ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240602

Hyperparameter Importance Analysis for Multi-Objective AutoML

Daphne Theodorakopoulos[a,b,*], Frederic Stahl[a] and Marius Lindauer[b,c]

[a]Marine Perception Research Department, German Research Center for Artificial Intelligence (DFKI)
[b]Institute of Artificial Intelligence (LUH|AI), Leibniz University Hannover
[c]L3S Research Center

Abstract. Hyperparameter optimization plays a pivotal role in enhancing the predictive performance and generalization capabilities of ML models. However, in many applications, we do not only care about predictive performance but also about additional objectives such as inference time, memory, or energy consumption. In such multi-objective scenarios, determining the importance of hyperparameters poses a significant challenge due to the complex interplay between the conflicting objectives. In this paper, we propose the first method for assessing the importance of hyperparameters in multi-objective hyperparameter optimization. Our approach leverages surrogate-based hyperparameter importance measures, i.e., fANOVA and ablation paths, to provide insights into the impact of hyperparameters on the optimization objectives. Specifically, we compute the a-priori scalarization of the objectives and determine the importance of the hyperparameters for different objective trade-offs. Through extensive empirical evaluations on diverse benchmark datasets with three different objective pairs, each combined with accuracy, namely time, demographic parity loss, and energy consumption, we demonstrate the effectiveness and robustness of our proposed method. Our findings not only offer valuable guidance for hyperparameter tuning in multi-objective optimization tasks but also contribute to advancing the understanding of hyperparameter importance in complex optimization scenarios.

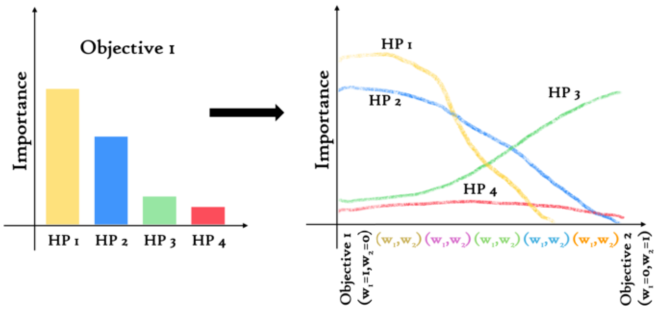

Figure 1: Overview of the MO-fANOVA method. On the left, the importance of each hyperparameter for Objective 1 is shown. On the right, our extension for the importance of each hyperparameter for different weightings of the objectives is displayed exemplarily.

1 Introduction

The selection of appropriate hyperparameter configurations significantly impacts a model's ability to capture underlying patterns in the data and produce accurate predictions. Optimizing hyperparameters is one of the main focus areas of Automated Machine Learning (AutoML) [22, 7]. Given a use case and a model, it is usually unknown which hyperparameters are worth tuning to achieve a good performance. Providing insights into this (e.g., [34, 42, 31]) is valuable since it allows the design of better configuration spaces and gives a better understanding of the learning dynamics of Machine Learning (ML) algorithms. HyperParameter Importance (HPI) offers a systematic method to gain insights into the influence of hyperparameters on the model's performance. Understanding and optimizing hyperparameters are crucial steps in building effective ML models. They allow us to fine-tune our algorithms, improve performance, and achieve better generalization. So far, most efforts in hyperparameter optimization (HPO) have been focused on single-objective optimization, mainly targeting the predictive performance of models. Recently, there has been a trend towards multi-objective HPO [18, 3, 13, 15, 6, 23], which allows the consideration of several objectives, including fairness, memory consumption, training time, inference time, and energy consumption. In a Multi-Objective Optimization (MOO) scenario, where we optimize conflicting objectives simultaneously, it becomes more challenging to approximate the unknown Pareto front and to discern the relative significance of individual hyperparameters. The configurations on the Pareto front are the set of non-dominated solutions.

Conventional methods for assessing HPI in single-objective scenarios rely on univariate sensitivity analysis techniques, like variance-based methods [21] or partial dependence plots [30]. For instance, techniques like fANOVA have been widely used to decompose the variance of model performance into contributions from individual hyperparameters [21]. While effective for single-objective optimization, these methods may not capture the intricate interactions among multiple objectives in multi-objective settings. So far, no methods have assessed the importance of hyperparameters within the context of MOO.

In this paper, we propose a novel method called weighted multi-objective HyperParameter Importance (MO-HPI). Although our approach is, in principle, usable with any surrogate-based HPI measure, we specifically focus on two widely-used methods, fANOVA [21] and ablation path analysis [14, 4]. Our methodology involves training surrogate models, e.g., random forests, with the hyperparameter configuration data and the respective objective results. For better vi-

* Corresponding Author. Email: daphne.theodorakopoulos@dfki.de.

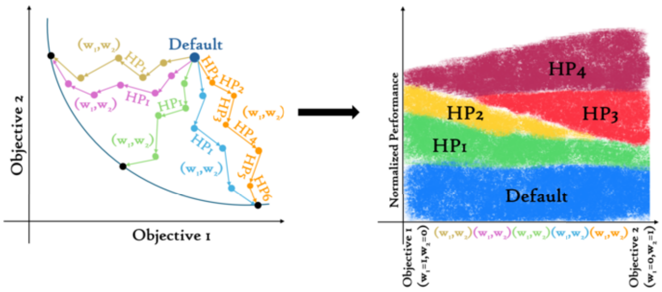

Figure 2: Overview of the MO-ablation path analysis. On the left, an exemplary Pareto front is displayed, with several ablation paths going from the default configuration to different configurations on the Pareto front. Every path is associated with a weighting of the objectives and thus gives a different value for the difference in performance per hyperparameter. We convert this to the plot on the right, where the total performance for different weightings is displayed as a stacked plot of hyperparameter contributions.

sualiziability and interpretability, we focus on bi-objective scenarios, but in principle, our approach can be used for any number of objectives. We use weighted sums of the objectives as target variables, allowing us to effectively capture the change in importance from one objective to another. Figure 1 shows the concept of MO-fANOVA, and Figure 2 the idea for the MO-ablation path analysis.

