Optimizing solvers for real-world expensive black-box optimization with applications in vehicle design

Long, F.X.

Chapter 3

Landscape Analysis of Engineering Optimization Problems in Automotive Crash

As introduced in Section 1.1, we propose considering cheap-to-evaluate representative functions with similar optimization landscape characteristics for the fine-tuning of optimization configurations. Following this, we focus on having a better understanding about the optimization landscape characteristics of real-world expensive BBO problems w.r.t. some benchmark functions. In this context, two different approaches to capture the optimization landscape characteristics of BBO problems are first introduced in Section 3.1, namely based on the widely-used ELA features and the latent space information extracted using deep NN models. Next, our investigation is extended to analyze the function properties of a large set of BBOB problem instances in Section 3.2, since the BBOB functions are heavily considered throughout this thesis. This is followed by an analysis of the optimization problem classes for several real-world automotive crashworthiness optimization problems in Section 3.3. Lastly, Section 3.4 summarizes and concludes this chapter.

3.1 Capturing the Landscape Characteristics of Optimization Problems

To capture the optimization landscape characteristics of expensive BBO problems, we evaluate the potential of two different approaches. Firstly, we consider numerically quantifying the representative landscape characteristics of BBO problems using some ELA features, which has shown promising potential in previous work. As explained in detail in Section 3.1.1, the respective optimization problem classes of BBO problems can be conveniently identified based on their ELA features.

Apart from the hand-crafted ELA features, we investigate a feature-free alternative in capturing the optimization landscape of BBO problems using deep NN models, which we call *DoE2Vec*, as introduced in Section 3.1.2. Essentially, we propose an automated self-supervised representation learning approach to characterize the optimization landscape of BBO problems based on some latent space representations.

3.1.1 Landscape Features using ELA

An overview of our ELA-based approach proposed to quantify the optimization landscape characteristics of real-world expensive BBO problems is visualized in Figure 3.1, consisting of four steps in total. Essentially, by comparing the ELA features of a BBO problem against some test functions, such as the well-known BBOB functions, the optimization problem class of the BBO problem can be identified based on their similarity in terms of ELA features. In other words, a BBO problem belongs to the same optimization problem class as the test functions that have similar ELA features. Subsequently, these test functions with similar optimization landscape characteristics can be considered as representative functions for further usages, e.g., fine-tuning of algorithm configurations for the BBO problem. In the following, our approach is explained in detail:

Input: A set of DoE samples of the BBO problem instance to-be-solved is required as input. Unless otherwise stated, the Sobol' sampling strategy [122] is employed for the generation of DoE samples, as suggested in [109]. Nevertheless, any sampling strategy or a custom DoE can be utilized in practice.

Data pre-processing: The input DoE samples are pre-processed, where duplicated samples or samples with incomplete data are discarded, e.g., missing values due to interrupted FE simulation runs. To facilitate a fair comparison of ELA features between the BBO problem and test functions, the search space is rescaled

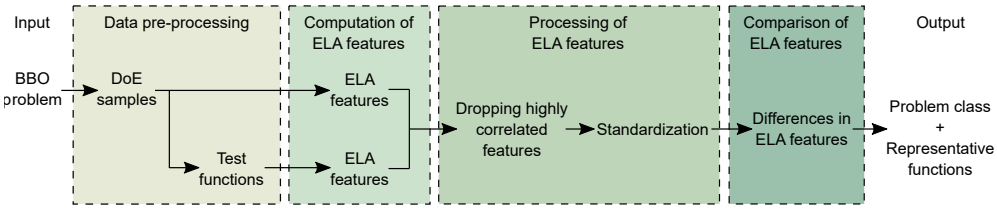

Figure 3.1: Overview of our approach to characterize the optimization landscape of real-world expensive BBO problems, requiring some DoE samples as input and consisting of four steps as marked with boxes. By comparing the ELA features of a BBO problem against those of some test functions, the optimization problem class of the BBO problem can be determined. On top of that, suitable representative functions can be identified based on the ELA features. Figure taken from [73].

to $[-5, 5]^d$ using Equation 3.1, which is commonly considered for the BBOB functions.

$$x_{new} = \frac{x_{orig} - a_{min}}{a_{max} - a_{min}} \cdot (b_{max} - b_{min}) + b_{min} \,, \tag{3.1}$$

where x_{orig} and x_{new} are the design variables before and after rescaling, a_{min} and a_{max} are the original minimum and maximum scale range, and b_{min} and b_{max} are the minimum and maximum scale range after rescaling. Following this, the same DoE samples and problem dimensionality are utilized to compute the objective values of test functions. To minimize potential inherent bias in ELA feature computation, the objective values are min-max normalized before the computation [100].

Computation of ELA features: ELA features are separately computed for the BBO problem and test functions using `pflacco` [98, 101], which was implemented based on `flacco` [59, 60]. Among the more than 300 available ELA features, we only consider features that fulfill the following requirements:

- Features that can be cheaply computed using the input DoE samples, without requiring additional resampling or function evaluations;
- Features that concern only the DoE samples $\mathcal{X}$ are neglected, e.g., some of the PCA features; and
- Features that are not informative about the problem landscape are ignored, e.g., concerning the computational costs.

Since an analytical form is not known and function evaluations are costly in real-world expensive BBO problems, computation of ELA features that require

resampling is particularly limited. Consequently, mainly the eight ELA feature classes summarized in Table 3.1* are considered in this thesis, which can be relatively cheaply computed without resampling.

To improve the reliability of ELA feature computation, particularly for a small DoE sample size in expensive BBO problems, and to minimize the impact of random sampling in ELA feature computation, we consider the mean ELA feature values computed based on a bootstrapping strategy. In brief, the ELA feature computation is repeated for 20 times using a subset of the input DoE samples, consisting of roughly 80% or $0.8 \cdot n$ samples that are randomly selected in each repetition. Eventually, the mean feature values are considered for further analysis. Furthermore, when the BBOB functions are considered as test functions, the ELA feature values are additionally averaged across the first 20 BBOB instances to improve their robustness. In cases where a feature computation fails, e.g., when the sample size is too small for computing the level set or linear model features, it will be skipped. Consequently, fewer ELA features will be computed for such a problem instance.

Processing of ELA features: Before the ELA features between the BBO problem and test functions are compared against each other, two feature processing steps are performed as follows:

1. Since many of the ELA features are highly correlated and redundant [109, 150], only a subset of all ELA features computed is considered for the comparison. Precisely, for each highly correlated feature pair based on Pearson's correlation coefficient (> 0.95), the feature that has a higher mean correlation with other remaining features is removed. Meanwhile, ELA features that have a constant value across all functions are automatically neglected; and
2. To ensure that the computed ELA features are within a comparable scale range, they are standardized by removing the mean and scaling to unit variance.

Both steps are first performed on the ELA features of the test functions, e.g., BBOB functions, and then applied to those of the BBO problem, using the same

*It is worth noting that, `flacco` was initially employed for the ELA feature computation during some early investigations in this thesis, which was later replaced by `pflacco` due to its higher compatibility with our optimization pipeline. As such, the ELA features computed might differ slightly between investigations. However, this variation does not significantly impact our results according to some preliminary testings.

Table 3.1: Brief descriptions of the ELA features from eight feature classes considered in this thesis, with the respective labels for feature classes and ELA features. Using `flacco` (in `R`), more level set features can be computed, as highlighted in gray color, which are not yet implemented in `pflacco` (in `Python`). Table taken from [73].

Feature class	**Description**	**ELA feature**
y-distribution (`ela_distr.*`)	Distribution of function values. 3 features	`skewness` `kurtosis` `number_of_peaks`
Level set (`ela_level.*`)	Measure the performance of different classification methods based on function value thresholds. 9 features (in `pflacco`) (or 18 features in `flacco`)	`mmce_lda_{10,25,50}` `mmce_qda_{10,25,50}` `mmce_mda_{10,25,50}` `lda_qda_{10,25,50}` `lda_mda_{10,25,50}` `qda_mda_{10,25,50}`
Meta-model (`ela_meta.*`)	Fitting quality of linear and quadratic models with and without interactions. 9 features	`lin_simple.{adj_r2,intercept}` `lin_simple.coef.{min,max,max_by_min}` `lin_w_interact.adj_r2` `quad_simple.{adj_r2,cond}` `quad_w_interact.adj_r2`
Dispersion (`disp.*`)	Comparison of dispersion between initial sample points and subset of points based on function value thresholds. 16 features	`ratio_mean_{02,05,10,25}` `ratio_median_{02,05,10,25}` `diff_mean_{02,05,10,25}` `diff_median_{02,05,10,25}`
NBC (`nbc.*`)	Comparison of distance between all sample points towards nearest points and nearest points with better function value. 5 features	`nn_nb.{sd_ratio,mean_ratio,cor}` `dist_ratio.coeff_var` `nb._fitness.cor`
PCA (`pca.*`)	Information based on PCA on initial sample points. 2 features	`expl_var_PC1.{cov_init,cor_init}`
Linear model (`limo.*`)	Measure the average coefficient vectors across multiple linear models. 4 features	`avg_length.{reg,norm}` `length.mean` `ratio.mean`
ICoFiS (`ic.*`)	Measure of smoothness, ruggedness, and neutrality of the landscape through random walk. 5 features	`h.max` `eps.{s,max,ratio}` `m0`

subset of ELA features as well as the same mean and variance for standardization. This is necessary to minimize potential information leakage, similar to the over-fitting problem in a ML context. Moreover, both steps are essential to improve the comparison results based on a distance-based metric in the next step.

Comparison of ELA features: Using the Euclidean distance metric and an equal weighting of all remaining ELA features, the similarity between the BBO problem and test functions w.r.t. ELA features can be quantified. Subsequently, the optimization problem class of the BBO problem can be identified based on its neighboring test functions with a small difference in ELA features.

Output: The landscape characteristics of the BBO problem in terms of ELA features and a set of neighboring test functions with similar optimization landscape characteristics are provided as output. These similar test functions can be potentially considered as representative functions for the BBO problem for further usages.

