Synergies of Deep and Classical Exploratory Landscape Features for Automated Algorithm Selection

Moritz Seiler, Urban Skvorc, Carola Doerr, Heike Trautmann

Synergies of Deep and Classical Exploratory Landscape Features for Automated Algorithm Selection

Moritz Seiler[1], Urban kvorc[1], Carola Doerr[2], and Heike Trautmann[1,3]

[1] Machine Learning and Optimisation, University of Paderborn, Germany
{moritz.seiler,urban.skvorc,heike.trautmann}@uni-paderborn.de
[2] Sorbonne Université, CNRS, LIP6, Paris, France
carola.doerr@lip6.fr
[3] DMB Group, University of Twente, Netherlands

Abstract. Per-instance automated algorithm selection (AAS) aims at leveraging the complementarity of optimization algorithms with respect to different problem types. State-of-the-art AAS methods for numerical black-box optimization rely on supervised learning techniques that are supported by exploratory landscape analysis (ELA) feature sets. Recent works question the generalization ability of popular AAS approaches, which motivated the design of alternative feature sets.
In this work, we take a closer look at the recently proposed set of Deep ELA features and investigate the ways in which Deep ELA complements the classical ELA feature sets. To this end, we first study the correlation between the two feature collections, both through pairwise classification and through regression models. The complementarity observed in these analyses is confirmed by an AAS study, where models combining deep and classical features outperform those that are restricted to selecting from only of the two collections.

Keywords: Black-box Optimization · Exploratory Landscape Analysis · Automated Algorithm Selection · Deep Learning · Self-Supervised Learning · Feature Selection

1 Introduction

Automated Algorithm Selection (AAS) [24], aimed at automatically selecting the best performing algorithm for a given problem, as well as the closely related task of *Automated Algorithm Configuration* (AAC), which similarly aims at selecting the best performing configuration of a single algorithm, have long been of interest to the optimization community [2, 13]. This interest stems from the fact that both AAS and AAC can drastically reduce the runtime needed to solve a given optimization problem and simultaneously substantially improve optimization performance. As the algorithm performance is closely tied to the landscape of the optimization problem being solved, the task of selecting the best-performing algorithm or configuration requires some low-level information about the optimization landscape which are usually described as numerical values, referred to as *(instance-)features*.

In continuous single-objective black-box optimization, AAS and AAC have recently seen a growth in popularity, fueled in large parts by the availability of dedicated feature sets such as *Exploratory Landscape Analysis* (ELA) [15], which allow for the transformation of problem samples into landscape features without requiring prior domain-expert knowledge of the problems. ELA features can be conveniently calculated using programming libraries such as *flacco* [13] and *pflacco* [22]. This has led to a large amount of research in this area [5, 10, 11, 12, 16, 19, 20, 27]. In these works, the authors utilize landscape features and algorithm selection or configuration to demonstrate an overall reduced runtime in comparison to relying solely on the *Single-Best Solver* (SBS). In other words, the algorithm selection model is capable of selecting specific algorithms for each problem in a problem suite to improve performance compared to only using a single overall best-performing algorithm.

Despite the benefits provided by ELA, the features it produces are not without flaws. Commonly criticized drawbacks include (1) a large correlation between features, (2) concerns over the expressiveness and robustness of the features [23], (3) additional computational costs to compute the features, and (4) a lack of generalizability of the features to problems sets outside of the one that they were developed for [14, 17, 30].

As a result, there is currently a large research interest in developing alternative approaches for automatically representing a problem landscape. For example, the authors of [3, 25, 28] proposed learned features for continuous optimization problems. Rather than relying on experts to design feature sets manually, those authors used different training tasks to automatically extract instance features. While van Stein et al. [28] used a simple multi-layered autoencoder architecture including an unsupervised training task, Cenikj et al. [3] used a deep transformer architecture [29] trained on the *Black-Box Optimization Benchmarking* suite [BBOB, 9] by predicting the function identifier. Seiler et al. [25] also utilized a transformer architecture but relied on a self-supervised learning strategy that does not require any labeling.

However, when developing such learned features, it is important to ensure that these features are not simply novel, but that they also complement each other, as well as the original ELA features. Without complementarity, such newly developed features would only restate the knowledge that is already contained in the traditional ELA features, without necessarily improving the overall understanding of a problem's landscape. Nevertheless, we must also ensure that these newly developed features do not just contain novel information, but also still capture existing knowledge. This ensures that the newly developed features can be used on their own, and applied to domains for which no current feature sets exist.

Another benefit of performing complementarity analysis is that, particularly in the case of approaches based on deep learning, interpreting their results can be difficult. Comparing these newly developed features to ELA features, which are designed to correspond to a set of well-understood high-level problem properties, such as modality, global-to-local optima contrast, or separability, can help us better understand the behavior of the newly developed features.

Our contribution: In this paper, we will focus on the recently proposed *Deep ELA* feature set proposed in [25]. Deep ELA was shown to combine the benefits of both traditional ELA features (their ease of computation and lack of overfitting) with those of deep-learning-based feature-free approaches (in particular, their invariance to common problem transformations and their low correlation). Deep ELA has already shown promising results, drawing level with or even outperforming both traditional ELA and feature-free approaches in a single-objective AAS study [25]. However, the complementarity of the newly developed Deep ELA features and the original ELA features (which we will refer to as classical ELA features in the rest of the paper) is yet to be examined. Using correlation and regression analyses, we first observe that there is some overlap in the information captured by the classical ELA features and the Deep ELA ones. However, we also show that the sets complement each other, in that some features of the ELA set are difficult to predict by the Deep ones. We further verify this complementarity with an AAS study, which shows that combining the two approaches improves the performance over the stand-alone feature sets.