To analyze MO-HPI and demonstrate that it is reasonable in the context of ML, we pose the following research question: **How to convert surrogate-based HPI methods to meaningfully measure the importance of hyperparameters within multi-objective optimization?** We further frame the following subquestions: Are the results intuitively correct? How do the proposed methods compare? Can we gain new insights into the impact of hyperparameters on the optimization process that previous methods could not provide?

To answer our research questions and validate the efficacy of our proposed method, we conduct three empirical evaluations on the well-known benchmark datasets MNIST [9], Adult Census Income [2] and CIFAR10 [25]. To show that the approach can be applied to diverse objectives, we consider three pairs of objectives, each combined with accuracy: training time, demographic parity loss, and energy consumption, and discuss the results accordingly.

By advancing the understanding of HPI in complex optimization scenarios, our method offers a valuable technique that can aid researchers and practitioners in tackling challenging MOO problems more effectively.

2 Related Work

Several techniques are available for evaluating the importance of hyperparameters. **Surrogate models**, such as Gaussian process models [35] or random forests [8], have been employed to approximate the relationship between hyperparameters and performance. They can predict the performance of a given hyperparameter configuration based on an empirical dataset of configurations and their performances. Breiman [8] introduced how random forests can attribute importance by observing performance changes when removing attributes. Forward selection [20] uses this idea by selecting the hyperparameters that most affect the surrogate's performance by starting from no hyperparameters and iteratively adding the most impactful ones. Local Parameter Importance [5] studies the performance changes of a configuration along each hyperparameter by considering the variance in performance when changing the hyperparameter. Recent works [1, 37] use Shapley values [41] to measure the HPI in Bayesian optimization inspired by the fact that Shapley values can quantify the hyperparameter attributions for the acquisition function.

Another method is **ablation path analysis** [14, 4], which compares the default and optimized configuration to measure hyperparameter contributions. It creates an ablation path from the default configuration to a target configuration by changing the hyperparameter with the largest increase in performance in each iteration. We note that these ablation paths do not follow traditional ablation studies in which only a single hyperparameter is changed, and all others are fixed; however, this allows a full path through the configuration space from the default configuration to a target configuration. While there is only one path in a single-objective scenario, potentially, there are many in the MOO setting (cf. Figure 2).

fANOVA [21] identifies the importance of individual hyperparameters and interactions among them. For each hyperparameter, it measures how it contributes to the variance in performance. This is done, e.g., by training a random forest as a surrogate model and subsequently decomposing the variance of each tree into contributions to each subset of hyperparameters. Based on these methods, several works have been published. For example, van Rijn and Hutter [42], Probst et al. [34] and Moussa et al. [31] considered HPI across datasets. PED-ANOVA [44] improved fANOVA using Pearson divergence to work better on arbitrary subspaces of the search space, e.g., the subspace of the top-performing configurations. This paper contributes to this body of work by extending the methodology to MOO.

3 Weighted HPI for Multiple Objectives

Our main idea is simple and intriguing, given the insight that we are interested in the HPI for different objective tradeoffs. The main challenge lies in (i) how to map this to different HPI methods, (ii) how to compute this efficiently, (iii) how to visualize this, and (iv) how to interpret this. Challenges (i)-(iii) are described in this section, and (iv) follows as part of the following experiment and discussion sections. Overall, our work employs a framework for assessing the influence of hyperparameters on multiple objectives using any surrogate-based HPI analysis methods, allowing it to be efficient in computing analyses. Given a performance meta-dataset obtained by a MO-HPO optimizer, we train surrogate models to predict the objectives on unevaluated configurations; this is then used to calculate the HPI. The steps will be explained in more detail in the following section. While we discuss our approaches for two competing objectives, the approach can be straightforwardly extended to more objectives. The implementation is available on GitHub at https://github.com/automl/hpi_for_mo_automl.

In the following, we use the following notation. (O_1, O_2) denote two objective values. $(\tilde{O_1}, \tilde{O_2})$ are the normalized versions of them. We use $\vec{W}$ to denote a vector of weights weighting the objectives. If we sum the weighted objectives, we obtain Y_w. All the evaluated configurations are collected in Λ, where $\lambda \in \Lambda$ denotes one evaluated configuration. hps are hyperparameter names and hp_{min} is the most important one. To get the predicted (weighted) objective values r of any configuration, we train two surrogate models $\mathcal{S}_{obj}$, each trained with Λ and O_{obj} for the ablation path analysis, weighted after prediction. For fANOVA, we train one surrogate model per weighting $\mathcal{S}_w$ with Λ and Y_w (weighted before training).

3.1 Data Preparation.

The performance meta-dataset contains model configurations with their performance on several objectives. That means a table where each row contains hyperparameter configurations with the respective performance of all measured objectives. First, each objective is normalized using the min-max normalization across all evaluated configurations. Normalization is done to scale objectives, such as time, so that all objectives live on the same scale and thus are comparable in their magnitude. Next, the configuration data is converted to numerical values to deal with categorical data and NaN values. NaN values could occur because of hierarchical structures based on conditional hyperparameters, i.e., hyperparameters that are only part of the configuration if a certain value of another hyperparameter is chosen.

3.2 Weighting Scheme

Each configuration on the Pareto front implicitly refers to one weighted tradeoff of the objectives. Only Pareto-efficient objective pairs (o_1, o_2) are considered to retrieve the corresponding weighting. Subsequently, the normalized objective pairs are scaled to sum up to 1 by adding each pair up and dividing each of the two values by that sum to be used as weighting ($\tilde{o_1} = \frac{o_1}{o_1+o_2}$).