In our approach, the optimization landscape of real-world expensive BBO problems are characterized based on some ELA features computed using an initial set of DoE samples, without requiring additional costly function evaluations. While the effectiveness of ELA features could be potentially sacrificed, since they are dependent on factors like DoE sample size and problem dimensionality, as discussed in Section 2.3, this step is essential to facilitate an application of our approach for expensive BBO problems. Given that adding more DoE samples can be particularly limited in expensive BBO problems, we propose to consider an initial DoE sample size that is affordable w.r.t. real-world constraints, while the representative landscape characteristics can still be sufficiently captured. Based on our preliminary testing, for instance, a DoE of around $20 \cdot d$ samples seems to be an optimal trade-off for the automotive crashworthiness optimization problems investigated in this thesis (Chapter 5). Nonetheless, a closer investigation on this topic is necessary to have a better understanding.

Apart from considering some test functions with similar optimization landscape characteristics as representative functions, another alternative could be using some data-driven surrogate models trained on some DoE samples for the fine-tuning of algorithm configurations. While this approach seems to be much more straightforward, our approach has the advantage that optimization configurations are fine-tuned for a particular optimization problem class, rather than only for that particular problem instance, similar to a ML over-fitting problem. In other words, optimal configurations identified using our approach could perform equally well on other problem instances

that belong to the same optimization problem class. A similar research topic was briefly explored in [157].

3.1.2 DoE2Vec: A Feature-free Approach using Autoencoder

On the contrary to the hand-crafted ELA features, here we investigate a feature-free approach to capture the optimization landscape characteristics of real-world expensive BBO problems. Principally, informative latent representation vectors are extracted from some DoE samples using deep NN models, e.g., Variational Autoencoder (VAE), and thus, the name DoE2Vec. Subsequently, the low-dimensional representations of the optimization landscape of BBO problems can be captured in an automated, generic, and unsupervised manner. Compared to the classical ELA approach, DoE2Vec has the following advantages, namely:

- Domain knowledge in optimization landscape analysis w.r.t. feature engineering and feature selection is not strictly required;
- Bias toward particular landscape characteristics in the latent representations, if any, can be minimized; and
- Independent of the sampling method.

In the following, a brief introduction to the VAEs employed in DoE2Vec is presented, followed by an overview of the DoE2Vec workflow, and lastly some result discussions and limitations of DoE2Vec are provided. To the best of our knowledge, a similar approach to capture the optimization landscape characteristics of BBO problems using VAEs has not yet been attempted.

Variational Autoencoder in DoE2Vec

Similar to a standard Autoencoder (AE), the VAEs employed in DoE2Vec have a similar architecture, consisting of three components, namely an encoder, a hidden layer, also known as bottleneck, and a decoder, as shown in Figure 3.2. In DoE2Vec, the objective values $\mathcal{Y}$ of some DoE samples are provided as input for a VAE. Essentially, the encoder projects the input space $\mathcal{Y}$ to a representative feature space $\mathcal{H}$, i.e., $v\colon \mathcal{Y} \to \mathcal{H}$, while the decoder transforms the feature space back to the input space $w\colon \mathcal{H} \to \hat{\mathcal{Y}}$ [19]. To better capture crucial representations of the input space, a VAE is constructed using an input layer and an output layer of the same dimension $d_{\mathcal{Y}}$, but a bottleneck

layer of lower dimension $d_{\mathcal{H}}$, i.e., $d_{\mathcal{H}} < d_{\mathcal{Y}}$. By encoding the latent space as a distribution using a mean and variance layer, along with a sampling method, the latent space can be properly regularized to provide more meaningful latent representations.

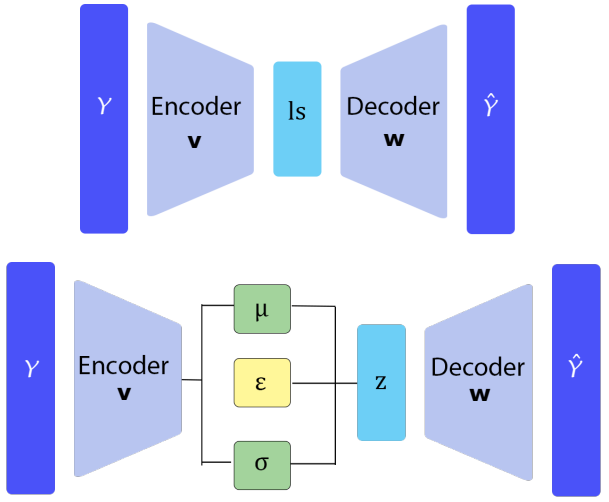

Figure 3.2: Example of standard AE (*top*) and VAE (*bottom*) considered in DoE2Vec. In comparison to an AE, the latent space of a VAE is encoded as a distribution using a mean and variance layer, together with a sampling method. The latent space z is defined by $z = \mu + \sigma \cdot \epsilon$, where ϵ denotes the sampling using a mean μ and a variance σ. Figure taken from [139].

During the unsupervised training process, a VAE attempts to optimally reconstruct the original input space $\mathcal{Y}$, by minimizing the reconstruction error $\mathcal{L}_{\text{VAE}}$ defined in Equation 3.2, which can be considered as a way to estimate the quality of learned representations.

$$\begin{aligned} \mathcal{L}_{\text{VAE}}(\mathcal{Y}, \hat{\mathcal{Y}}) &= \beta \cdot \mathcal{L}_{\text{KL}}(\mathcal{Y}, \hat{\mathcal{Y}}) + \mathcal{L}_{\text{MSE}}(\mathcal{Y}, \hat{\mathcal{Y}}) \,, \\ \mathcal{L}_{\text{KL}}(\mathcal{Y}, \hat{\mathcal{Y}}) &= \frac{1}{2} \sum_{i=1}^{|\mathcal{Y}|} (\exp(\sigma_i) - (1 + \sigma_i) + \mu_i^2) \,, \\ \mathcal{L}_{\text{MSE}}(\mathcal{Y}, \hat{\mathcal{Y}}) &= \sum_{y \in \mathcal{Y}, \hat{y} \in \hat{\mathcal{Y}}} (y - \hat{y})^2 \,, \end{aligned} \tag{3.2}$$

where a weighting factor β is included to control the trade-off between a regularization term and a reconstruction error, μ denotes the mean, and σ denotes the variance latent layers in a VAE model. In DoE2Vec, the regularization term is expressed using

the Kullback–Leibler (KL) divergence $\mathcal{L}_{\text{KL}}$, while the Mean Squared Error (MSE) $\mathcal{L}_{\text{MSE}}$ is used for the reconstruction error. Moreover, the VAEs employed in DoE2Vec have an architecture of seven fully connected layers in total, namely:

- The encoder is composed of four fully connected layers, starting with the input layer size d_y, which is equal to the DoE sample size n. This is followed by two hidden layers with a size of $n/2$ and $n/4$ respectively. For the mean and log variance of the latent space, the latent size ls ($ls < n/4$) is assigned; and
- The decoder is composed of three fully connected layers, having a size of $n/4$, $n/2$, and n for the final output layer.

Furthermore, the Rectified Linear Unit (ReLU) activation function is assigned to the hidden layers, while the sigmoid activation function is used for the final output layer in decoder.

Using a grid search approach, the impact of two parameters on the model performance is briefly analyzed, namely the latent size ls and the KL loss weight β, as shown in Figure 3.3. Based on our analysis, a combination of either 24 or 32 latent size (expressed as `VAE-24` or `VAE-32`) and β of 0.001 seems to be a good compromise for minimizing the loss functions. Subsequently, these parameters are considered in our following investigations. While a fine-tuning of the model parameters could potentially further improve the model performance, this is not considered here, since we are not focusing on reconstructing the optimization landscape. Instead, we consider the reconstruction as a mean to evaluate the quality of learned representations.

The complete specification of different VAEs is available in [138] and the pre-trained models are available in our repository [137].

Workflow of DoE2Vec

Similarly to the ELA-based approach illustrated in Figure 3.1, the workflow of DoE2Vec is visualized in Figure 3.4. Instead of the BBOB functions, the function generator from Section 4.1 is integrated in DoE2Vec to generate a diverse set of test functions in terms of optimization landscape. Precisely, test functions belonging to different optimization problem classes can be randomly generated using this function generator, which we call *random* functions or Randomly Generated Function (RGF). A detailed explanation of RGFs is provided in Section 4.1.

In the first step, the DoE samples $\mathcal{X}$ of a BBO problem is taken as input, which is re-scaled to $[-5, 5]^d$. Using the random function generator, a preferably large set of RGFs

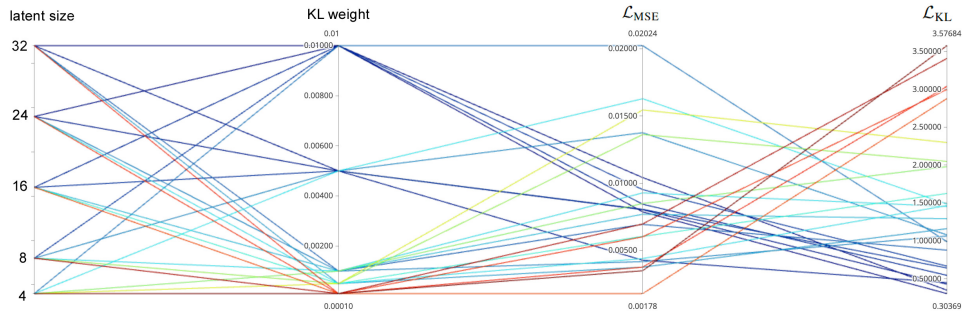

Figure 3.3: Parallel coordinates plot of different latent sizes ls and KL loss weights β in relation to the regularization term $\mathcal{L}_{\text{KL}}$ and reconstruction error $\mathcal{L}_{\text{MSE}}$. Each color or line represents a combination of model parameters. Generally, conflict between both loss terms can be observed, where a parameter combination with a lower $\mathcal{L}_{\text{MSE}}$ tends to have a higher $\mathcal{L}_{\text{KL}}$, and vice versa. *From left to right:* Latent size, KL loss weight, $\mathcal{L}_{\text{MSE}}$, and $\mathcal{L}_{\text{KL}}$.

can be generated and evaluated using the same DoE samples. Next, the corresponding objective values $\mathcal{Y}$ are rescaled to $[0, 1]$, which are then provided for the training of VAEs. Subsequently, the trained VAEs can be employed to compute the latent space representations of the BBO problem. Using a distance-based metric like Euclidean distance, a set of RGFs similar to the BBO problem in terms of latent representations can be identified. Since the latent representations can be readily utilized, a feature processing step is not strictly necessary here, such as normalizing or eliminating highly correlated features.