Paper organization: This paper is structured as follows. In general, we aim to analyze the complementarity and supplementarity of these two approaches, classical ELA and Deep ELA, in multiple ways. First, we provide the background necessary for the understanding of this paper, including an overview of classical ELA and Deep ELA (see Chapter 2). In Chapter 3, we compare the pairwise correlation of the Deep and classical ELA features. In Chapter 4 we train a *Support Vector Machine* (SVM) model to predict the classical ELA features using the Deep ELA features. Finally, we perform a study on AAS and compare the performance of models trained using the classical and Deep ELA features, as well as a model that uses the combination of both the classical and the Deep ELA features (see Chapter 5). If the classical and Deep ELA features are complementary, we would expect that the performance of the model increases when both types of features are used. On the other hand, we expect that Deep ELA features provide comparable performance if they mostly substitute classical ELA features. Last, we discuss in Chapter 6 the obtained results as well as their implications, summarize our results, and present avenues for future work.

2 Background and Methodology

To perform AAS using machine-learning models, we require some way of describing the landscape of an optimization problem that can be used by the model. One of the most common ways to do so in state-of-the-art research is the extraction of *Exploratory Landscape Analysis* (ELA) features. This approach, introduced by Mersmann et al. [15], allows for the automatic extraction of low-level ELA (instance-) features that correspond to specific high-level problem properties, such as the problem's modality or the ruggedness of its landscape. Further, the

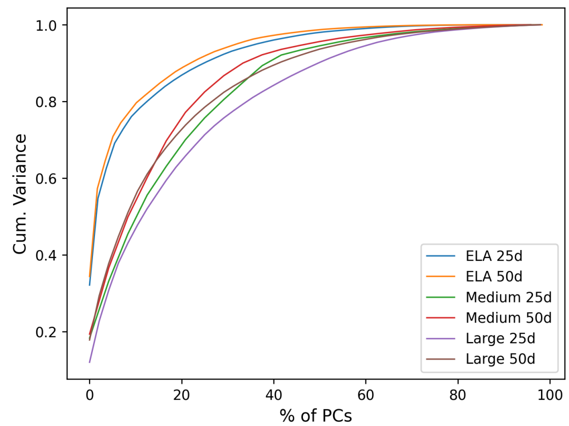


Fig. 1. Explainable Variance of classical (labeled as ELA **25d**/**50d**) and Deep ELA features (labeled as Medium/Large **25d**/**50d**). **25d** and **50d** correspond to the sample size while medium/large describes the model size of the Deep ELA models. Deep ELA features contain less redundant information as their principal components explain less of the overall variance.

ELA features can be calculated purely from a small amount of problem samples. Usually, a sample size of $50d$ or for improved robustness $250d$ are commonly chosen sample sizes that scale linearly with the dimensionality d of the decision space.

In the following, we will focus on a subset of these features, omitting categories that require additional samples (sampled iteratively) or the transformation from set-based into graph-based representation. These approaches differ substantially from Deep ELA which requires only a single set of samples and, hence, inhibit a fair comparison. In addition, the features that rely on graph-based representation are computationally expensive when applied to problems with a larger dimensionality. The used feature categories are:

Meta-Model features, which fit either a linear or a quadratic model to the provided problem samples, and use the metrics of this mode (such as the model's R^2 score) for the features.

PCA features, which compute the Principal Components of the sample, and use metrics such as the number of components required to explain 90% of the variance of the original samples as the features.

Nearest Better Clustering features, which examine the samples based on the distance between their nearest neighbors versus their better neighbors, where a better neighbor is a neighbor with better fitness.

Dispersion features, which compare the distribution of a subset of samples with the best fitness values against the distribution of the entire sample set.

Information Content features, based on a method for approximating the ruggedness of the problem landscape.

y-Distribution features, which represent descriptive statistics of the distribution of the fitness values, such as their skewness or kurtosis.

Linear model features, which divide the sample space into cells, and fit a linear model on each of the cells separately, then compare the various statistics of these models.

Fitness Distance Correlation features, which calculate the distances between samples in objective and in decision space, and compare various statistics between the two.

Specifically, in this paper, we consider the normalized variants of these features by `min-max` normalizing the objective space prior to the feature computation as proposed by [21].

However, as already noted in the introduction (see Chapter 1), ELA landscape features are not without flaws, and recent research has proposed several alternative methods. Hence, we will focus on an approach named Deep ELA [25], which proposed four pre-trained deep-learning models that were trained on 200 million randomly generated single- and multi-objective continuous optimization problems. Two of these models accept up to six input dimensions (medium models) while the other two accept up to twelve dimensions (large models). In addition, the authors experimented with input sizes of $25d$ and $50d$ for each medium and large architecture. The pre-trained models are transformer-based [29] and were trained using a self-supervised training task, closely following the idea proposed by [4].

Both ELA features and feature-free approaches have shown very promising results when applied to AAS and AAC. Kerschke et al. [11] provide an overview of AAS in the field of optimization, and note the success of using ELA to address the algorithm selection problem. In the field of single-objective continuous optimization, AAS has primarily focused on the 24 BBOB problems [9] of the COCO [8] benchmarking platform, a set of popular and well-known problems for benchmarking continuous optimization algorithms. Kerschke and Trautmann have shown that it is possible to reduce the expected running time on the set of 24 problems by half compared to a single best solver [12]. Prager et al. achieved only slightly worse results using a feature-free approach [20]. Finally, Deep ELA [25] consistently outperformed the feature-free approach described in [20], and in some cases outperformed the ELA-based approach described in [12]. Further, Deep ELA can be applied to AAS of multi-objective optimization problems (see Seiler et al. [25]) out of the box and have shown promise in combinatorial optimization [1]. Yet, we limit the scope of this paper to single-objective optimization problems.