3.3 Multi-Objective fANOVA

Algorithm 1 Multi-Objective fANOVA

Input: evaluated and encoded configurations $\lambda \in \Lambda$, corresponding evaluated objective values (O_1, O_2)

$(\tilde{O_1}, \tilde{O_2}) \leftarrow$ normalized objective values
$\vec{W} \leftarrow$ is_pareto_efficient($(\tilde{O_1}, \tilde{O_2})$)
for each $\vec{w}$ in $\vec{W}$ **do**
    $Y_w \leftarrow w_1 \cdot \tilde{O_1} + w_2 \cdot \tilde{O_2}$
    $\mathcal{S}_w \leftarrow$ surrogate model trained with (Λ, Y_w)
    Calculate fANOVA importances with $\mathcal{S}_w$, e.g. following [21]
end for

A single target variable (Y) is created for each weighting (w_1, w_2) derived from the evaluated configurations $\lambda \in \Lambda$ on the Pareto front by summing the weighted objectives values (O_1, O_2):

$$Y_w = w_1 \cdot O_1 + w_2 \cdot O_2 \tag{1}$$

Given the weighting and the prepared data, MO-fANOVA can be calculated. Algorithm 1 shows the procedure in detail. A probabilistic surrogate model is trained for each weighting. The model takes the encoded configurations as input and the weighted normalized sum of the objectives as the target variable. After that, the fANOVA importance is calculated using the trained model for the corresponding weighting. See Figure 1 for an exemplary depiction of the outcome.

3.4 Multi-Objective Ablation Path Analysis

As shown by Biedenkapp et al. [4], ablation paths, i.e., the path of flipping the value of a hyperparameter from a given default configuration to a target configuration, can be efficiently approximated by using a surrogate model trained on meta-data collected by algorithm configuration or HPO. However, the standard procedure of computing these ablation paths by a greedy scheme [14] only considers a single objective.

Algorithm 2 Multi-Objective Ablation Path Analysis

Input: evaluated and encoded configurations $\lambda \in \Lambda$, corresponding evaluated objective values (O_1, O_2), hyperparameters hps

1: $\vec{W} \leftarrow$ is_pareto_efficient((O_1, O_2))
2: **for** each obj in (O_1, O_2) **do** ▷ Not normalized
3:     $\mathcal{S}_{obj} \leftarrow$ surrogate model trained with Λ, O_{obj}
4: **end for**
5: **for** each $\vec{w}$ in $\vec{W}$ **do**
6:     $\lambda_{\text{best}} \leftarrow \arg\min_{\Lambda}(w_1 \cdot O_1 + w_2 \cdot O_2)$
7:     $r_{\text{previous}} \leftarrow w_1 \cdot \tilde{\mathcal{S}}_1(\lambda_{\text{default}}) + w_2 \cdot \tilde{\mathcal{S}}_2(\lambda_{\text{default}}))$
8:     $r_{\text{min}} \leftarrow r_{\text{previous}}$
9:     $\lambda_{\text{previous}} \leftarrow \lambda_{\text{default}}$
10:     $\lambda_{\text{current}} \leftarrow \lambda_{\text{previous}}$
11:     **while** $r_{\text{min}} \leq r_{\text{previous}}$ **do** ▷ As long as we find improvements
12:         **for** hp in hps **do**
13:             $\lambda_{\text{current}}[hp] \leftarrow \lambda_{\text{best}}[hp]$
14:             $r \leftarrow w_1 \cdot \tilde{\mathcal{S}}_1(\lambda_{\text{current}}) + w_2 \cdot \tilde{\mathcal{S}}_2(\lambda_{\text{current}}))$
15:             **if** $r < r_{\text{min}}$ **then**
16:                 $hp_{\text{min}} \leftarrow hp$ ▷ Most important hyperparameter
17:                 $r_{\text{min}} \leftarrow r_{\text{total}}$
18:             **end if**
19:             $\lambda_{\text{current}} \leftarrow \lambda_{\text{previous}}$
20:         **end for**
21:         Add hp_{min} with r_{min} to ablation path
22:         $\lambda_{\text{previous}}[hp_{\text{min}}] = \lambda_{\text{best}}[hp_{\text{min}}]$
23:         Remove hp_{min} from hps
24:     **end while**
25: **end for**

We propose to extend the ablation path analysis for multiple objectives as follows; see also Algorithm 2. For both objectives, a surrogate model is trained with the encoded configuration data and the objectives before normalization (Line 2). For each weighting, the ablation path is calculated (starting from line 5). The incumbent λ_{best} is the corresponding configuration to the minimum of the scalarized target variable Y_w of Equation 1 (Line 6). Starting from the default configuration, each hyperparameter value is changed individually to the value of the incumbent configuration, and the performance is estimated with the trained surrogate models (Lines 12-20). The results of the models are normalized with the same normalization as the respective objective values, weighted, and summed up as performance (Line 14). The hyperparameter value leading to the highest difference in performance (toward the optimization objective) will be changed in the configuration (Lines 15-18 and line 22). The ablation is repeated with the altered configuration λ_{current} and the remaining hyperparameters until the incumbent configuration is reached (Line 11). The difference in performance is recorded for each weighting.

4 Experiments

Using our multi-objective HyperParameter Importance (MO-HPI) approach, we evaluated three HPO problems with different data, models, and objectives. The overview can be seen in Table 1. We chose rather simple benchmarks and models so that the results of the MO-HPI approach are easier to validate by common knowledge.

4.1 General Setup

The basic setup was the same for all experiments. We used SMAC [19, 27] as one of the state-of-the-art HPO tools [12] to perform HPO.