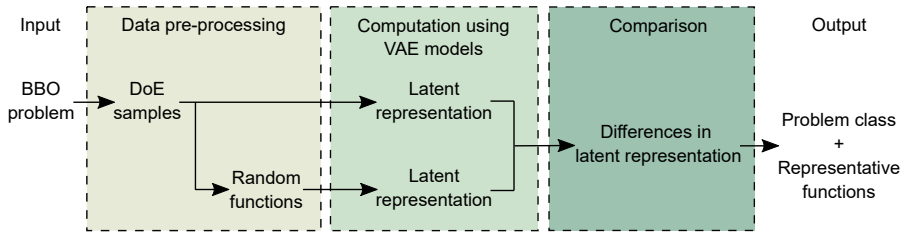

Figure 3.4: Overview of the workflow of DoE2Vec. Basically, VAEs are employed to capture the latent representations of BBO problems, which are trained using a preferably large set of RGFs covering a variety of optimization problem classes.

Performance Assessment of DoE2Vec

To evaluate the performance of DoE2Vec in capturing representative latent representations of BBO problems, two investigations are performed, namely (i) visual inspection of the optimization landscapes and (ii) classification of the BBOB functions. In general, our investigations have the following setup, namely a DoE of 256 samples generated using Sobol' sampling and a set of 250 000 RGFs for the model training.

i. Visual comparison of optimization landscape: Firstly, we evaluate the potential of DoE2Vec in identifying RGFs that have optimization landscape characteristics similar to BBO problems. Since an analytical analysis can be challenging, our investigation instead focuses on visual inspection of the optimization landscape based on 24 BBOB functions (of the first instance) in 2-d. Precisely, the respective RGFs having the most similar latent space representations in terms of the Euclidean distance are identified for each BBOB function. As visually compared in Figure 3.5, a well-fitting RGF can be identified for most BBOB functions, even for highly nonlinear functions like F22. Nevertheless, a clear visual difference in the optimization landscape between some BBOB functions and RGFs can be observed, such as F16 and F24. This indicates that the representative landscape characteristics of such complex functions are insufficiently captured using our DoE2Vec approach.

ii. Classification of the BBOB functions: Secondly, another attractive application of DoE2Vec is to classify optimization problems according to their high-level function properties, such as multi-modality, global structure, and funnel structure [119]. Since these high-level properties often determine the complexity of optimization problems, an appropriate capturing of these properties are crucial for algorithm configuration fine-tuning purposes. For the multiclass classification tasks, a standard RF model with 100 trees from `sklearn` [94] is employed. Using the following setup, we compare the effectiveness of ELA features and/or latent representations computed using different AEs or VAEs for the multiclass classification tasks:

- Standard AE using the same architecture as VAE models, except that the latent space is now a single dense layer. Subsequently, the performance of two AEs and two VAEs is evaluated, consisting of `AE-24`, `AE-32`, `VAE-24`, and `VAE-32`;

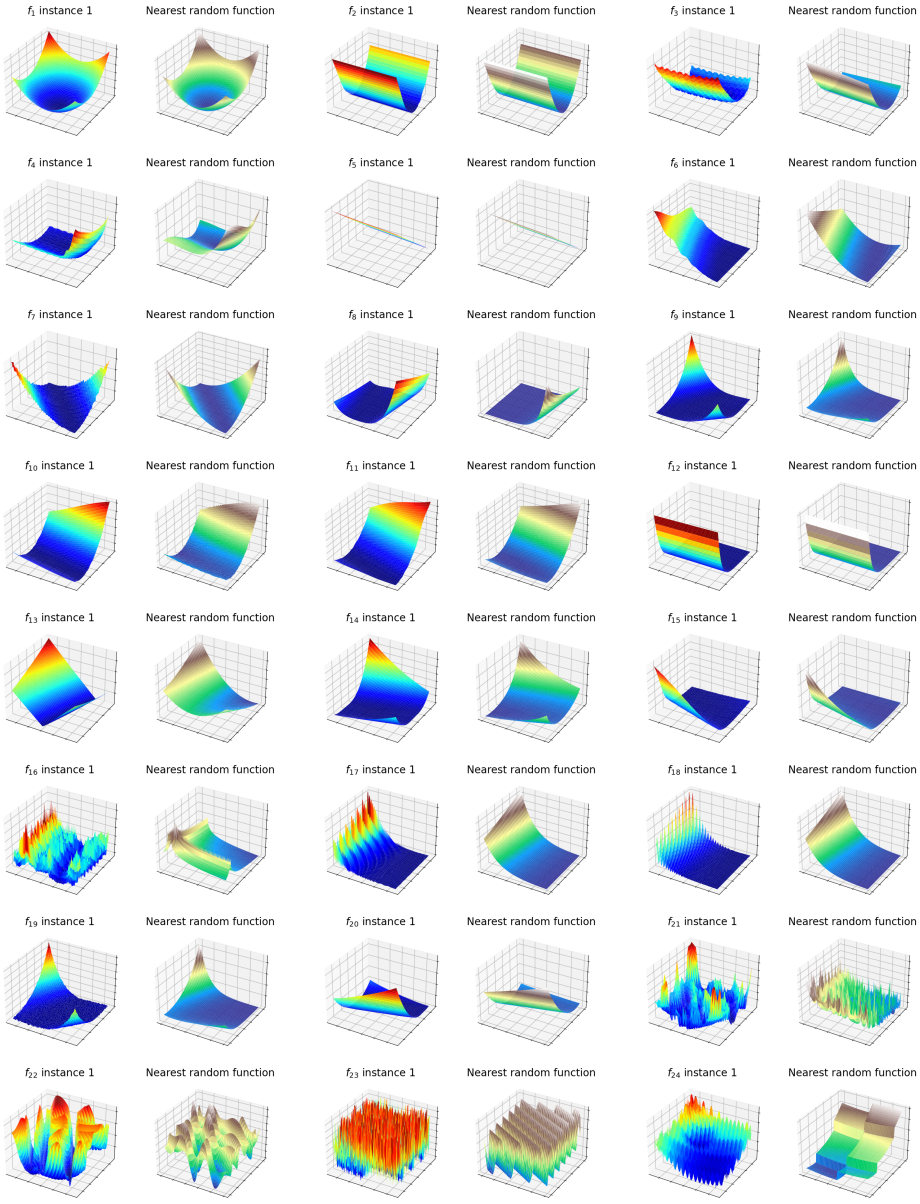

Figure 3.5: Pairs of 24 BBOB functions (*left side*) and their respective nearest RGF (*right side*) identified based on some latent space representations computed using `VAE-24`. Figure taken from [139].

- The classical ELA features, which are specifically designed for such function classification tasks; and
- A combination of ELA features and latent representations captured using `VAE-32` for investigating the complementary effect of different feature sets.

The classification results based on macro F1 scores using different approaches are summarized in Table 3.2. Generally, a good performance can be achieved by DoE2Vec using both AE and VAE, particularly in low problem dimensionality. On the other hand, the ELA approach constantly outperforms DoE2Vec, especially in classifying the global structure and multimodal landscapes, which however comes as no surprise. Fascinatingly, the classification performance can be further improved using a combination of ELA and DoE2Vec, revealing the complementary effect of both feature sets. In other words, the latent representations learned using DoE2Vec can be employed next to the ELA features, for a better handling of classification tasks.

Table 3.2: Classification results based on macro F1 scores averaged across 10 repetitions using a standard RF model. The RF model is first trained using the feature representations (AE, VAE, ELA, and ELA combined with `VAE-32`) of the first 100 instances for each BBOB function, and then validated using instance 101 to 120. *The results using PCA, reduced Multiple Channel (rMC), and Transformer are directly taken from previous work in [119], using an identical experimental setup, but without repetition. Table taken from [139].

d	Task	AE-24	AE-32	VAE-24	VAE-32	ELA	PCA*	rMC*	Transformer*	ELA-VAE-32
2	multimodal	0.875	0.849	0.877	0.856	0.984	0.994	0.971	0.991	**0.991**
	global struct.	0.903	0.904	0.902	0.889	0.983	0.992	0.965	0.991	**0.998**
	funnel	0.985	0.974	0.956	0.978	**1.000**	0.999	0.995	1.000	**1.000**
5	multimodal	0.908	0.903	0.880	0.889	0.963	0.897	0.947	0.991	**0.998**
	global struct.	0.838	0.828	0.810	0.793	**1.000**	0.807	0.859	0.978	**1.000**
	funnel	**1.000**	**1.000**	0.996	0.991	**1.000**	0.990	0.989	1.000	**1.000**
10	multimodal	0.877	0.813	0.844	0.838	**1.000**	0.839	0.952	0.974	**1.000**
	global struct.	0.794	0.737	0.783	0.745	0.902	0.774	0.911	0.963	**0.991**
	funnel	0.998	0.993	0.997	0.993	0.972	0.977	0.991	**1.000**	0.997
20	multimodal	0.726	0.722	0.700	0.694	0.970	-	-	-	**0.991**
	global struct.	0.689	0.621	0.606	0.626	0.972	-	-	-	**0.997**
	funnel	0.993	0.982	0.985	0.982	**1.000**	-	-	-	**1.000**

Limitations of DoE2Vec

Based on our investigations, DoE2Vec has shown motivating potential in capturing representative latent representations of BBO problems, which could be potentially

used to identify representative functions for further usages. The complete result discussions are available in [139]. Instead of replacing the classical ELA features, we propose to extend the ELA feature set with the latent space representations computed using DoE2Vec. Meanwhile, the following limitations of DoE2Vec have been identified, namely:

1. The latent space representations computed using VAEs are a black-box that are extremely difficult to be interpreted;

2. DoE2Vec is scale-invariant, but not rotation- or translation-invariant, which could be improved by using other loss functions; and

3. In cases where no pre-trained VAE is available, e.g., when using a custom DoE, the model must be trained from scratch. Depending on the DoE sample size and number of RGFs, the training of VAEs could be relatively time-consuming compared to the ELA feature computation.