3 Correlation Between Deep and Classical ELA

We begin our analysis with the comparison of the correlation between the Deep and classical ELA. In all that follows, we label the Deep ELA features as either *'Medium* $25d$*'*, *'Medium* $50d$*'*, *'Large* $25d$*'* or *'Large* $50d$*'*, depending on the size of

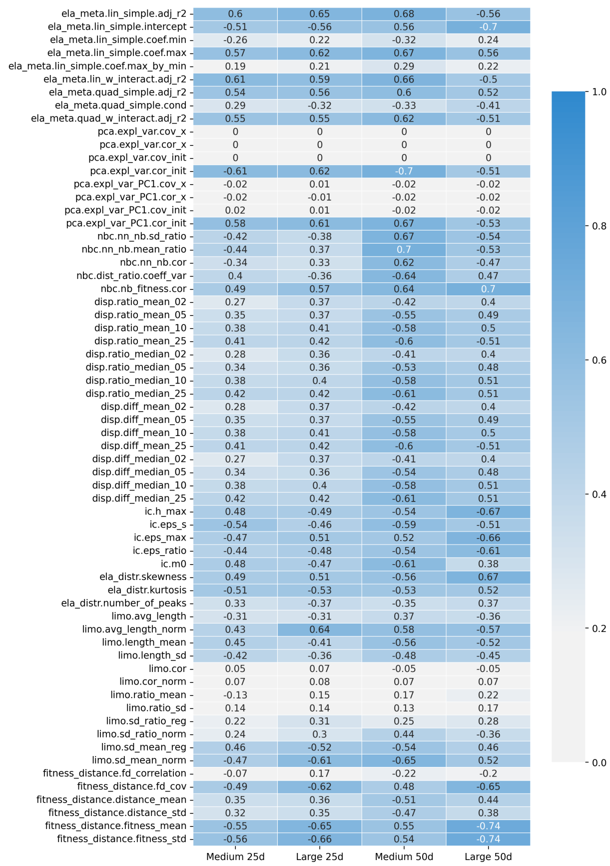
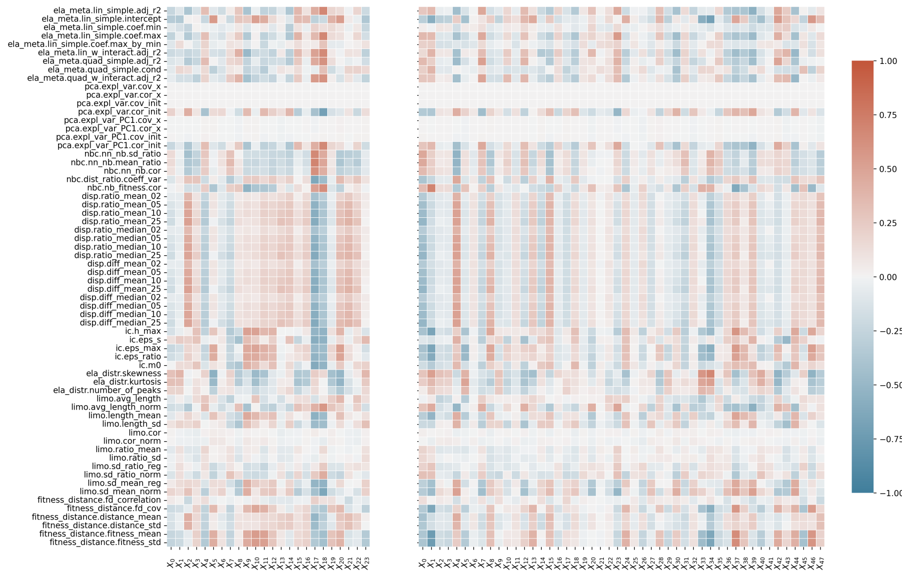


Fig. 2. Strongest correlation between the classical ELA features (rows) and the features of the Deep ELA models, for each of the four Deep ELA models (columns). The color scheme is based on absolute values; the darker the color, the stronger the (positive or negative) correlation.

the Deep learning model and the number of evaluated solutions per function that were used to train it, to which we refer to as *'sample size'* in the following. In this and all of the following experiments, both the classical and Deep ELA features are calculated on the 24 noiseless single objective BBOB problems, specifically on the first 20 instances of each function in dimensions 2, 3, 5 for the medium-sized model and, in addition, for dimension 10 for the large models. Following the recommendations made in [19, 20, 26], we base our feature computations on $50d$ and, in addition, also $25d$ samples, taken from the search space by *Latin Hypercube Sampling* (LHS). The Deep ELA features were computed by following closely the procedure described in [25].

Figure 1 shows the cumulative variance of the features calculated using the two approaches after performing *Principal Component Analysis* (PCA). We can



(a) Medium 50d (b) Large 50d

Fig. 3. Correlation heatmap between classical and Deep ELA features. In (a) the correlation between the Medium $50d$ model and the classical ELA features while in (b) the Large $50d$ model is depicted.

see that there is a distinct difference in the explained variance between the Deep and classical ELA. For the classical features, the first few components alone already explain about 60% of the variance, while the Deep features require a larger amount of components to reach this level. This indicates that the Deep ELA features contain less redundant information as PCA is efficient in reducing redundancy as well as noise. Hence, a low number of PCs that explain a large proportion of the original variance indicates that the original data contains many redundant and noisy information.

Further, Figure 3 shows the pairwise Spearman correlation between the classical and Deep ELA features using a sample size of $50d$ and the medium and large ELA models. We omit the comparison for the models with a sampling size of $25d$ as this produced similar results. Further, we can see that most classical ELA features have a weak to medium correlation to Deep ELA, which indicates that Deep ELA is somewhat representative of the characteristics of classical ELA. An exception to these are the `pca` and to an extent the `limo` feature groups, which are only weakly correlated with the Deep ELA features.