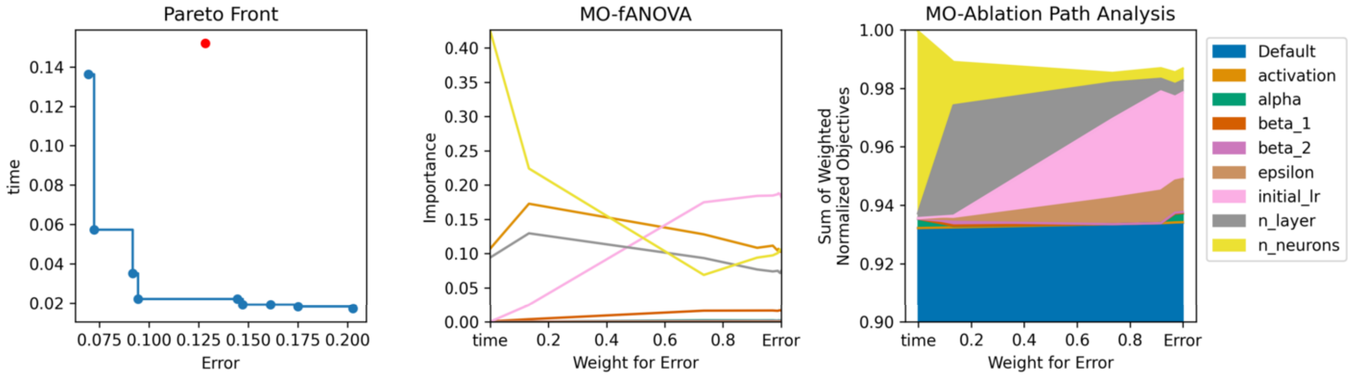

Figure 3: Results for the *time* experiment. The Pareto front is on the left (error vs. training time in seconds), with the red dot being the default performance. The MO-fANOVA results are in the middle, and the MO-ablation path analysis is on the right. The x-axis corresponds to the weighting of the minimum error objective.

Table 1: Overview of the experiments

Name	Dataset	Model	Objective 1	Objective 2
time	MNIST	MLP	1-Accuracy	Training Time
fairness	Adult Census	MLP	1-Accuracy	Demographic Parity Loss
energy	CIFAR10	ResNet	1-Accuracy	Energy Consumption

The multi-objective algorithm ParEGO [24] with Random Online Aggressive Racing (ROAR) [19] and 1000 trials was used for optimization. ROAR randomly selects a configuration from the hyperparameter space and only keeps track of the top 10 non-dominated configurations. We note that our primary goal was not to achieve the best possible multi-objective performance but to use a sufficient amount of performance data generated by an HPO tool. Since Moosbauer et al. [30] showed that post hoc analysis of HPO runs can be biased by too strong exploitation, we focus here on unbiased randomly generated data. For both HPI methods, we trained random forests with 100 trees as surrogate models. Note that reproducing our experiments will lead to slightly different results for the *time* and *energy* experiment, as those variables depend on factors such as the hardware.

4.2 Experiment: Time

Table 2: Hyperparameter Space for Multi-Layer Perceptron

Hyperparameter	Range	Scale	Default
n_layer	1-5	linear	3
n_neurons (same for all layers)	8-256	log	132
activation	logistic, tanh, relu	-	tanh
initial_lr	0.0001-0.1	log	0.01
alpha	0.0001-1.0	log	0.1
beta_1	0.1-1.0	log	0.5
beta_2	0.1-1.0	log	0.5
epsilon	1e-10-1e-06	log	1e-8

The first experiment used the MNIST dataset with Multi-layer Perceptron (MLP) classifiers based on Sci-kit learn's implementation [33]. The objective "training time" was measured by the wall-clock time the classifier used for training. The error was measured by 1 minus the accuracy on the test set. The MLPs were configured with 50 maximum epochs and Adam as the DL optimizer. SMAC tuned the MLPs hyperparameter w.r.t. the configuration space in Table 2. Figure 3 shows the Pareto front (left), the MO-fANOVA (middle), and the MO-ablation path analysis (right) results for the *time* experiment. The Pareto front shows the set of non-dominated solutions. The red point represents the default configuration. Note that for the x-axis, only the weighting for the error objective is displayed. Since the total weighting always sums to 1, the opposite weighting applies to the respective other objective — in this case, training time, so $w_2 = 1 - w_1$. MO-fANOVA thus displays the HPI from a low weighting of the error, so a high weighting of the time objective, to a high weighting of the error and a low weighting of time. For example, the number of neurons starts at an importance of around 0.4 when only considering the time objective and decreases the more the error objective is weighed in. The ablation path analysis plot is stacked, with each segment representing the performance increase attributed to tuning a specific hyperparameter for each weighting. The blue part shows the performance of the default configuration. Note that the y-axis only starts at 0.9, which means the default configuration already does quite well without tuning the hyperparameters.

4.3 Experiment: Fairness

The second experiment used the Adults Census Income dataset [2], which contains several variables about US citizens with the binary target variable annual income higher or lower than 50'000$. We only used the numeric and binary nominal variables of the dataset. It contains the sensitive variables "sex" and "race". We calculated the fairness loss as the second objective for the experiment based on the "race" variable prediction using demographic parity (DP). It is calculated by the absolute difference of the mean proportions of positive predictions y in each group, where the groups are defined by a sensitive variable s.

$$DPLoss = \left| \frac{\sum_{i=1}^{n} y_i(s_i = 0)}{n} - \frac{\sum_{i=1}^{n} y_i(s_i = 1)}{n} \right| \tag{2}$$

The MLPs were set up the same as in the *time* experiment. Figure 4 shows the Pareto front, the MO-fANOVA, and the MO-ablation path analysis results for the *fairness* experiment. While the optimization found several fair configurations, an error remained, with a top accuracy of around 83%. The plots can be interpreted the same way as described in section 4.2.