Following this, substantial work is necessary to significantly improve the performance and reliability of DoE2Vec. Thus, this topic is not pursued any further in this thesis. Instead, we focus on the ELA-based approach proposed in Section 3.1.1 to capture the optimization landscape characteristics of real-world expensive BBO problems.

3.2 Analyzing the Function Properties of BBOB Instances

To identify appropriate representative functions for real-world expensive BBO problems, their optimization landscape characteristics are compared against those of some test functions, as proposed in Section 3.1.1. In this context, the BBOB functions could potentially serve as the best candidate for this purpose, and thus, an in-depth understanding is necessary. Commonly in previous work, only the first few BBOB instances are considered for investigations. Nonetheless, this could significantly impact the investigation outcomes, especially if these instances are not representative of the overall space of instances. Following this, we analyze a large set of BBOB instances to better understand the representativeness of different BBOB instances w.r.t. their underlying function properties. Precisely, we investigate the properties of BBOB instances in terms of (i) ELA features in Section 3.2.1, (ii) the performance of optimization algorithms in Section 3.2.2, and (iii) the global function properties in Section 3.2.3.

Here, we focus on the first 500 instances of the BBOB functions in 5-d and 20-d. Furthermore, a confidence level of 99% is considered in all statistical comparisons, i.e., a null hypothesis is rejected, if the p-value is lower than 0.01. Full descriptions of the experimental setup and results are available in [76, 77].

3.2.1 Landscape Characteristics based on ELA Features

Our investigation begins with an analysis of the optimization landscape characteristics in terms ELA features. Using the ELA-based approach proposed in Section 3.1.1, a final set of 65 ELA features is computed, where some of the PCA features that only concern the DoE samples $\mathcal{X}$ are additionally included for a comprehensive analysis. To obtain the ELA feature distributions, a total of 100 DoEs having 1 000 samples each are generated using LHS for each BBOB instance, where the DoEs are identical across instances.

A. Distributions of ELA features: In the first step, the distributions of ELA features between different BBOB instances are compared using the pairwise two-sample Kolmogorov-Smirnov (KS) test [53], using the null hypothesis *the ELA distribution is similar between both problem instances*. This ends up with altogether $\frac{500 \cdot 499}{2} = 124\,750$ comparison pairs for each ELA feature. To mitigate the effect of multiple comparisons, the Benjamini-Hochberg (BH) correction method [11] is additionally applied. For a convenient interpretation, the results are aggregated across all BBOB instances. Figure 3.6 shows the average rejection rate of the null hypothesis, i.e., the fraction of tests that rejects the null hypothesis.

In 5-d, some ELA features clearly differ between the BBOB instances, particularly the `ela_meta.lin_model.intercept`. Nevertheless, this does not necessarily indicate that all instances should be considered different, since some ELA features are not invariant to scaling of the objective function, such as the linear model intercept [152]. On the other hand, barely any test rejections can be observed in some features, such as the PCA features. This is because they are primarily computed using the DoE samples $\mathcal{X}$, which are identical across instances. While the objective values $\mathcal{Y}$ can principally influence some of the PCA features, their impact is relatively marginal.

On a per-function basis, some commonalities in many of the ELA features can be noticed. Interestingly, no difference between instances for almost all ELA

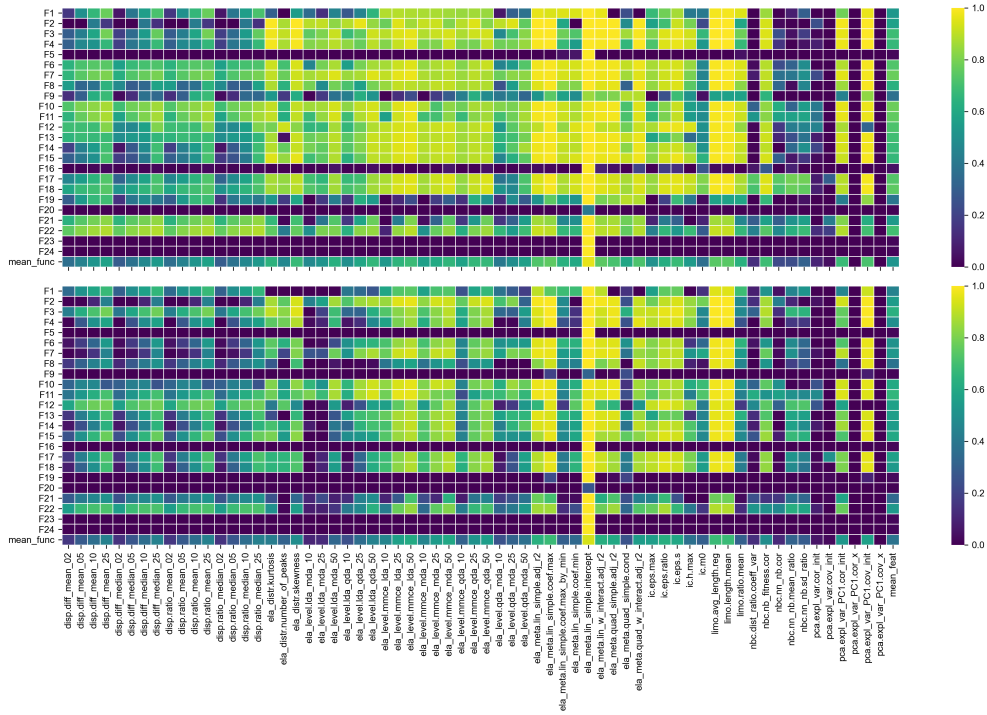

Figure 3.6: Average rejection rate of the null hypothesis *the distribution of ELA feature between instances is similar*, aggregated over 500 BBOB problem instances in 5-d (*top*) and 20-d (*bottom*). A lighter color represents a higher rejection rate. An extra row (*bottom*) is included for the average across all BBOB functions and an extra column (*right*) for the average across all ELA features. Figure taken from [76].

features can be observed in functions like F5, F16, F23, and F24. It is worthwhile to point out that even for a simple function like F1, many features differ between instances. Since translation is the only transformation applied to F1 [44], which uniformly at random moves the global optimum within $[-4, 4]^d$, the high-level landscape properties must be well preserved. While this is true as long as F1 is treated as an unconstrained problem, this is not the case in ELA, where samples are drawn within a bounded domain for feature computation. Following this, translating the function can significantly impact the low-level landscape features, which might explain the huge differences in ELA features across F1 instances.

Overall, a similar pattern can be observed in 20-d BBOB functions with a reduced magnitude. Moreover, functions like F9, F19, and F20 now barely show any statistical difference between instances.

B. Dimensionality reduction: Based on the standardized ELA features, i.e., by removing the mean and scaling to unit variance, the distributions of BBOB instances are projected to the 2-d ELA feature space using the t-distributed Stochastic Neighbor Embedding (t-SNE) approach [136], as shown in Figure 3.7. While it is clear that most instances of the same BBOB functions are clustered together, several BBOB instances are spread throughout the feature space, indicating that these outlying instances might be less similar compared to those clustering one. This is particularly noticeable in 5-d, where several functions are somewhat spread across the projected space. On the other hand, the BBOB function clusters seem to be much more stable in 20-d, matching our interpretations earlier in Figure 3.6. In fact, the differences between BBOB functions can be indeed easier detected in higher dimensionality based on ELA features, as shown in [111], which matches the more well-defined problem clusters in Figure 3.7.

C. Representativeness of BBOB instances 1–5: Based on the statistical analysis performed in Figure 3.6, we evaluate the representativeness of the first five BBOB instances that are commonly considered in literature in terms of ELA features. Instead of aggregating the rejections on a per-feature level, we do it now per-function. An instance can be considered as an outlier, and thus, non-representative, if the fraction of test rejections against other instances is high, while the overall fraction of pairwise test rejections is low. The average fraction of rejections across ELA features is visualized in Figure 3.8, where the rejection rates of the first five instances are highlighted. Generally, there is no obvious

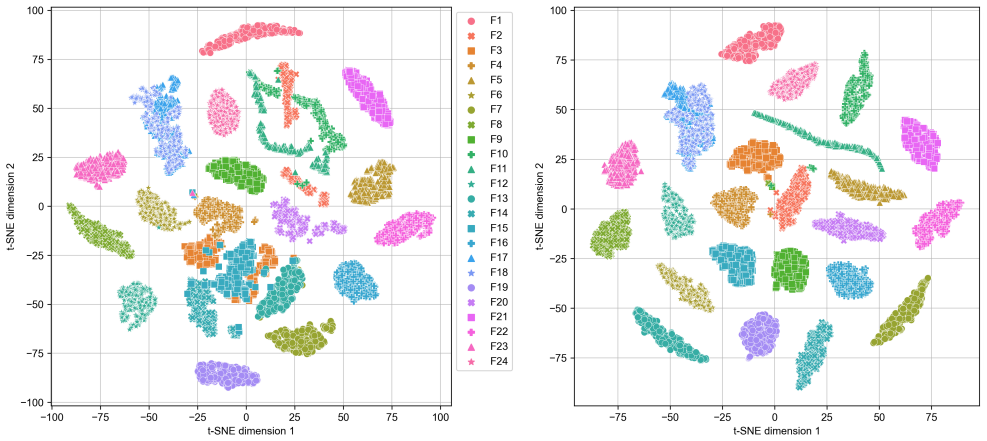

Figure 3.7: Projection of the high-dimensional ELA feature space (altogether 64 features, without `ela_meta.lin_simple.intercept`) to a 2-*d* visualization for the BBOB functions in 5-*d* (*left*) and 20-*d* (*right*) using the t-SNE approach. Each dot represents a BBOB instance and each color represents a BBOB function. Figure taken from [76].

case, where the five instances are all outliers. While some instances might have a slightly different rejection rate than the remaining instances, we could not conclude that the choice of selecting these five would be any better or worse than other instances of the same function.