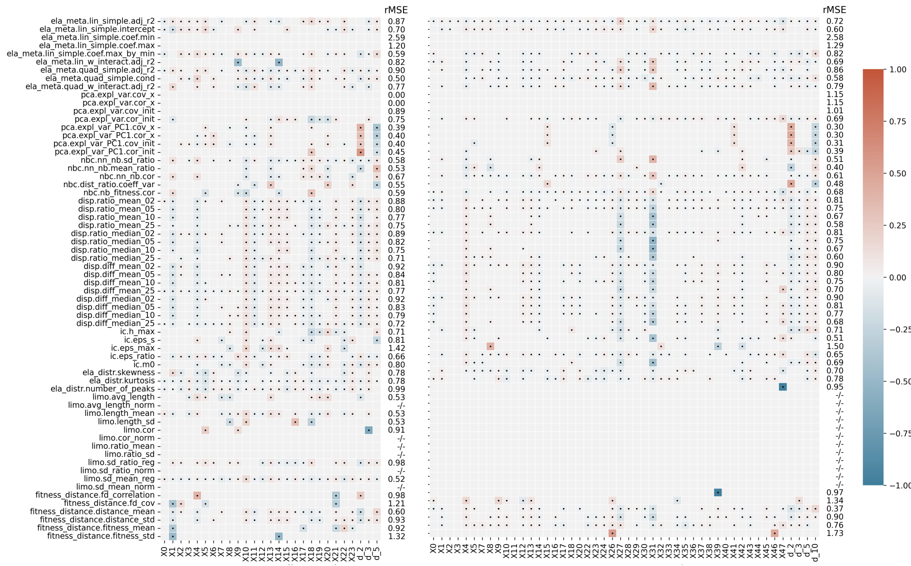
Finally, despite the fact that Deep ELA features are not comparable between different models as the training routine does not enforce any specific feature to be learned, we still see similar patterns emerge in both of the models. Further, as it is evident in Figure 2, the maximal correlation values between any Deep ELA feature and every classical ELA feature have similar absolute values across the different models. This is particularly interesting as (again) the training routine of the Deep ELA models does not enforce any specific and pre-defined feature to be learned. Instead, it lets the models learn any meaningful representation given an optimization problem as input. Yet, we can see in Figure 3 and Figure 2, that all four models learn similar features as these features have a similar correlation to the expert-designed classical ELA features. Yet, pairwise correlations are only a weak method to verify that certain feature sets contain similar information.

4 Classical ELA Feature Prediction

In this chapter, we conduct an additional experiment to determine whether the information contained in the entire set of Deep ELA features matches up with the information provided by the classical ELA features. To achieve this, the Deep ELA features are used to train a linear SVM model that predicts the classical ELA features. Further, we use *Recursive Feature Elimination* [RFE, 7] to select the most relevant Deep ELA features for predicting the classical ones, which allows us to see which Deep ELA features are most closely related to the classical features. RFE removes the least relevant feature based on the SVM's coefficients, iteratively.

Figure 4 shows the result of the feature prediction experiment. Here, the labels on the side of the figure show the *standardized root Mean Square Error* (rMSE) between the predicted and the target feature. Individual cells show us which Deep ELA features were considered important by the model to predict each classical ELA feature, after performing recursive feature elimination according to the coefficients of the trained SVM model. One can see that most of the features can be well predicted using an SVM model, which reinforces the findings from the correlation analysis, i.e. that the Deep ELA features mostly capture the information provided by the classical ELA features. In addition, one can also see that the `pca` feature category is predicted well, despite the low correlation with individual Deep ELA features, which shows the importance of this type of analysis, as it is important to consider the interactions between individual ELA features.

Predicting the feature `ela_meta.lin_simple.coef.max` achieves somewhat worse performance than the rest of the features in both models. This indicates that, despite most classical ELA features can be well substituted, Deep ELA features still do not capture all classical ELA features. On the other hand, we can also identify Deep ELA features that are somewhat irrelevant for predicting classical ELA features. An example is the X_{11} feature of the Large 50d model depicted in Figure 4 (b). This deep feature may be unrelated to classical ELA features and may capture certain information that is not captured by classical ELA features at all. Again, we find similar rMSE values between the Medium 50d and Large 50d models, which strengthens our finding that the Deep ELA models learn similar representations although their training routine does not enforce learning to encode anything specific structures or information of a given optimization problem.



(a) Medium 50d (b) Large 50d

Fig. 4. Feature importance of the Deep ELA features used to predict classical ELA ones using a Linear SVM in combination with recursive feature elimination. Selected Deep ELA features are indicated by a black dot, with the color of its surrounding box indicating the feature's importance. In some cases, ELA features contained missing values (which is within the definition of classical ELA features, e.g. `limo` features require a certain number of candidate solutions which is not always given in our setup) and, hence, no prediction was performed.

5 Automated Algorithm Selection Study

Finally, we examine the performance of the classical and Deep ELA features for the task of *Automated Algorithm Selection* (AAS). These experiments are performed using two different machine learning techniques to train the algorithm selectors: *k Nearest Neighbors* (kNN) and *Random Forests* (RF), using the same methodology as described in [12, 20, 25]. In addition, we applied *Sequential Forward Feature Selection* (SFFS) in combination with 5-fold cross-validation. The process of feature selection is in the outer loop. Therefore, the selected features are the same across the five folds and the five trained models. For both learners, kNN and RF, and for each of the two sampling sizes (25d and 50d) three different models are trained: (1) using classical ELA features (ELA), (2) using Deep ELA features (Deep), and (3) using a combination of both classical and Deep features (Comb.). The last model starts with the best set of Deep ELA features and sequentially adds classical ELA features. Other than that, we used the same setup as described in Kerschke et al. [12, 20, 25].