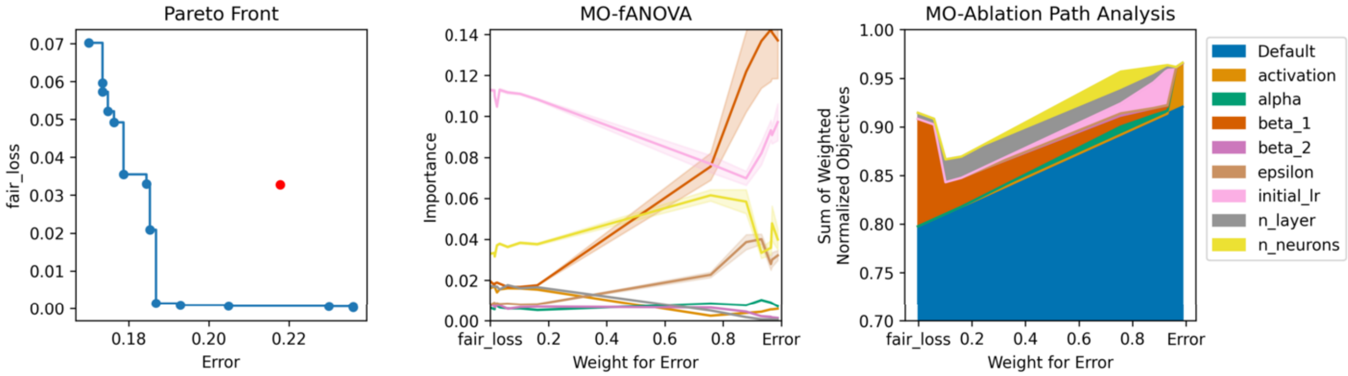

Figure 4: Results for the *fairness* experiment. The Pareto front is on the left (error vs. demographic parity loss), with the red dot being the default performance. The MO-fANOVA results are in the middle, and the MO-ablation path analysis is on the right. The x-axis corresponds to the weighting of the minimum error objective.

4.4 Experiment: Energy Consumption

Our last experiment used the Torchvision implementation of ResNets [17] with the first and the last layer set to a size of 3. The models were trained with AdamW and a cosine annealing learning rate scheduler with 200 maximum iterations. They were trained for 50 epochs with early stopping, which monitored the test accuracy with a patience of 6 epochs. The CIFAR10 dataset was always normalized with mean and standard deviation. Optionally, the data was augmented with a random crop of size 32 and a random horizontal flip. The objective "inference energy", so for predicting the test set, was estimated with CodeCarbon [26, 29][1]. The configuration space is displayed in Table 3. Figure 5 shows the Pareto front, the MO-fANOVA, and the MO-ablation path analysis results for the *energy* experiment. The plots can be interpreted the same way as described in section 4.2.

Table 3: Hyperparameter Space for ResNet

Hyperparameter	Belongs to	Range	Scale	Default
layer1	Model	1-30	linear	15
layer2	Model	1-30	linear	15
zero_init_residual	Model	true or false	-	true
augment	Dataset	true or false	-	false
learning rate	Optimizer	0.0001 - 0.1	log	0.01
weight_decay	Optimizer	0.00001 - 0.1	log	0.001
eps	Optimizer	1e-10-1e-06	log	1e-8

5 Discussion

Our research aimed to **investigate whether surrogate-based HyperParameter Importance (HPI) methods could be converted to meaningfully measure HPI within MOO**. We introduced the concept of weighting the objectives based on configurations along the Pareto front and applied it to two HPI methods: fANOVA and ablation path analysis. Subsequently, we conducted three MO-HPO experiments with diverse datasets, models, and objectives to assess the usefulness of the two MO-HPI approaches. In the following, we discuss our research questions related to the experimental results from the last section.

[1] https://github.com/mlco2/codecarbon

5.1 Are the results intuitively correct?

The results show the effectiveness of our method. All three experiments provide plots that make sense from the ML perspective, and thus, we conjecture that the results are intuitively correct. This can be seen in Figure 3 for the ablation and 5 for fANOVA, where hyperparameters related to network size (such as layer size and number of layers) strongly influence training time and energy consumption, consistent with the understanding that larger neural networks require more energy and time. A similar observation can be made for the learning rate. Its impact on the error is to be expected, given its well-known importance for achieving high accuracy. Additionally, in the second experiment, the learning rate plays a role in fairness (cf. Figure 4). The same is valid for data augmentation in the *energy* experiment, as is shown in Figure 5, it has a high importance, and it is known to heavily influence performance [43]. However, some unexpected results, like the high importance of beta_1 in the *fairness* experiment, are not intuitively explainable and warrant further exploration. We note that recent results showed that improved fairness can be strongly related to hyperparameters [11]. Overall, we therefore claim that the results are intuitively correct, which answers our first research question positively.

5.2 How do the proposed methods compare?

When comparing the two methods, i.e., MO-fANOVA and MO-ablation path analysis, some interesting observations can be made. In the *time* experiment, the activation function is less important when measured by the ablation path analysis than by fANOVA, but epsilon gains more importance for the error objective in the ablation path analysis. Another surprising observation is that in the *fairness* experiment, beta_1 is deemed important by both measures, but for the fANOVA measure, it has a strong influence on the error side, and for the ablation path analysis, it has a stronger influence on the fairness. Another difference is the high importance of the learning rate in Figure 4 for fANOVA. At the same time, there is not much contribution of the learning rate visible for the ablation path analysis. In the *energy* experiment, in the ablation path analysis, data augmentation rated high, and the learning rate has minimal relevance in comparison to fANOVA.