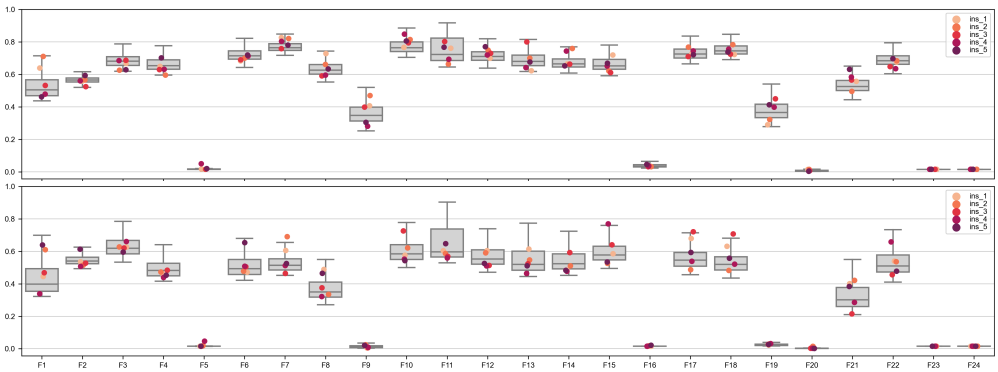

Figure 3.8: Average fraction of rejections of pairwise tests between one instance and each of the remaining ones, aggregated over all ELA features from Figure 3.6. The first five instances of each function are highlighted, while the boxplots (*gray*) show the distribution for all 500 instances in 5-*d* (*top*) and 20-*d* (*bottom*). Figure taken from [76].

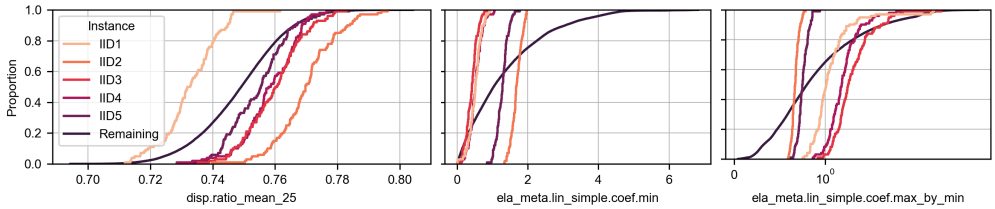

Figure 3.9: ECDF curves of normally (*left*) and non-normally distributed (*middle* and *right*) ELA features for F1 in 5-d, showing instances 1–5 and all remaining instances. Figure taken from [76].

Apart from considering the overall representativeness based on the aggregation of all ELA features, we also analyze each feature in more detail. Using three ELA features of F1 in 5-d as an example, namely `disp.ratio_mean_25`, `ela_meta.lin_simple.coef.min`, and `ela_meta.lin_simple.coef.max_by_min`, where the pairwise statistical analysis shows a large number of rejections, the Empirical Cumulative Distribution Functions (ECDF) curves in Figure 3.9 show the differences in distribution of instances 1–5 against the remaining instances. Despite the differences between instances can be relatively large, there is no evidence to conclude that the first five instances would be less representative than any other instance set. Interestingly, while some ELA features are seemingly normally distributed, this is not the case for all features. In fact, based on our normality tests for each ELA feature distribution in [77], some features are *non-Gaussian*, such as the distribution and meta-model features.

3.2.2 Algorithm Performance across Instances

Apart from analyzing the ELA features, we also investigate the performance of optimization algorithms across different BBOB instances. In this investigation, we consider eight derivative-free optimization algorithms available in `Nevergrad` [104], namely DiagonalCMA (a variant of CMA-ES), DE, EMNA, NGOpt14, PSO, RS, RCobyla, and SPSA, using a budget of 10 000 function evaluations and 50 repetitions. In this context, the algorithm performances are evaluated based on the best-found solution achieved after 1 000 and 10 000 evaluations on 500 BBOB instances in 5-d. To determine whether there are significant differences in algorithm performance between instances, the Mann-Whitney U (MWU) test with the null hypothesis *the algorithm performances are similar across instances* is employed, along with the BH correction

method. On top of this pairwise testing, we additionally consider a one-vs-all comparison using the same procedure, where the algorithm performance of a selected instance is compared against the remaining 499 instances.

Rather than the absolute objective values, we consider the relative performance measure in our investigation, i.e., precision from the global optimum. Subsequently, it is to be expected that RS is invariant across instances, as observed in Figure 3.10. In general, most of the algorithms have a stable performance on all BBOB functions, which mostly matches the results from Figure 3.6. On the opposite side, SPSA shows differences in performance between instances, particularly for F1, F5, F19, F20, F23, and F24. This indicates that SPSA is not invariant to the transformations implemented for the BBOB instance generation, matching with the observation that SPSA displays clear structural bias in [144].

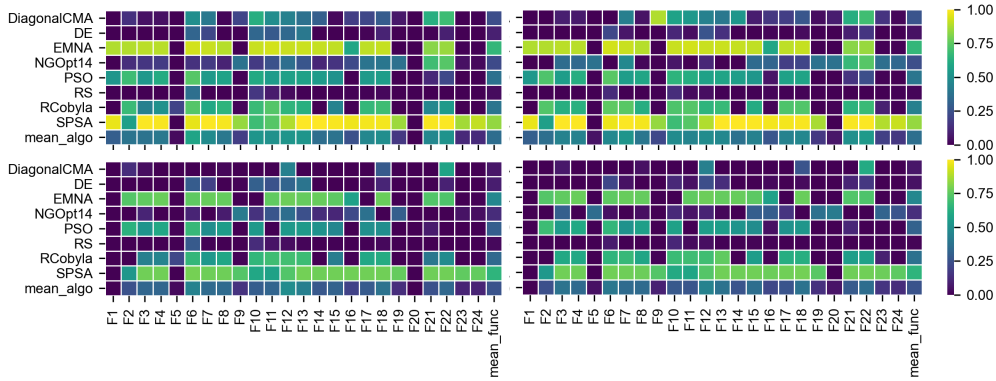

Figure 3.10: Average rejection rate of the null hypothesis *the algorithm performances are similar across instances*, aggregated over BBOB instances per function. The optimization results based on pairwise (*top row*) and one-vs-all (*bottom row*) comparisons are shown, using 1 000 (*left column*) and 10 000 (*right column*) function evaluations. The average values are shown in the last column and last row of each figure. Figure taken from [76].

While some algorithms, specifically DiagonalCMA and DE, are expected to be invariant to the transformations used for the BBOB instance generation, this assumption does not seem to hold for some BBOB functions like F12. This indicates that the instances lead to statistically different performances of these algorithms, which might be explainable considering that these algorithms treat the optimization problems as being box-constrained, while the BBOB function transformations assume that the domain is unconstrained [43]. Furthermore, while the algorithms might be invariant to rotation and transformation, applying these mechanisms can have an effect on the initializa-

tion, which eventually impacts the algorithm performances [143]. This is intended for the BBOB suite, since it is stated that *"If a solver is translation invariant (and hence ignores domain boundaries), this [running on different instances] is equivalent to varying the initial solution"* [43]. While this is true for unconstrained optimization, it is not as straightforward for box-constrained problems like the BBOB functions, since changing the initialization method could significantly influence algorithm behavior.

3.2.3 Global Function Properties

For most BBOB functions, the transformation mechanism generally consists of rotations and translations, which are applied differently to preserve the high-level function properties [44]. Nonetheless, the impact of transformation processes on the low-level features of BBOB functions can not always be as easily interpreted. Consequently, the differences between instances of each function are influenced by the associated transformation procedure, resulting in some functions being much more stable than others.

One aspect of the BBOB instances that is being treated differently is the location of the global optimum. For most of the BBOB functions, the location of the global optimum is uniformly sampled in $[-4, 4]^d$ by construction, where the optimum is moved from the default location $\mathbf{0}^d$ using translation. Nevertheless, a different procedure is used in some BBOB functions, such as F5. In Figure 3.11, the true location of the global optima for the first 500 BBOB instances in 2-d is visualized. While the distribution pattern of the global optimum for most of the BBOB functions is similar to F1, the following exceptions can be noticed:

- The asymmetric pattern for F4 is due to the fact that the even coordinates are being used differently than the odd ones by construction;
- For F8, a scaling transformation is applied before the final translation, resulting in the optimum being confined to a smaller space; and
- For other functions, namely F9, F19, F20, and F24, the problem construction requires a different setup, and thereby, the optima are distributed differently.

3.2.4 Summary

Based on our analysis, statistically significant differences between problem instances in terms of ELA features can be observed in some BBOB functions, which seemingly

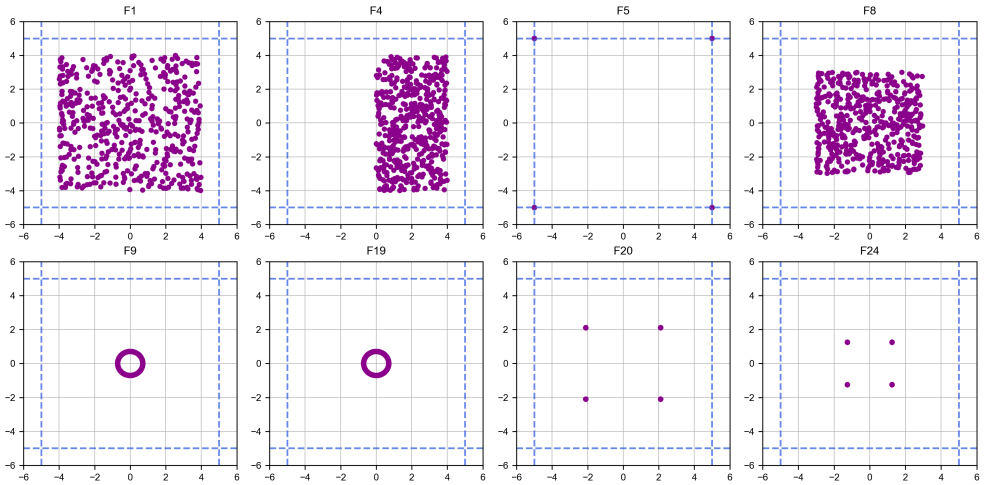

Figure 3.11: Distributions of the global optima for 500 instances of selected BBOB functions in 2-d. The remaining BBOB functions not shown here have a distribution pattern similar to F1, refer to [77]. Each violet dot represents the optimum location of a BBOB instance in the commonly used search space $[-5, 5]^2$, marked using dashed lines. Figure taken from [76].

lessens with increasing dimensionality. Following this, care should be taken when relying on ELA features to represent instances, e.g., for landscape-aware HPO, since the choice of BBOB instances could potentially lead to a different result.