Table 1 lists classical ELA features that are often selected by either the classical ELA or the combined selectors. To be more precise, it lists all ELA

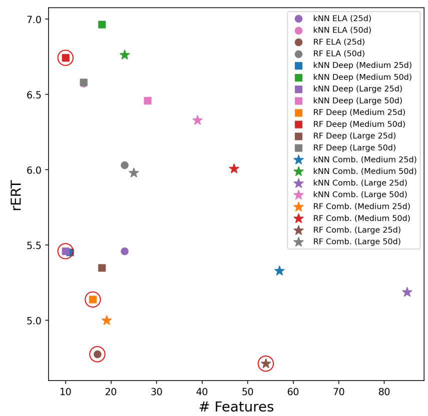
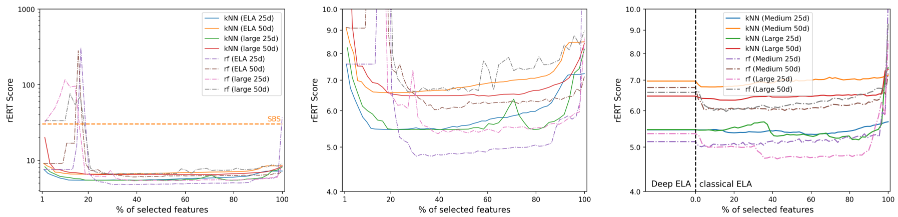


Fig. 5. Visualization of all trained AAS models based on the achieved performance (rERT) and the total number of features used. Non-dominated selectors are highlighted by a red circle.

features that were at least two times more often selected by one over the other. On the other hand, the dimensionality features as well as the PCA-based features are significantly less required by the combined selectors. This indicates again that the Deep ELA features represent similar meanings. On the other hand, the dispersion features are more often utilized by the combined selectors, noticeably. This indicates that the Deep ELA features do not substitute this information well.

Subfigures (a) and (b) in Figure 6 show how the performance of the models changes with the number of features selected. One can see that in the case of using just the classical or the Deep ELA features individually, only $20\% - 40\%$ of all available features produces close to optimal results. We also see that the kNN models work well even with a very small amount of features, while the random forests require a certain amount of features to reach performance better than the SBS. Subfigure (c) visualizes the improvements of a selector that utilizes only Deep ELA by the gradual addition of classical ELA features. We see that, for most models, an addition of only a few classical features results in a noticeable performance improvement, with the best-performing model achieving a further increase once around 30% of the features have been added. For most models, using all of the available landscape features produces a sharp decrease in performance. Figure 6 (c) also depicts a clear separation between $25d$ and $50d$ sample sizes, indicating that smaller sample sizes outperform larger ones. Selecting a sample size is always a trade-off between the accuracy of the computed features and additional costs. Larger sample sizes provide more (redundant) information which leads to higher stability of the compute features but also increases costs as more candidate solutions must be evaluated. In our case, we found that smaller costs ($25d$) outperform the higher feature quality ($50d$), indicating that ELA features (both Deep and classical) are even stable enough for small sample sizes.



(a) Large Models vs class. ELA

(b) Large Models vs class. ELA (Zoomed)

(c) Comb. of Deep and class. ELA

Fig. 6. Figures (a) and (b) show how the AAS model's rERT score changes with the number of features used, with Figure (b) showing only the rERT range of [4.0, 10.0] to allow for a more detailed examination once the rERT score starts to converge. Figure (c) depicts the impact of adding classical ELA features to the best set of Deep ELA features. Overall models, the best performance is achieved by a combined RF model 18 Deep ELA and 36 classical ELA features., for a total of 54 features.

Finally, Table 2 shows the full results of the algorithm selection study after the feature selection has been performed. Only a single model is trained for each column, with the rows representing the single model's performance separately by dimension and BBOB feature group. The selectors' performance is assessed by using the *relative Expected Running Time* (rERT), which measures the average time needed to reach a target precision (the found optimal solution is within a $\varepsilon = 0.01$ distance to the true global optima), normalized by the best-achieved average across the entire table. The rERT values also include the feature-costs which are in this case the number of sampled and evaluated candidate solutions.

One can see that all machine learning models mostly outperform the SBS baseline, with some exceptions in dimensions and feature groups where the SBS performance is close to the VBS. The final row presents the overall performance of each algorithm, with all algorithms that are stochastically tied to the best selector marked with a star using *Robust Ranking* with an $\alpha = 0.05$ [6]. Examples are (1) RF-based selector trained on classical ELA features and a sample size of $25d$ as well as (2) all models (kNN and RF as well as their Deep and Combined variants) trained on the Medium $25d$ features. The stochastic approach does not produce a clear winner in terms of the methodology used. However, in terms of pure performance, we can see that in all cases, the combined model using both classical and Deep ELA features outperforms the other two models. This indicates that neither Deep ELA nor classical ELA are complete substitutes for one another. In fact, both feature types encode some information that the other does not.

Overall, all types of models performed impressively compared to the SBS, outperforming it by a large margin both overall and in the majority of feature categories and dimensionalities. This matches prior work which has shown that ELA can be effectively used for automated algorithm selection on the BBOB problems, and also shows that Deep ELA achieves results competitive with the current state-of-the-art on this problem.

Table 1. List of the classical ELA features that are selected frequently by the classical ELA-based algorithm selectors and rarely by the Deep ELA-based selectors and vv.