The difference in the results of the methods is mostly due to their distinct nature. It is well known that the ablation paths struggle with correctly attributing importance in case of interactions of hyperparameters [14]. Since the value of two or more hyperparameters has

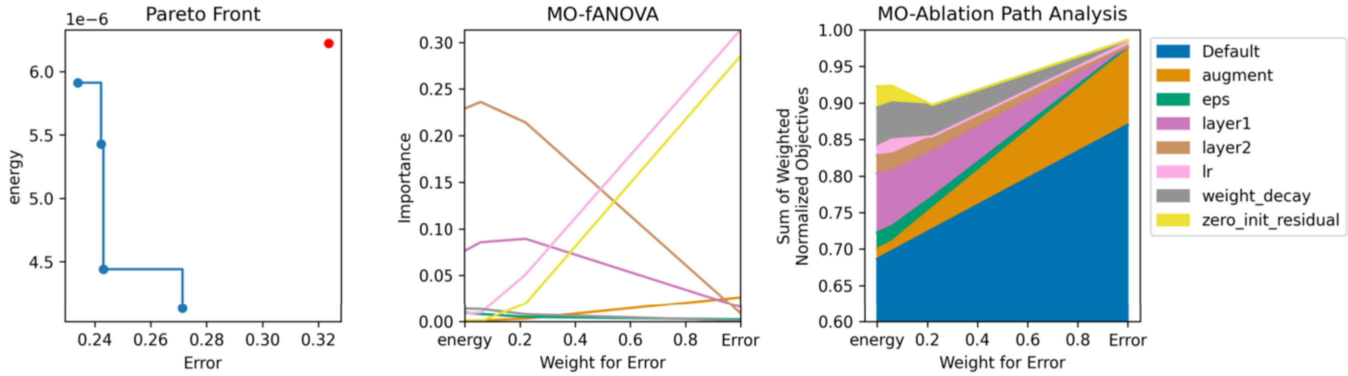

Figure 5: Results for the *energy* experiment. The Pareto front is on the left (error vs. inference energy consumption), with the red dot being the default performance. The MO-fANOVA results are in the middle and the MO-ablation path analysis is on the right. The x-axis corresponds to the weighting of the minimum error objective.

to be changed before an effect on the performance can be observed, the importance is only attributed to the last changed hyperparameter. Consequently, the performance differences upon changing the other hyperparameters might be lower than they actually are. In the single objective scenario, this is partially mitigated by also displaying the order of changes, which is not possible with our visualization.

Furthermore, the ablation results strongly depend on the chosen default configuration as a reference point. If the default configuration is already fairly close to one of the extreme ends of the Pareto front, there are little performance gains that can be attributed to the hyperparameters. Some hyperparameters might even seem unimportant because they are already set to the optimal value concerning the dominating objective. Therefore, it is important to always interpret the ablation path in view of the default configuration. Since fANOVA for main effects studies the variance for each hyperparameter independent of the others, it is more stable in that sense. Our method also allows us to study the importance of higher-order interaction effects with fANOVA, as in the single-objective case. To avoid cluttering our plots and results, we have omitted this here.

Lastly, it is important to note that ablation paths are considered a local HPI analysis since their path through the configuration space potentially only covers a small part of it and can be seen as a local interpretation of the incumbent configuration given a default configuration. In contrast, fANOVA covers the entire configuration space and thus provides more high-level insights, which, on the other hand, might not be important for specific performance improvements w.r.t. the default configuration. Overall, it is helpful to use both methods, as they provide diverse insights.

5.3 Can we gain new insights that previous methods could not provide?

We could simply run HPI studies on each objective independently and compare their results. We argue that by considering different tradeoffs of the objectives – as typically done in MOO – we gain valuable insights from our methods that were not possible previously.

Considering the tradeoffs between the objectives, we can make the following observations. In the fANOVA plot of the *time* experiment (cf. Figure 3), most hyperparameters significantly impacting only one of the objectives intersect at approximately an equal weighting of the objectives. This suggests that the MO-fANOVA in this case contains as much information as calculating fANOVA independently. Nevertheless, even in Figure 3, it is not a trivial linear trend of the importance values of the hyperparameters over different objective tradeoffs. In Figure 4 for the *fairness* experiment, the intersections are still at around the same position but are not in the center of the plot. For the *energy* experiment (cf. Figure 5), the intersections are even at different x-positions (i.e., objective weightings).

In addition, it is evident that certain hyperparameters lose importance rapidly (e.g., zero_init_residual in Figure 5 for the ablation path analysis), while others remain important for longer before they become irrelevant (e.g., beta_1 in Figure 4 for the ablation path analysis). Moreover, some hyperparameters would not be tuned when only looking at the objectives independently. For instance, in the ablation path analysis in Figure 4 (right), the initial learning rate only contributes significantly to the performance for certain tradeoffs but not to one of the extrema of the Pareto front.

Furthermore, it is interesting to observe that certain hyperparameters consistently retain importance, for instance, in Figure 4, for fANOVA, both the initial learning rate and the number of neurons remain influential. Conversely, some hyperparameters remain unimportant across all weightings. This consistency provides clarity regarding whether these hyperparameters require tuning. Finally, the stacked visualization makes it possible to easily see the relative contribution of a hyperparameter for different points on the Pareto front, which make up the weighting. Therefore, we conclude that there is more information in the plots that consider different objective tradeoffs compared to analyzing the objectives independently.

5.4 Limitations and Future Work

As is always the problem with post-hoc analysis, although our results are reasonable, the absence of ground truth data prevents us from guaranteeing their correctness. Nevertheless, we believe that there is sufficient empirical evidence allowing us to conjecture that our method is correct. Although several approaches for studying (single-objective) results of HPO are already presented [21, 4, 30, 44, 1, 37], a systematic set of criteria as proposed for interpretable ML [32] is still missing in the context of interpretable AutoML.

In our experiments, we have not explored MOO scenarios with more than two objectives because visualization and interpretability would become complex. Nonetheless, our approach can be extended to more objectives. Consider that the weighting scheme is calculated by dividing each coordinate value of a point on an n-dimensional Pareto front with the total sum of all values of that point, with each objective previously normalized between 0 and 1. For both methods, the target Y can now be calculated as the weighted sum of objectives with which the surrogate models can be trained. Plotting the results for three objectives in a 3D plot would be possible, but not trivially beyond that. We note that many-objective optimization (i.e., more than three objectives) could be relevant in practice but lead to a large fraction of the configurations being on the Pareto front. To the best of our knowledge, there are no reasonable approaches for many-objective HPO to date.