Regarding the performance of optimization algorithms, only RS is close to being fully invariant across different BBOB instances. Meanwhile, differences in performance between instances of the same function can be observed for algorithms like CMA-ES, which are typically considered rotation invariant. This might be related to the impact of instance transformations on the effectiveness of algorithm initialization [143], or it might be due to the fact that we consider the problems to be box-constrained, which seemingly invalidates the assumptions considered in the transformation mechanisms.

Lastly, differences in the distribution pattern of the global optimum between BBOB instances can be observed as well. While the optimum location is confirmed to be uniform at random in $[-4, 4]^d$ for most BBOB functions, this is not always the case, since some problem formulations seemingly require different transformation mechanisms.

3.3 Landscape Characteristics of Automotive Crashworthiness Optimization

To learn the optimization landscape characteristics of automotive crashworthiness optimization, we have collected 20 automotive crashworthiness optimization problems from previous vehicle development projects performed by BMW, a German premium automobile manufacturer. Consisting of different automotive crash scenarios, namely side crash, rear crash, roof crash, and frontal crash, as summarized in Table 3.3, the DoE samples were generated using the Modified Extensible Lattice Sequence (MELS) sampling available in `HyperStudy`, which is a sequential lattice space-filling DoE approach developed based on the Sobol' sequences.

Table 3.3: Summary of 20 automotive crash problem instances analyzed in this thesis, consisting of four different crash scenarios. Table taken from [73].

Problem instance	Scenario	Design variables	Sample size
Crash_1	Side crash	22	59
Crash_2	Side crash	22	309
Crash_3	Side crash	22	309
Crash_4	Side crash	16	150
Crash_5	Side crash	16	102
Crash_6	Side crash	16	132
Crash_7	Side crash	20	329
Crash_8	Side crash	20	330
Crash_9	Side crash	20	333
Crash_10	Side crash	18	530
Crash_11	Side crash	13	150
Crash_12	Side crash	14	99
Crash_13	Rear crash	12	180
Crash_14	Rear crash	12	259
Crash_15	Roof crash	19	100
Crash_16	Roof crash	17	107
Crash_17	Frontal crash	22	487
Crash_18	Frontal crash	20	246
Crash_19	Frontal crash	8	248
Crash_20	Frontal crash	8	246

Due to the expensive FE simulation runs, as is commonly in automotive crash problems, some of the DoE sample sizes might be too small in relation to the high dimensionality for a reliable ELA feature computation, particularly in `Crash_1`. However, adding more DoE samples is ruled out here, since we have no access to these FE simulations. For problem instances from the same crash scenario, they were mainly different in terms of vehicle models and load cases, e.g., different pole positions for

side crashes. Furthermore, the design variables were the thicknesses of different vehicle components to-be-optimized, e.g., the thicknesses of rocker panels in side crash scenarios. During these development projects, the quality of a vehicle design was evaluated using the following five objectives, which were measured and quantified as scalar FE outputs:

1. Mass (M): Weight of components;
2. Maximum force (F_{max}): Maximum impact force during crash;
3. Intrusion ($Intr$): Magnitude of inward structural deformation;
4. Energy absorption (EA): The amount of kinetic energy absorbed during crash; and
5. Rotation (Rot): Rotational deformation of components during crash. This metric was introduced to measure the average vertical deformation of FE nodes between inner and outer side of components.

Principally, the mass objective provides information about the vehicle weight, fuel consumption, and manufacturing costs, while the remaining objectives are about the structural performance of a vehicle design. Throughout this thesis, we also refer to these objectives as *crash functions*. Nonetheless, depending on the purpose of each vehicle development project, not all five objectives were always considered, and therefore, not all of them were available for our study. In fact, out of the maximum of $20 \times 5 = 100$ potentially available ones, only a total of 48 crash functions were available.

3.3.1 Optimization Problem Classes

Using our approach proposed in Section 3.1.1, the ELA features of automotive crash problems are computed and compared against those of the BBOB functions. The differences in ELA features are visualized using agglomerative hierarchical clustering approach [91], where problems are clustered together in a bottom-up fashion. Precisely, this approach starts with each problem as a cluster and progressively merges clusters until one large cluster is left, consisting of all problems. Correspondingly, we consider the Ward's method [154] as a linkage criterion for the cluster merging strategy, by minimizing the within-cluster variance. In other words, clusters are selected for merging based on the smallest increase in the within-cluster sum of squared error

after merging, which is proportional to the Euclidean distance. In such way, the problem class of automotive crash problems can be easily identified based on the BBOB functions within the same cluster.

The clustering results are visualized in Figure 3.12, using the average pairwise distance between different BBOB functions as our reference in determining to which extent a crash function is similar to its neighboring BBOB functions. Generally, we observe that many of the crash functions are rather separated from the BBOB functions, especially the intrusion function in `Crash_8` and `Crash_9`, and the maximum force function in `Crash_16`. The extremely large Euclidean distance of the intrusion function in `Crash_8` is primarily due to the standardized `pca.expl_var_PC1.cov_init` feature, indicating that our standardization approach for ELA features using the mean and variance from BBOB functions might not be appropriate for all crash functions. The fact that some crash functions, such as the rotation function in `Crash_6`, are clustered in the same group as F16 and F23 suggests that they could have similar landscape characteristics, e.g., highly rugged and repetitive landscape. As expected, all mass functions are clustered in the same group as F5, since mass is linearly dependent on the thicknesses of vehicle components. Based on the clustering patterns, we observe that many automotive crash problems are indeed different from the BBOB functions in terms of landscape characteristics, revealing that these automotive crash problems belong to problem classes other than those covered by the BBOB suite. Consequently, the BBOB functions might be insufficiently representative for these automotive crash problems.

For a better understanding of the clustering results, each of the ELA features is closely examined, using `Crash_2` as a representative example due to its high dimensionality and sufficiently large DoE sample size. As visualized in Figure 3.13, several ELA features show remarkable differences in feature values between the crash functions and BBOB functions, e.g., `ela_distr.kurtosis` and `ela_level.mmce_qda_10`. For a fair evaluation, the distribution of ELA features between the crash functions and BBOB functions are compared using the two-sample KS test with the null hypothesis *the distribution of ELA features between them is similar*. Since the null hypothesis is rejected for some ELA features with a confidence level of 95%, the distribution of these ELA features is indeed different between the crash functions and BBOB functions, which might explain the separation in the clustering patterns observed previously.

For a convenient interpretation, the statistical comparison of ELA feature distributions between the crash functions and BBOB functions for all 20 automotive crash problems is summarized in Figure 3.14. While no obvious ELA feature has a com-

3.3 Landscape Characteristics of Automotive Crashworthiness Optimization

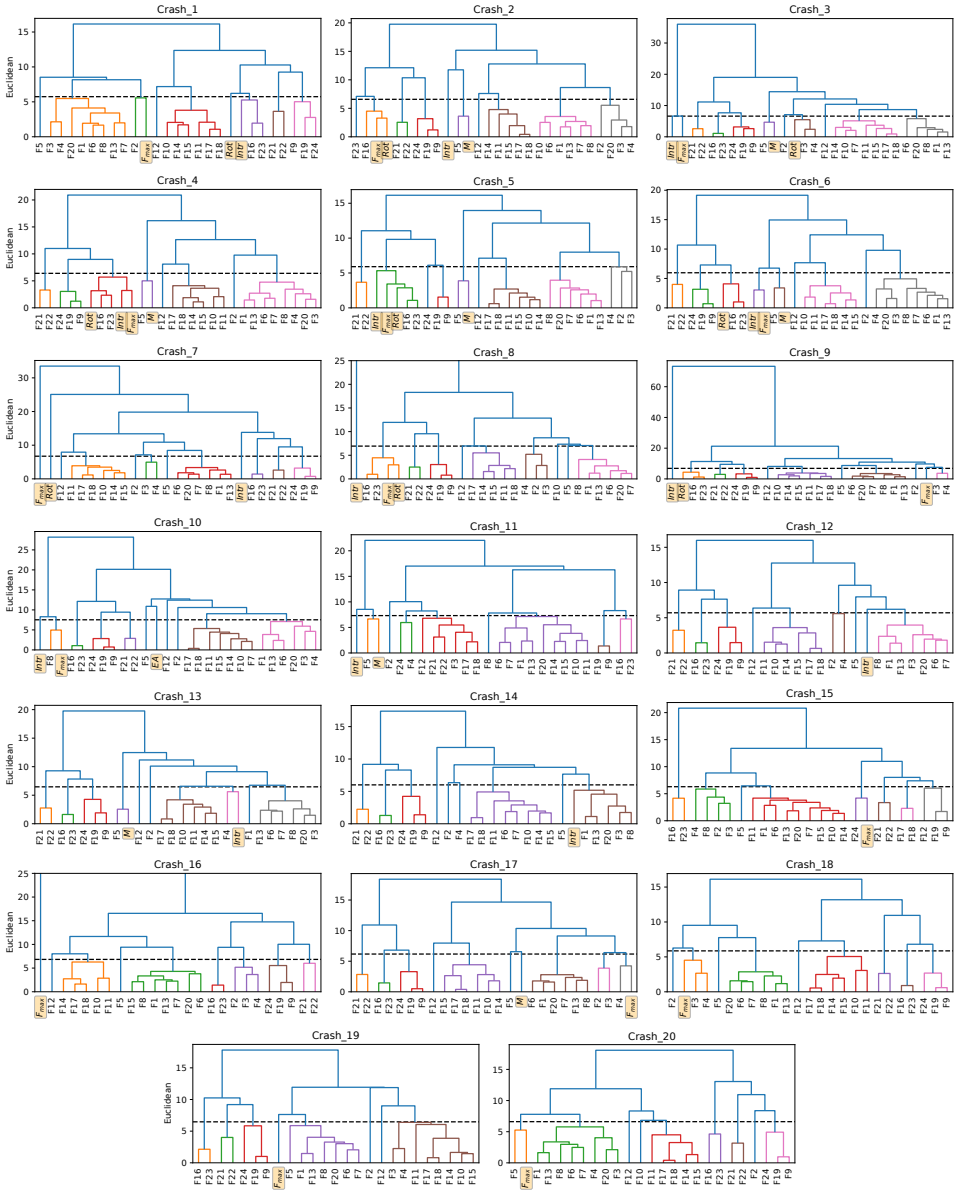

Figure 3.12: Clustering patterns of the crash functions and 24 BBOB functions for all 20 automotive crash problems, where the labels of crash functions are highlighted in orange color. The reference Euclidean distance is marked with a dashed line, and clusters below it are assigned with different colors. The intrusion function in `Crash_8` has a Euclidean distance of around 700 to the main cluster, and the maximum force function in `Crash_16` has a distance of around 300, which are cut-off due to visualization purposes. Figure taken from [73].