Feature	ELA	Comb.	Feature	ELA	Comb.
d_2	7	3	ic.eps_max	6	4
d_3	6	2	ic.eps_ratio	3	5
disp.diff_median_02	0	2	ic.eps_s	7	4
disp.diff_median_05	1	3	ic.h_max	4	2
disp.diff_median_10	1	3	limo.avg_length	3	1
disp.diff_median_25	4	2	limo.length_mean	3	1
disp.ratio_mean_25	5	3	nbc.dist_ratio.coeff_var	1	3
ela_distr.number_of_peaks	4	2	pca.expl_var.cor_init	4	1
ela_distr.skewness	7	5	pca.expl_var.cor_x	6	3
ela_meta.lin_simple.coef.max	0	2	pca.expl_var.cov_init	6	3
fitness_distance.fd_correlation	2	0	pca.expl_var.cov_x	6	4
fitness_distance.fitness_std	4	6	pca.expl_var_PC1.cov_init	4	2

Figure 5 plots all of the models based on their performance versus the number of features used. We can see a set of non-dominated models consisting of five AAS models: *'RF Deep (Medium* $50d$*)'*, *'kNN Deep (Large* $25d$*)'*, *'RF Deep (Medium* $25d$*)'*, *'RF (ELA* $25d$*)'*, and *'RF Comb. (Large* $25d$*)'*. From this, we see that the deep features provide us both with a model that achieves the absolute best performance, as well as the one with the fewest features. However, the *'RF (ELA* $25d$*)'* model still achieves performance close to the best algorithm with only a small decrease in performance. We find those two models the most interesting: *'kNN Deep (Large* $25d$*)'* and *'RF Comb. (Large* $25d$*)'*. For the first one, the performance is still exceptionally good but only requires a minimal amount of features. On the other hand, *'RF Comb. (Large* $25d$*)'* provides the best performance but requires multiple times more features — especially a mixture of classical and Deep ELA features. Nonetheless, a sample size of $25d$ and the large Deep ELA model seem to be (generally speaking) superior and will be our main focus for future work.

6 Discussion & Conclusion

In this paper, we have analyzed the complementarity between classical and Deep ELA features as well as the substitutability of classical ELA by Deep ELA features. We additionally analyzed their performance for the task of automated algorithm selection, both when used individually and when the two are combined and used in a single model. Our experiments have shown that there is some amount of complementarity between Deep and classical ELA. While there is a certain degree of overlap, we have shown that the Deep ELA features provide additional information not captured by the classical ELA features and vice versa. This can be seen both in the complementarity analysis, as well as in the consistent improvement in algorithm selection performance when using a combined model of both classical and Deep ELA features. In general, all of the examined models achieved impressive algorithm selection results, consistent with prior work which shows great promise in algorithm selection on the BBOB problem set. Of the examined models, the combined model using classical ELA and the Large Deep ELA model trained on a sample size of $25d$ achieved the absolute best performance, this performance was not significantly better than some of its competitors. Especially, all models trained on the Medium $25d$ Deep

Table 2. Results of the AAS study. The numbers of selected features are in brackets; * indicates a result that is stochastically tied to the best selector with $\alpha = 0.05$ using the robust ranking technique as proposed by [6]. The first column (D) indicates the dimensionality of the problem while the second column (FGrp) indicates the function groups as defined in BBOB.