In future work, MO-AutoML could incorporate the proposed method to actively choose the most important hyperparameters to tune. This could be done in the MO setting by calculating the integrals of the MO-HPI over the different tradeoffs. This provides a way to quantify the importance one-dimensionally without having to decide on a tradeoff. This will lead to more efficient AutoML that learns to actively consider the HPI for different tasks, datasets, and models. Moreover, it is already known that post-hoc analysis with partial dependence plots of HPO runs can be skewed because HPO approaches, such as Bayesian optimization [40], tradeoff exploitation and exploration and thus bias the sampled configurations accordingly [30]. So far, it is not known how this affects approaches such as fANOVA and ablation paths, but it is reasonable to assume that they are similarly negatively affected. Therefore, future work has to include how this will affect our approach and new approaches for how to fix skewed results under biased data. In addition, our methods should be integrated into interactive tools, such as DeepCave [38] or IOHprofiler [10]. Another functionality could be enabling users to select configurations on the Pareto front and convert them into objective weightings.

6 Conclusion

In multi-objective hyperparameter optimization, assessing the importance of hyperparameters is challenging due to conflicting objectives. Our proposed method leverages surrogate-based hyperparameter importance measures, specifically fANOVA and ablation paths, in conjunction with a-priori scalarization across various objective tradeoffs. While we focus on those two HPI methods, our approach can also work with other HPI methods, such as Partial Dependence Plots [30], symbolic regression [39], variants of fANOVA [44], or maybe even with visualization methods such as parallel coordinate plots [16] or configuration footprints [5, 36]. Empirical evaluations demonstrate the effectiveness and interpretability of our method. We believe our proposed approach will enhance the analysis of multi-objective AutoML results and thus contribute to a human-centered AutoML paradigm [28].

Acknowledgements

The DFKI Niedersachsen (DFKI NI) is funded in the "zukunft.niedersachsen" by the Lower Saxony Ministry of Science and Culture and the Volkswagen Foundation (funding no. ZN3480). Marius Lindauer acknowledges support by the German Federal Ministry of the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (GreenAutoML4FAS project no. 67KI32007A) and was funded by the European Union (ERC, “ixAutoML”, grant no.101041029). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

European Research Council
Established by the European Commission

References

[1] M. Adachi, B. Planden, D. Howey, K. Muandet, M. Osborne, and S. Chau. Looping in the human: Collaborative and explainable bayesian optimization. *CoRR*, abs/2310.17273, 2023.

[2] B. Becker and R. Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[3] S. Belakaria, A. Deshwal, and J. Doppa. Max-value entropy search for multi-objective Bayesian optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alche Buc, E. Fox, and R. Garnett, editors, *Proceedings of the 33rd International Conference on Advances in Neural Information Processing Systems (NeurIPS’19)*. Curran Associates, 2019.

[4] A. Biedenkapp, M. Lindauer, K. Eggensperger, C. Fawcett, H. Hoos, and F. Hutter. Efficient parameter importance analysis via ablation with surrogates. In S. Singh and S. Markovitch, editors, *Proceedings of the Thirty-First Conference on Artificial Intelligence (AAAI’17)*, pages 773–779. AAAI Press, 2017.

[5] A. Biedenkapp, J. Marben, M. Lindauer, and F. Hutter. CAVE: Configuration assessment, visualization and evaluation. In R. Battiti, M. Brunato, I. Kotsireas, and P. Pardalos, editors, *Proceedings of the International Conference on Learning and Intelligent Optimization (LION)*, Lecture Notes in Computer Science. Springer-Verlag, 2018.

[6] M. Binder, J. Moosbauer, J. Thomas, and B. Bischl. Multi-objective hyperparameter tuning and feature selection using filter ensembles. In J. Ceberio, editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO’20)*, pages 471—479. ACM Press, 2020.

[7] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A. Boulesteix, D. Deng, and M. Lindauer. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley IRDMKD*, page e1484, 2023.

[8] L. Breiman. Random forests. *MLJ*, 45:5–32, 2001.

[9] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[10] C. Doerr, H. Wang, F. Ye, S. van Rijn, and T. Bäck. IOHprofiler: A benchmarking and profiling tool for iterative optimization heuristics. *arXiv:1810.05281 [cs.NE]*, 2018.

[11] S. Dooley, R. S. Sukthanker, J. P. Dickerson, C. White, F. Hutter, and M. Goldblum. Rethinking bias mitigation: Fairer architectures make for fairer face recognition. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023.

[12] K. Eggensperger, P. Müller, N. Mallik, M. Feurer, R. Sass, A. Klein, N. Awad, M. Lindauer, and F. Hutter. HPOBench: A collection of reproducible multi-fidelity benchmark problems for HPO. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Curran Associates, 2021.

[13] T. Elsken, J. Metzen, and F. Hutter. Efficient multi-objective Neural Architecture Search via lamarckian evolution. In *Proceedings of the International Conference on Learning Representations (ICLR’19)*, 2019. Published online: iclr.cc.

[14] C. Fawcett and H. Hoos. Analysing differences between algorithm configurations through ablation. *Journal of Heuristics*, 22(4):431–458, 2016.

[15] S. Gardner, O. Golovidov, J. Griffin, P. Koch, W. Thompson, B. Wujek, and Y. Xu. Constrained multi-objective optimization for automated machine learning. In L. Singh, R. De Veaux, G. Karypis, F. Bonchi, and J. Hill, editors, *Proceedings of the International Conference on Data*

Science and Advanced Analytics (DSAA'19), pages 364–373. IEEE, IEEE, 2019.

[16] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, and D. Sculley. Google Vizier: A service for black-box optimization. In S. Matwin, S. Yu, and F. Farooq, editors, *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17)*, pages 1487–1495. ACM Press, 2017.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR'16)*, pages 770–778. CVF and IEEE CS, IEEE, 2016.