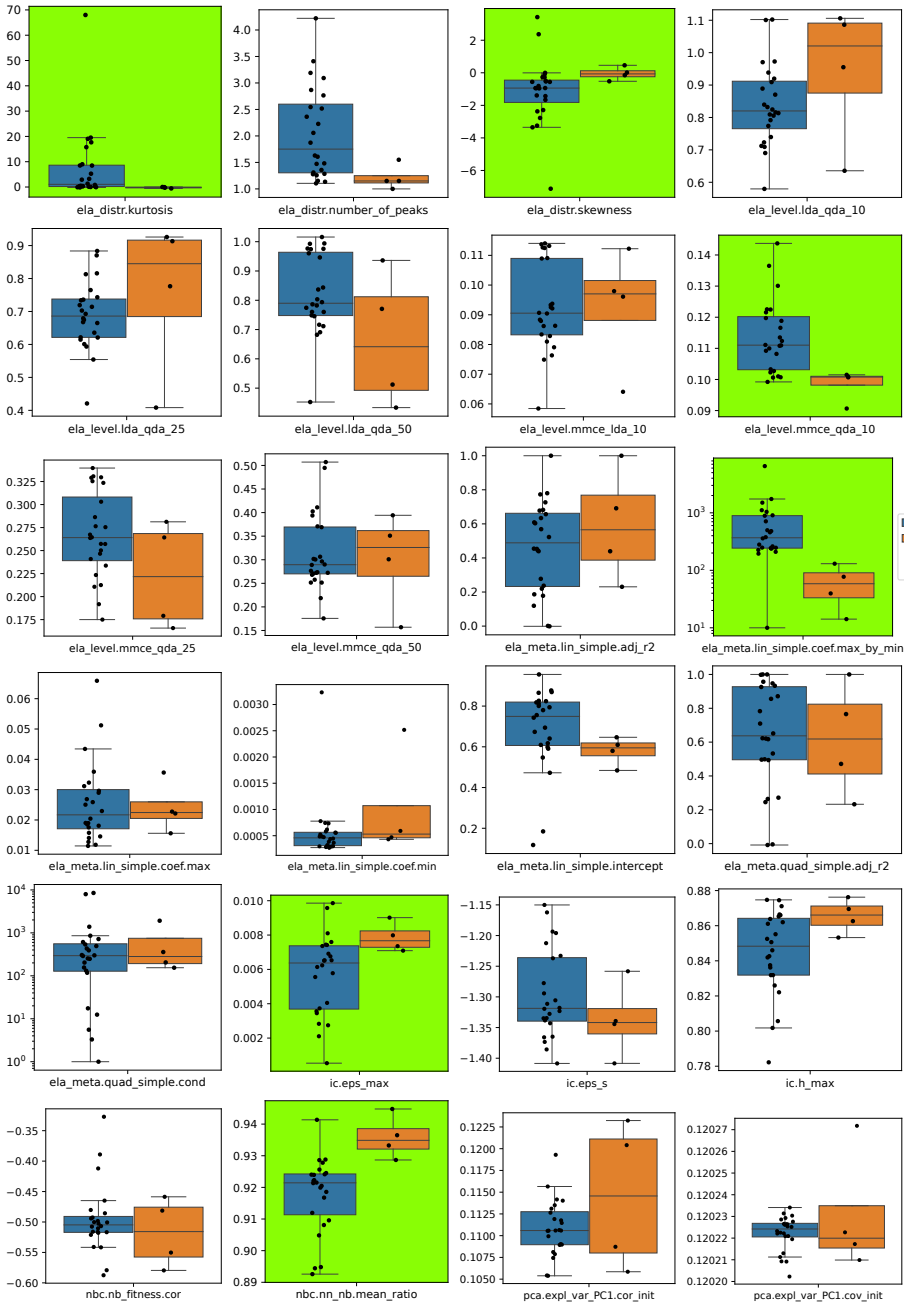

Figure 3.13: Distribution of ELA features in raw feature values considered for the clustering of 24 BBOB functions (*blue color*) and four crash functions (*orange color*) in `Crash_2`. An ELA feature is highlighted with green color, if the null hypothesis *the distribution of ELA features between the BBOB and crash functions is similar* is rejected based on the KS test with a confidence level of 95%. Figure taken from [73].

pletely different distribution between the crash functions and BBOB functions in all cases, the null hypothesis is rejected in many of the dispersion features. Subsequently, the crash functions and BBOB functions might have a different degree of dispersion, which quantifies the size of search space region with better solutions.

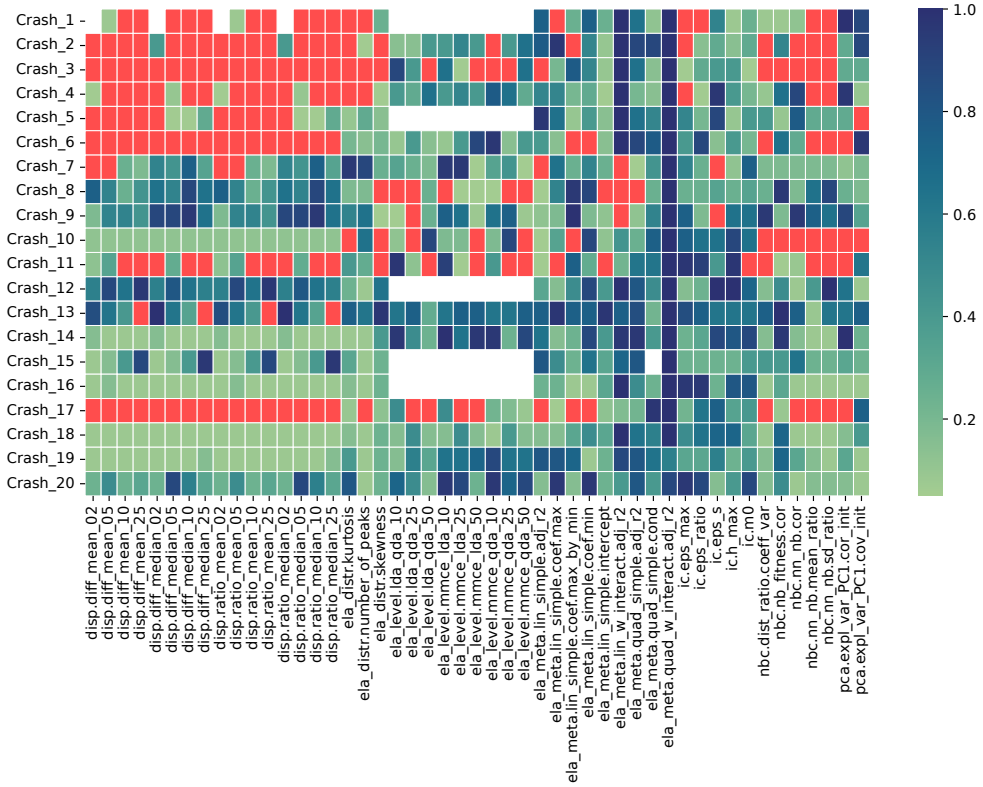

Figure 3.14: Comparison of all 49 ELA feature distributions between the crash and BBOB functions for all 20 automotive crash problems, based on the p-value computed using the KS test. A larger p-value (darker color) indicates that the null hypothesis *the distribution of ELA features between the crash and BBOB functions is similar* is less likely to be rejected, while a smaller p-value (lighter color) for a higher chance of rejection. A red color indicates that the null hypothesis is rejected with a confidence level of 95%, i.e., p-value less than 0.05, while a white color indicates that an ELA feature computation is skipped, e.g., due to a small sample size. Figure taken from [73].

Attempting to verify our hypothesis *all automotive crash problems belong to the same optimization problem class*, where a general one-for-all representative function and/or optimizer for automotive crashworthiness optimization is possible, we further

analyze the problem class of different crash problems. Using the t-SNE approach, the high-dimensional ELA feature space is projected to a 2-d visualization for all crash functions and BBOB functions, as shown in Figure 3.15. In line with our observations earlier, many of the crash functions are separated from the BBOB functions, indicating that these functions belong to different problem classes. Rather than forming one cluster, the crash functions are spread across the ELA feature space, forming their own problem classes, even for crash functions of the same type, e.g., intrusion. As opposed to our hypothesis, this insight suggests that the problem classes of automotive crashworthiness optimization could be different, depending on the problem definition. Consequently, each crash problem must be treated separately, such as the identification of appropriate representative functions and fine-tuning of optimization algorithms. In other words, an *instance*-based fine-tuning of optimization configurations approach is necessary, due to the diverse optimization problem classes across crash problems. Nonetheless, we are fully aware that this experimental setup might be inadequate to reach a decisive conclusion and an in-depth investigation is necessary, since the crash DoEs investigated have different sample sizes and dimensionalities, which could impact the ELA feature computation.