			Classical ELA				Large ($25d$)				Large ($50d$)				Medium ($25d$)				Medium ($50d$)			
		SBS	(23) kNN	(14) kNN	(17) RF	(23) RF	(10) kNN	(85) kNN	(18) RF	(54) RF	(28) kNN	(39) kNN	(14) RF	(25) RF	(11) kNN	(57) kNN	(16) RF	(19) RF	(18) kNN	(23) kNN	(10) RF	(47) RF
D	FGrp	(HCMA)	$25d$	$50d$	$25d$	$50d$	Deep	Comb.	Deep	Comb.	Deep	Comb.	Deep	Comb.	Deep	Comb.	Deep	Comb.	Deep	Comb.	Deep	Comb.
2	1	**3.71**	9.12	12.37	9.61	14.00	9.03	8.81	9.35	8.49	14.98	14.41	14.59	12.71	10.35	8.41	7.93	8.31	16.77	14.44	14.35	11.92
	2	5.80	3.23	3.61	3.26	3.52	3.26	**2.63**	2.78	2.65	3.54	3.54	4.42	3.53	3.01	3.11	2.65	3.45	3.51	3.53	4.68	3.73
	3	6.29	3.07	4.38	3.08	4.12	4.51	3.32	3.06	**2.73**	4.26	4.28	4.68	4.14	3.87	3.25	3.85	3.23	4.62	4.85	4.52	4.34
	4	25.34	6.74	6.85	6.63	6.81	7.31	7.04	7.11	5.72	4.36	4.48	5.04	6.21	4.76	5.99	5.92	5.78	4.00	4.47	5.35	**3.83**
	5	44.95	4.91	4.82	3.20	2.86	4.28	3.99	4.15	3.46	4.04	2.82	4.65	3.21	4.22	4.07	3.72	2.69	3.69	3.23	6.28	**2.54**
	all	17.69	5.50	6.52	5.23	6.38	5.78	5.26	5.40	**4.69**	6.35	6.00	6.77	6.06	5.33	5.05	4.90	4.74	6.64	6.21	7.13	5.33
3	1	356.10	10.98	15.53	**10.16**	15.07	10.79	11.43	11.11	10.36	15.96	15.75	16.08	14.68	10.88	10.75	10.97	11.09	16.00	16.04	15.77	15.40
	2	4.46	2.65	3.34	**2.49**	3.31	2.61	2.79	2.52	2.55	3.45	3.44	3.46	3.18	2.59	2.62	2.83	2.60	3.36	3.31	3.74	3.41
	3	4.98	2.55	4.05	2.59	3.72	3.88	2.72	2.56	**2.43**	3.75	3.76	3.85	3.80	2.88	2.60	3.17	2.92	3.72	3.63	3.85	3.88
	4	**2.63**	6.58	6.71	5.43	4.98	5.66	5.99	6.31	6.32	5.72	5.44	5.46	5.24	5.93	6.02	4.02	3.69	5.95	5.89	4.23	4.29
	5	66.81	2.56	2.71	2.09	2.02	2.53	2.64	2.33	2.39	1.92	2.16	4.86	**1.69**	2.77	2.62	2.12	2.03	2.72	2.79	2.94	2.86
	all	90.43	5.16	6.60	4.64	5.92	5.20	5.21	5.07	4.90	6.27	6.22	6.88	5.82	5.11	5.02	4.70	**4.54**	6.47	6.46	6.20	6.07
5	1	**11.99**	17.59	22.95	16.79	22.01	19.06	17.42	17.52	16.72	23.19	23.25	23.50	21.97	17.52	17.40	17.81	17.27	24.40	23.43	25.57	23.04
	2	3.90	3.07	3.30	2.85	4.19	3.65	3.54	3.61	4.11	2.58	2.77	3.28	3.36	2.49	2.49	**2.46**	2.47	3.71	3.70	3.38	2.98
	3	4.21	3.72	4.91	**3.17**	4.80	4.80	3.52	4.85	3.48	5.06	4.87	5.56	4.95	3.70	3.52	3.67	3.64	4.47	4.47	3.91	4.40
	4	4.29	4.86	4.15	4.00	3.86	4.28	4.17	4.09	3.95	3.95	3.89	4.50	3.44	2.96	3.78	3.70	3.09	4.00	3.89	2.88	**2.61**
	5	7.67	6.08	2.11	1.60	2.33	1.71	2.39	2.61	1.38	1.17	**1.16**	1.48	1.75	2.18	1.71	1.45	1.90	1.47	1.79	1.25	2.10
	all	6.52	7.23	7.66	**5.80**	7.57	6.83	6.32	6.66	6.00	7.38	7.37	7.84	7.25	5.91	5.92	5.96	5.81	7.77	7.61	7.56	7.19
10	1	**2.74**	9.54	16.34	8.63	15.42	9.54	9.60	9.50	8.60	16.76	16.58	16.20	15.41	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
	2	**2.16**	2.43	2.71	2.40	2.70	2.43	2.42	2.42	2.41	2.69	2.69	2.63	2.61	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
	3	**2.76**	3.62	4.20	2.79	3.72	3.62	3.60	3.59	2.83	4.88	4.56	4.06	3.67	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
	4	2.02	2.05	1.95	1.86	1.93	2.05	2.04	2.06	**1.85**	2.05	2.06	2.08	2.01	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
	5	23.64	1.74	1.78	1.68	1.76	2.16	1.77	4.76	**1.40**	2.15	2.07	2.49	1.50	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
	all	6.85	3.94	5.51	3.52	5.20	4.02	3.95	4.55	**3.46**	5.83	5.71	5.61	5.14	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
all	1	93.63	11.81	16.80	11.30	16.62	12.11	11.82	11.87	**11.04**	17.72	17.49	17.59	16.20	12.92	12.19	12.24	12.22	19.06	17.97	18.56	16.79
	2	4.08	2.84	3.24	2.75	3.43	2.99	2.84	2.83	2.93	3.06	3.11	3.45	3.17	2.70	2.74	**2.65**	2.84	3.53	3.51	3.93	3.37
	3	4.56	3.24	4.38	2.91	4.09	4.20	3.29	3.51	**2.87**	4.49	4.37	4.54	4.14	3.49	3.12	3.57	3.26	4.27	4.32	4.09	4.20
	4	8.57	5.06	4.91	4.48	4.40	4.83	4.81	4.89	4.46	4.02	3.97	4.27	4.22	4.55	5.27	4.55	4.19	4.65	4.75	4.15	**3.57**
	5	35.77	3.82	2.86	2.14	2.24	2.67	2.70	3.46	2.16	2.32	2.05	3.37	**2.04**	3.06	2.80	2.43	2.21	2.63	2.60	3.49	2.50
	all	30.37	5.46	6.57	4.80*	6.27	5.46	5.19	5.42	**4.76***	6.46	6.33	6.78	6.07	5.45*	5.33*	5.19*	5.03*	6.96	6.76	6.97	6.20

ELA models are stochastically tied to the Large $25d$ model, indicating superior performance of the $25d$ models versus the $50d$ ones.

The models trained using only the Deep ELA features also performed well, with some achieving performance that is stochastically tied to the best-performing model, and with one particular model achieving such performance using only 11 features total. The fact that the Deep ELA features were able to match the performance of the classical features makes them an important candidate for research into domains that cannot be tackled yet by classical ELA, such as multi-objective optimization, opening doors to a large amount of potential future research.

Another area that still needs to be assessed is the generalizability of Deep ELA to problems outside of the BBOB benchmark suite, where classical ELA features have been shown to struggle. In addition, Prager et al. [20] have demonstrated the successful application of cost-sensitive learning. In future work, we will investigate this approach more closely — especially in combination with Deep ELA features.

Finally, it will also be important to assess the complementarity of Deep ELA with other recently developed approaches for describing problem landscapes, such as TransOPT [3] and Topological Landscape Analysis [18]. However, the methods presented in this paper should be easily applicable to such a comparison, and we plan to conduct such an analysis in the future.

References

[1] Alissa, M., Sim, K., Hart, E.: Automated Algorithm Selection: from Feature-Based to Feature-Free Approaches. Journal of Heuristics **29**(1), 1–38 (2023)

[2] Belkhir, N., Dréo, J., Savéant, P., Schoenauer, M.: Per Instance Algorithm Configuration of CMA-ES with limited Budget. In: Proceedings of the 19th Annual Conference on Genetic and Evolutionary Computation (GECCO). pp. 681–688. ACM (2017)

[3] Cenikj, G., Petelin, G., Eftimov, T.: TransOpt: Transformer-based Representation Learning for Optimization Problem Classification (2023)

[4] Chen, X., Xie, S., He, K.: An Empirical Study of Training Self-Supervised Vision Transformers (2021)

[5] Derbel, B., Liefooghe, A., Vérel, S., Aguirre, H.E., Tanaka, K.: New Features for Continuous Exploratory Landscape Analysis based on the SOO Tree. In: Friedrich, T., Doerr, C., Arnold, D.V. (eds.) Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms, FOGA. pp. 72–86. ACM (2019)

[6] Fawcett, C., Vallati, M., Hoos, H.H., Gerevini, A.E.: Competitions in AI – Robustly Ranking Solvers Using Statistical Resampling (Aug 2023)

