and Lee, S. 2017. A Unified Approach to Interpreting Model Predictions. In *Proc. of NeurIPS*.
- Moosbauer, J.; Casalicchio, G.; Lindauer, M.; and Bischl, B. 2022. Enhancing Explainability of Hyperparameter Optimization via Bayesian Algorithm Execution. arXiv:2206.05447.
- Moosbauer, J.; Herbinger, J.; Casalicchio, G.; Lindauer, M.; and Bischl, B. 2021. Explaining Hyperparameter Optimization via Partial Dependence Plots. In *Proc. of NeurIPS*.
- Muschalik, M.; Baniecki, H.; Fumagalli, F.; Kolpaczki, P.; Hammer, B.; and Hüllermeier, E. 2024. Shapiq: Shapley Interactions for Machine Learning. In *Proc. of NeurIPS*.

- Novello, P.; Poëtte, G.; Lugato, D.; and Congedo, P. M. 2023. Goal-oriented Sensitivity Analysis of Hyperparameters in Deep Learning. *J. Sci. Comput.*, 94(3).
- Owen, A. 2013. Variance Components and Generalized Sobol' Indices. *SIAM/ASA J. Uncert. Quant.*, 1(1).
- Pfisterer, F.; Schneider, L.; Moosbauer, J.; Binder, M.; and Bischl, B. 2022. YAHPO Gym - An Efficient Multi-objective Multi-fidelity Benchmark for Hyperparameter Optimization. In *Proc. of International Conference on Automated Machine Learning*.
- Probst, P.; Boulesteix, A.; and Bischl, B. 2019. Tunability: Importance of Hyperparameters of Machine Learning Algorithms. *J. Mach. Learn. Res.*, 20.
- Pushak, Y.; and Hoos, H. 2020. Golden Parameter Search: Exploiting Structure to Quickly Configure Parameters in Parallel. In *Proc. of GECCO*.
- Pushak, Y.; and Hoos, H. 2022. AutoML Loss Landscapes. *ACM Trans. Evol. Learn. Optim.*, 2(3).
- Rendsburg, L.; Heidrich, H.; and von Luxburg, U. 2020. NetGAN without GAN: From Random Walks to Low-Rank Approximations. In *Proc. of ICML*.
- Rodemann, J.; Croppi, F.; Arens, P.; Sale, Y.; Herbinger, J.; Bischl, B.; Hüllermeier, E.; Augustin, T.; Walsh, C. J.; and Casalicchio, G. 2024. Explaining Bayesian Optimization by Shapley Values Facilitates Human-AI Collaboration. *CoRR*.
- Rota, G. 1964. On the Foundations of Combinatorial Theory: I. Theory of Möbius Functions. In *Classic Papers in Combinatorics*. Springer.
- Rozemberczki, B.; Watson, L.; Bayer, P.; Yang, H.; Kiss, O.; Nilsson, S.; and Sarkar, R. 2022. The Shapley Value in Machine Learning. In *Proc. of Intern. Joint Conf. Artificial Intelligence*.
- Schneider, L.; Schäpermeier, L.; Prager, R. P.; Bischl, B.; Trautmann, H.; and Kerschke, P. 2022. HPOxELA: Investigating Hyperparameter Optimization Landscapes by Means of Exploratory Landscape Analysis. In *Proc. of Parallel Problem Solving from Nature*.
- Segel, S.; Graf, H.; Tornede, A.; Bischl, B.; and Lindauer, M. 2023. Symbolic Explanations for Hyperparameter Optimization. In *Proc. of Intern. Conf. Automated Machine Learning*, volume 224.
- Shapley, L. 1953. A Value for N-person Games. In *Contributions to the Theory of Games (AM-28), Volume II*. Princeton University Press.
- Simon, S.; Kolyada, N.; Akiki, C.; Potthast, M.; Stein, B.; and Siegmund, N. 2023. Exploring Hyperparameter Usage and Tuning in Machine Learning Research. In *Proc. of Intern. Conf. AI Engineering*.