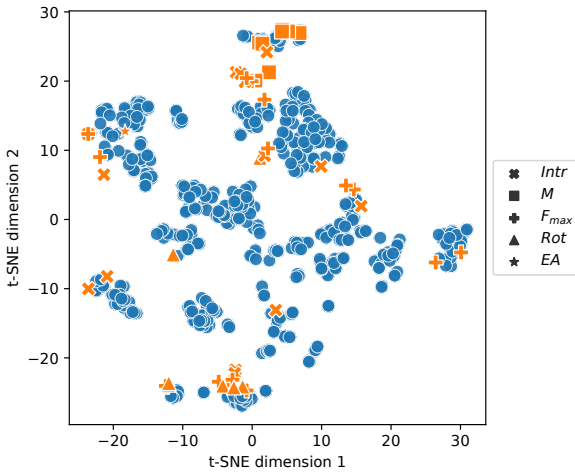

Figure 3.15: Projection of the ELA feature space to a 2d visualization for all crash functions (altogether 48 functions; *orange*) and BBOB functions (altogether $24 \times 20 = 480$ functions; *blue*) using the t-SNE approach. Figure taken from [73].

3.3.2 Comparison against Vehicle Dynamics Problems

Beyond that, we investigate the diversity of optimization problem classes across problem instances from different engineering domains. Precisely, the landscape characteristics of automotive crash problems are compared against those of vehicle dynamics problems, another common BBO problems in the automotive industry. In brief, the vehicle control systems like Anti-lock Braking System (ABS) must be optimally calibrated for maximum vehicle safety and ideal driving experience [24, 132].

Altogether five ABS problems using different tires and vehicle loads are considered, each consisting of two design variables and 10 101 DoE samples. Since such a large set of DoE samples is not available and/or infeasible for crash problems, only crash problems having a DoE sample size of at least $10 \cdot d$ in Table 3.3 are considered for comparison. Subsequently, we end up with a final set of 13 crash problems, consisting of seven side crashes, two rear crashes, and four frontal crashes. For the ELA feature computation, the approach proposed in [131] is employed, which slightly differs from our approach proposed in Section 3.1.1 in the following aspects:

- Instead of standardization, the computed ELA features are rescaled to a similar scale range using min-max scaling; and
- To discard highly correlated ELA features, a PCA approach is applied to the high-dimensional ELA feature vectors.

For a fair comparison of the optimization landscape characteristics, both steps are first performed on the BBOB functions and the identical scaling and transformation are then applied to the engineering problem instances.

As visualized in Figure 3.16, the high-dimensional ELA feature vectors are projected to a 2-d space using the first two PCA components. Notably, a clear separation between the crash and ABS problem instances can be observed, indicating that both domains are indeed different in terms of ELA features. Similar to our observations in Figure 3.15, the problem instances from the same engineering domain are scattered across the ELA feature space as well. Interestingly, the diversity within the same domain can be as large as across different domains.

To gain a deeper understanding about the separation between these two engineering domains, the corresponding ELA features are further analyzed, as shown in Figure 3.17. Here, the level set features are omitted, because the feature computation fails in many of the crash problem instances. Based on visual inspection, many of the ELA features show clear differences between the two engineering domains. For a fair

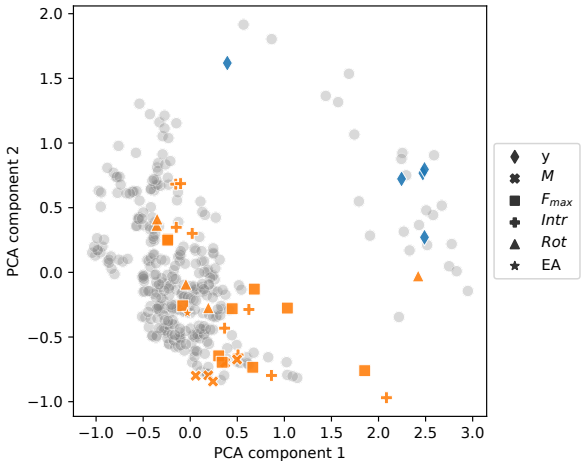

Figure 3.16: Visualization of the ELA feature space for the five ABS functions (*blue*), 30 crash functions (*orange*), and 336 BBOB functions (altogether 24 BBOB × 14 problem instances; *gray*) using the first two PCA components. Each dot represents a problem instance. Figure taken from [24].

evaluation, the distribution of ELA features between them is statistically analyzed using the two-sample KS test with the null hypothesis *the distribution of ELA features is similar*. In fact, many of the ELA features indeed have a different distribution between the crash functions and ABS problem instances, in line with our previous observations. Consequently, our results show that both engineering domains belong to different optimization problem classes. For a full result discussion, we refer to [24].

3.4 Conclusions

In this chapter, we focus on analyzing the optimization landscape characteristics of real-world expensive BBO problems, using automotive crashworthiness optimization as a representative example. Firstly, two approaches for capturing the optimization landscape characteristics are proposed in Section 3.1. On one hand, the optimization landscape characteristics of BBO problems are numerically quantified using the classical ELA features. On the other hand, the DoE2Vec approach is introduced, where the BBO problems are characterized based on some low-dimensional latent space representations computed using deep NN models like VAE. Next, a large set of BBOB instances is extensively analyzed in Section 3.2 from the perspectives of ELA features, optimization performances, and global function properties. Lastly, the optimization

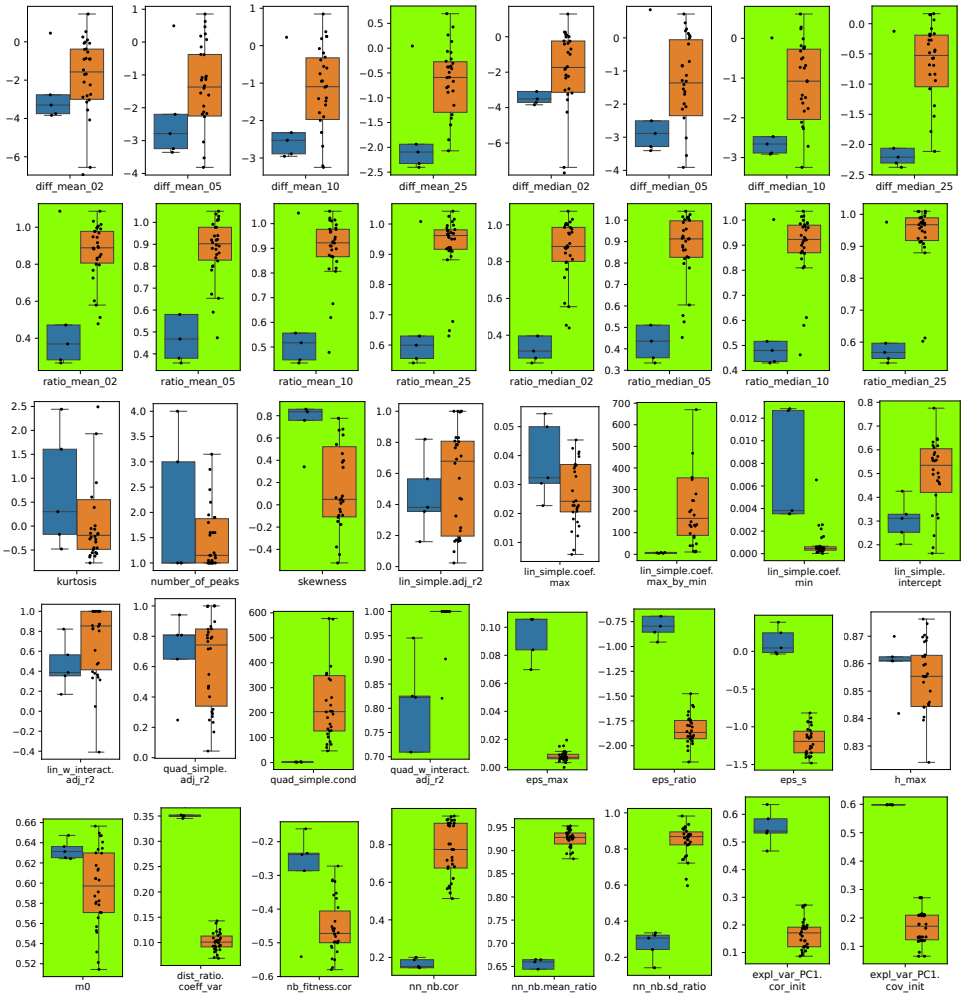

Figure 3.17: Comparison of 40 ELA features in raw feature values between the five ABS functions (*blue*) and 30 crash functions (*orange*). An ELA feature is highlighted with green color, if the null hypothesis *the distribution of ELA features between the ABS and crash functions is similar* is rejected based on the KS test with 95% confidence. Figure taken from [24].

landscape characteristics of some real-world automotive crash problem instances are analyzed in Section 3.3, based on a comparison against those of the BBOB functions and some vehicle dynamics problems.

Essentially, we provide an answer to *RQ1* in detail, by analyzing the optimization landscape characteristics of automotive crashworthiness optimization problems.

RQ1: Within the feature space defined by optimization landscape features, how are the distributions of real-world expensive BBO problems situated w.r.t. some benchmark functions?

Using our proposed ELA-based approach, the landscape characteristics of altogether 20 automotive crashworthiness optimization problems with a dimensionality between 8-d and 22-d for different vehicle crash scenarios are computed. Through hierarchical clustering and visualization of the ELA feature space, many of the crash problems are clustered into different optimization problem classes, which are separated from the BBOB functions. Following this, the BBOB functions are insufficient to represent these crash problems, which belong to distinguishable problem classes. In other words, the BBOB functions are inappropriate to be considered as representative functions for the automotive crash problems, e.g., for HPO purposes.

Beyond that, clear differences in the optimization landscape characteristics among the automotive crash problems can be observed. Instead of forming one cluster, the crash problems are spread throughout the ELA feature space, forming their own clusters, even for crash problems of the same type. Remarkably, the diversity of landscape characteristics within a single engineering domain can be as high as across different domains, when compared against some vehicle dynamic problems. Consequently, an instance-based HPO approach seems to be much more appropriate and effective for solving real-world expensive BBO problems, i.e., the fine-tuning of optimization configurations is performed specifically for a problem instance. In other words, optimization configurations should be fine-tuned according to a particular problem instance, rather than to an engineering domain.

Since appropriate representative functions for the automotive crashworthiness optimization problems cannot be identified from the BBOB suite, we shift our focus towards generating test functions having similar optimization landscape characteristics. Precisely, we evaluate the potential of a random function generator to create

similar test functions that are appropriate to be considered as representative functions for automotive crash problems in the following chapters.

The content of this chapter is mainly based on the author's contribution in the publications [24, 72, 73, 76, 139].