[7] Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene Selection for Cancer Classification using Support Vector Machines. Machine learning **46**, 389–422 (2002)

[8] Hansen, N., Brockhoff, D., Mersmann, O., Tusar, T., Tusar, D., ElHara, O.A., Sampaio, P.R., Atamna, A., Varelas, K., Batu, U., Nguyen, D.M., Matzner, F., Auger, A.: COmparing Continuous Optimizers: numbbo/COCO on Github (Mar 2019). `https://doi.org/10.5281/zenodo.2594848`, `https://doi.org/10.5281/zenodo.2594848`

[9] Hansen, N., Finck, S., Ros, R., Auger, A.: Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions. Research Report RR-6829, INRIA (2009), `https://hal.inria.fr/inria-00362633`

[10] Jankovic, A., Popovski, G., Eftimov, T., Doerr, C.: The Impact of Hyper-Parameter Tuning for Landscape-Aware Performance Regression and Algorithm Selection. In: Proceedings of the 23rd Annual Conference on Genetic and Evolutionary Computation (GECCO). pp. 687–696 (2021)

[11] Kerschke, P., Hoos, H.H., Neumann, F., Trautmann, H.: Automated Algorithm Selection: Survey and Perspectives. ECJ **27**, 3 – 45 (2019)

[12] Kerschke, P., Trautmann, H.: Automated Algorithm Selection on Continuous Black-Box Problems by Combining Exploratory Landscape Analysis and Machine Learning. Evolutionary Computation **27**(1), 99–127 (2019), pMID: 30365386

[13] Kerschke, P., Trautmann, H.: Comprehensive Feature-based Landscape Analysis of Continuous and Constrained Optimization Problems using the R-package flacco. Applications in Statistical Computing: From Music Data Analysis to Industrial Quality Improvement pp. 93–123 (2019)

[14] Kostovska, A., Jankovic, A., Vermetten, D., de Nobel, J., Wang, H., Eftimov, T., Doerr, C.: Per-run Algorithm Selection with Warm-starting using Trajectory-based Features. In: Parallel Problem Solving from Nature – PPSN XVII: 17th International Conference, PPSN, 2022. pp. 46–60. Springer (2022)

[15] Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., Rudolph, G.: Exploratory Landscape Analysis. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation (GECCO). pp. 829–836 (2011)

[16] Muñoz, M.A., Smith-Miles, K.: Generating New Space-Filling Test Instances for Continuous Black-Box Optimization. Evolutionary Computation (ECJ) **28**(3), 379–404 (2020)

[17] Petelin, G., Cenikj, G.: How Far Out of Distribution Can We Go With ELA Features and Still Be Able to Rank Algorithms? In: 2023 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 341–346 (2023)
[18] Petelin, G., Cenikj, G., Eftimov, T.: TLA: Topological Landscape Analysis for Single-Objective Continuous Optimization Problem Instances. In: 2022 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1698–1705 (2022)
[19] Prager, R.P., Seiler, M.V., Trautmann, H., Kerschke, P.: Towards Feature-Free Automated Algorithm Selection for Single-Objective Continuous Black-Box Optimization. In: 2021 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE (2021)
[20] Prager, R.P., Seiler, M.V., Trautmann, H., Kerschke, P.: Automated Algorithm Selection in Single-Objective Continuous Optimization: A Comparative Study of Deep Learning and Landscape Analysis Methods. In: Parallel Problem Solving from Nature–PPSN XVII: 17th International Conference, PPSN 2022. Springer (2022)
[21] Prager, R.P., Trautmann, H.: Nullifying the Inherent Bias of Non-invariant Exploratory Landscape Analysis Features. In: International Conference on the Applications of Evolutionary Computation (Part of EvoStar). pp. 411–425. Springer (2023)
[22] Prager, R.P., Trautmann, H.: Pflacco: Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems in Python. Evolutionary Computation pp. 1–25 (07 2023)
[23] Renau, Q., Dréo, J., Doerr, C., Doerr, B.: Expressiveness and Robustness of Landscape Features. In: Proceedings of the 21st Annual Conference on Genetic and Evolutionary Computation (GECCO) Companion. pp. 2048 – 2051. ACM (2019)
[24] Rice, J.R.: The Algorithm Selection Problem. Advances in Computers **15** (1976)
[25] Seiler, M.V., Kerschke, P., Trautmann, H.: Deep-ELA: Deep Exploratory Landscape Analysis with Self-Supervised Pretrained Transformers for Single- and Multi-Objective Continuous Optimization Problems *(First revision submitted to ECJ)* (2024)
[26] Seiler, M.V., Prager, R.P., Kerschke, P., Trautmann, H.: A Collection of Deep Learning-based Feature-Free Approaches for Characterizing Single-Objective Continuous Fitness Landscapes. In: Proceedings of the 24th Annual Conference on Genetic and Evolutionary Computation (GECCO) (2022)
[27] Seiler, M.V., Rook, J., Heins, J., Preuß, O.L., Bossek, J., Trautmann, H.: Using Reinforcement Learning for Per-Instance Algorithm Configuration on the TSP. In: 2023 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 361–368. IEEE (2023)
[28] van Stein, B., Long, F.X., Frenzel, M., Krause, P., Gitterle, M., Bäck, T.: DoE2Vec: Deep-learning Based Features for Exploratory Landscape Analysis. In: Silva, S., Paquete, L. (eds.) Proceedings of the 25th Annual Conference on Genetic and Evolutionary Computation (GECCO) Companion. pp. 515–518. ACM (2023)
[29] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is All You Need. Advances in neural information processing systems **30** (2017)
[30] Vermetten, D., Ye, F., Bäck, T., Doerr, C.: MA-BBOB: A Problem Generator for Black-Box Optimization Using Affine Combinations and Shifts. CoRR **abs/2312.11083** (2023)