- Smith-Miles, K.; and Tan, T. 2012. Measuring Algorithm Footprints in Instance Space. In *Proc. of IEEE Congress on Evolutionary Computation*.
- Snoek, J.; Swersky, K.; Zemel, R.; and Adams, R. 2014. Input Warping for Bayesian Optimization of Non-stationary Functions. In *Proc. of ICML*.
- Sturmfels, P.; Lundberg, S.; and Lee, S. 2020. Visualizing the Impact of Feature Attribution Baselines. *Distill*.
- Sun, Y.; Song, Q.; Gui, X.; Ma, F.; and Wang, T. 2023. AutoML in The Wild: Obstacles, Workarounds, and Expectations. In *Proc. of Human Factors in Computing Systems*.
- Sundararajan, M.; Dhamdhere, K.; and Agarwal, A. 2020. The Shapley Taylor Interaction Index. In *Proc. of ICML*.
- Sundararajan, M.; and Najmi, A. 2020. The Many Shapley Values for Model Explanation. In *Proc. of ICML*.
- Theodorakopoulos, D.; Stahl, F.; and Lindauer, M. 2024. Hyperparameter Importance Analysis for Multi-objective AutoML. arXiv:2405.07640.
- Tribes, C.; Benarroch-Lelong, S.; Lu, P.; and Kobayev, I. 2023. Hyperparameter Optimization for Large Language Model Instruction-tuning. arXiv:2312.00949.
- Tsai, C.; Yeh, C.; and Ravikumar, P. 2023. Faith-Shap: The Faithful Shapley Interaction Index. *J. Mach. Learn. Res.*, 24(94).
- van Rijn, J.; and Hutter, F. 2018. Hyperparameter Importance Across Datasets. In *Proc. of Intern. Conf. Knowledge Discovery & Data Mining*.
- Wang, C.; Liu, X.; and Awadallah, A. H. 2023. Cost-effective Hyperparameter Optimization for Large Language Model Generation Inference. In *Proc. of International Conference on Automated Machine Learning*.
- Wang, D.; Weisz, J.; Muller, M.; Ram, P.; Geyer, W.; Dugan, C.; Tausczik, Y.; Samulowitz, H.; and Gray, A. 2019. Human-ai Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. In *Proc. of ACM Human Computer Interaction*.
- Wang, Z.; Dahl, G.; Swersky, K.; Lee, C.; Mariet, Z.; Nado, Z.; Gilmer, J.; Snoek, J.; and Ghahramani, Z. 2024. Pre-trained Gaussian Processes for Bayesian Optimization. *J. Mach. Learn. Res.*
- Watanabe, S.; Bansal, A.; and Hutter, F. 2023. PED-ANOVA: Efficiently Quantifying Hyperparameter Importance in Arbitrary Subspaces. In *Proc. of Intern. Joint Conf. Artificial Intelligence*.
- Wever, M.; Tornede, A.; Mohr, F.; and Hüllermeier, E. 2020. LiBRE: Label-Wise Selection of Base Learners in Binary Relevance for Multi-Label Classification. In *Proc. of IDA*.
- Wever, M.; Tornede, A.; Mohr, F.; and Hüllermeier, E. 2021. AutoML for Multi-label Classification: Overview and Empirical Evaluation. *IEEE Pattern Anal. Mach. Intell.*, 43(9).
- Xin, D.; Wu, E. Y.; Lee, D. J. L.; Salehi, N.; and Parameswaran, A. 2021. Whither AutoML? Understanding the Role of Automation in Machine Learning Workflows. In *Proc. of Human Factors in Computing Systems*.
- Yin, Y.; Chen, C.; Shang, L.; Jiang, X.; Chen, X.; and Liu, Q. 2021. AutoTinyBERT: Automatic Hyper-parameter Optimization for Efficient Pre-trained Language Models. In *Proc. of Association of Computer Linguistics and Intern. Joint Conf. Natural Language Processing*.
- Zimmer, L.; Lindauer, M.; and Hutter, F. 2021. Auto-pytorch Tabular: Multi-fidelity MetaLearning for Efficient and Robust AutoDL. *IEEE Patt. Anal. Mach. Intell.*