[18] D. Hernández-Lobato, J. Hernández-Lobato, A. Shah, and R. Adams. Predictive entropy search for multi-objective bayesian optimization. In M. Balcan and K. Weinberger, editors, *Proceedings of the 33rd International Conference on Machine Learning (ICML'17)*, volume 48, page 1492–1501. Proceedings of Machine Learning Research, 2016.

[19] F. Hutter, H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In C. Coello, editor, *Proceedings of the Fifth International Conference on Learning and Intelligent Optimization (LION'11)*, volume 6683 of *Lecture Notes in Computer Science*, pages 507–523. Springer-Verlag, 2011.

[20] F. Hutter, H. Hoos, and K. Leyton-Brown. Identifying key algorithm parameters and instance features using forward selection. In P. Pardalos and G. Nicosia, editors, *Proceedings of the Seventh International Conference on Learning and Intelligent Optimization (LION'13)*, volume 7997 of *Lecture Notes in Computer Science*, pages 364–381. Springer-Verlag, 2013.

[21] F. Hutter, H. Hoos, and K. Leyton-Brown. An efficient approach for assessing hyperparameter importance. In E. Xing and T. Jebara, editors, *Proceedings of the 31th International Conference on Machine Learning, (ICML'14)*, pages 754–762. Omnipress, 2014.

[22] F. Hutter, L. Kotthoff, and J. Vanschoren, editors. *Automated Machine Learning: Methods, Systems, Challenges*. Springer, 2019. Available for free at http://automl.org/book.

[23] F. Karl, T. Pielok, J. Moosbauer, F. Pfisterer, S. Coors, M. Binder, L. Schneider, J. Thomas, J. Richter, M. Lang, E. Garrido-Merchán, J. Branke, and B. Bischl. Multi-objective hyperparameter optimization – an overview. *Transactions of Evolutionary Learning and Optimization*, 3(4):1—50, 2023.

[24] J. D. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

[25] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[26] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres. Quantifying the carbon emissions of machine learning. *arXiv:1910.09700*, 2019.

[27] 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. *JMLR*, 23(54):1–9, 2022.

[28] M. Lindauer, F. Karl, A. Klier, J. Moosbauer, A. Tornede, A. Müller, F. Hutter, M. Feurer, and B. Bischl. Position paper: A call to action for a human-centered automl paradigm. In *Proceedings of the international conference on machine learning*, 2024.

[29] K. Lottick, S. Susai, S. Friedler, and J. Wilson. Energy usage reports: Environmental awareness as part of algorithmic accountability. *arXiv:1911.08354*, 2019.

[30] J. Moosbauer, J. Herbinger, G. Casalicchio, M. Lindauer, and B. Bischl. Explaining hyperparameter optimization via partial dependence plots. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. Liang, J. Vaughan, and Y. Dauphin, editors, *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*, pages 2280–2291. Curran Associates, 2021.

[31] C. Moussa, Y. J. Patel, V. Dunjko, T. Bäck, and J. N. van Rijn. Hyperparameter importance and optimization of quantum neural networks across small datasets. *Machine Learning*, 113(4):1941–1966, 2024.

[32] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen, and C. Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI. *ACM Comput. Surv.*, 55(13s):295:1–295:42, 2023.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.

[34] P. Probst, A. Boulesteix, and B. Bischl. Tunability: Importance of hyperparameters of machine learning algorithms. *JMLR*, 20(53):1–32, 2019.

[35] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[36] A. Rasulo, K. Smith-Miles, M. Muñoz, J. Handl, and M. López-Ibáñez. Extending instance space analysis to algorithm configuration spaces. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion,*, pages 147–150. ACM, 2024.

[37] J. Rodemann, F. Croppi, P. Arens, Y. Sale, J. Herbinger, B. Bischl, E. Hüllermeier, T. Augustin, C. Walsh, and G. Casalicchio. Explaining bayesian optimization by shapley values facilitates human-ai collaboration. *CoRR*, abs/2403.04629, 2024.

[38] R. Sass, E. Bergman, A. Biedenkapp, F. Hutter, and M. Lindauer. Deepcave: An interactive analysis tool for automated machine learning. In M. Mutny, I. Bogunovic, W. Neiswanger, S. Ermon, Y. Yue, and A. Krause, editors, *ICML Adaptive Experimental Design and Active Learning in the Real World (ReALML Workshop 2022)*, 2022.

[39] S. Segel, H. Graf, A. Tornede, B. Bischl, and M. Lindauer. Symbolic explanations for hyperparameter optimization. In A. Faust, C. White, F. Hutter, R. Garnett, and J. Gardner, editors, *Proceedings of the Second International Conference on Automated Machine Learning*. Proceedings of Machine Learning Research, 2023.

[40] B. Shahriari, K. Swersky, Z. Wang, R. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[41] L. S. Shapley. *17. A Value for n-Person Games*, pages 307–318. Princeton University Press, Princeton, 1953. ISBN 9781400881970. doi: doi:10.1515/9781400881970-018. URL https://doi.org/10.1515/9781400881970-018.

[42] J. van Rijn and F. Hutter. Hyperparameter importance across datasets. In Y. Guo and F. Farooq, editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'18)*, pages 2367–2376. ACM Press, 2018.

[43] J. Wang and L. Perez. The effectiveness of data augmentation in image classification using deep learning. *arXiv:1712.04621 [cs.CV]*, 2017.

[44] S. Watanabe, A. Bansal, and F. Hutter. Ped-anova: Efficiently quantifying hyperparameter importance in arbitrary subspaces. In E. Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 4389–4396. International Joint Conferences on Artificial Intelligence Organization, 8 2023. doi: 10.24963/ijcai.2023/488. URL https://doi.org/10.24963/ijcai.2023/488. Main Track.