DOE2VEC: REPRESENTATION LEARNING FOR EXPLORATORY LANDSCAPE ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose DoE2Vec, a variational autoencoder (VAE)-based methodology to learn optimization landscape characteristics for downstream meta-learning tasks, e.g., automated selection of optimization algorithms. Principally, using large training data sets generated with a random function generator, DoE2Vec self-learns an informative latent representation for any design of experiments (DoE). Unlike the classical exploratory landscape analysis (ELA) method, our approach does not require any feature engineering and is easily applicable for high dimensional search spaces. For validation, we inspect the quality of latent reconstructions and analyze the latent representations using different experiments. The latent representations not only show promising potentials in identifying similar (cheap-to-evaluate) surrogate functions, but also can boost performances when being used complementary to the ELA features in classification tasks.

1 INTRODUCTION

Solving real-world black-box optimization problems can be extremely complicated, particularly if they are strongly nonlinear and require computationally expensive function evaluations, e.g., requiring simulator runs. As suggested by the no free lunch theorem in Wolpert & Macready (1997), there is no such things as a single-best optimization algorithm, that is capable of optimally solving all kind of problems. The task in identifying the most time- and resource-efficient optimization algorithms for each specific problem, also known as the algorithm selection problem (ASP) Rice (1976), is tedious and challenging, even with domain knowledge and experience. In recent years, landscape-aware algorithm selection has gained increasing attention from the research community, where the fitness landscape characteristics are exploited to explain the effectiveness of an algorithm across different problem instances van Stein et al. (2013); Simoncini et al. (2018). Beyond that, it has been shown that landscape characteristics are sufficiently informative in reliably predicting the performance of optimization algorithms, e.g., using Machine Learning approaches Jankovic & Doerr (2020); Jankovic et al. (2021); Dréo et al. (2019); Bischl et al. (2012); Kerschke & Trautmann (2019a); Pikalov & Mironovich (2021). In other words, the expected performance of an optimization algorithm on an unseen problem can be estimated, once the corresponding landscape characteristics have been identified. Interested readers are referred to Malan (2021); Muñoz et al. (2015a); Kerschke & Trautmann (2019a); Kerschke et al. (2019); Muñoz et al. (2015b).

Exploratory landscape analysis (ELA), for instance, considers six classes of expertly designed features, including y-distribution, level set, meta-model, local search, curvature and convexity, to numerically quantify the landscape complexity of an optimization problem, such as multi-modality, global structure, separability, variable scaling, search space homogeneity, basin size homogeneity, global to local optima contrast and plateaus Mersmann et al. (2010; 2011). Each feature class consists of a set of features, which can be relatively cheaply computed. Other than typical ASP tasks, ELA has shown great potential in a wide variety of applications, such as understanding the underlying landscape of neural architecture search problems van Stein et al. (2020), classifying the Black-Box Optimization Benchmarking (BBOB) problems Renau et al. (2021) and investigating the problem space of different benchmark problem sets Škvorc et al. (2020). Recently, ELA has been applied not only to analyze the landscape characteristics of crash-worthiness optimization problems from automotive industry, but also to identify appropriate cheap-to-evaluate functions as representative of the expensive real-world problems Long et al. (2022). While ELA is well established in

capturing the landscape characteristics of optimization problems, we would like to raise our concerns regarding the following aspects.

1. Many of the ELA features are highly correlated and redundant, particularly those within the same feature class Škvorc et al. (2020).
2. Some of the ELA features are insufficiently expressive in distinguishing problem instances Renau et al. (2019).
3. Since ELA features are manually engineered by experts, their feature computation might be biased in capturing certain landscape characteristics Seiler et al. (2020).
4. ELA features are less discriminative for high-dimensional problems Muñoz & Smith-Miles (2017).

Instead of improving the ELA method directly, e.g., searching for more discriminative landscape features, we approach the problems from a different perspective. In this paper, we introduce an automated self-supervised representation learning approach to characterize optimization landscapes by exploiting information in the latent space. By training a deep variational autoencoder (VAE) to learn the nonlinear transformations of a design of experiments (DoE), an informative feature vector can be extracted from the data set, which is essentially a generic low-dimensional representation of the optimization landscape. Thus, the name of our approach: *DoE2Vec*. While the functionality of our approach is fully independent of ELA, experimental results reveal that its performance can be further improved when combined with ELA (and vice versa). To the best of our knowledge, a similar application approach with (variational) autoencoders in learning optimization landscape characteristics is still lacking.

The remainder of this paper is structured as follow: Section 2 briefly introduces the state-of-the-art of representation learning of optimization landscapes as well as the concepts of (variational) autoencoder. This is followed by the description of our methodology in Section 3. Next, we explain and discuss our experimental results in Section 4. Lastly, conclusions and outlooks are provided in Section 5.

2 Representation of Optimization Landscape

In the conventional ELA approach, landscape features are computed primarily based on the fitness landscape (a DoE of some sample points $\mathcal{X}$ and the corresponding fitness values $f\colon \mathcal{X} \to \mathbb{R}$), with an assumption that the search space is continuous, i.e., $\mathcal{X} \subseteq \mathbb{R}^d$, where d represents the dimension. In this work, we consider ELA features similar to those in Long et al. (2022), which do not require additional sampling, and compute them with the package `flacco` by Kerschke & Trautmann (2019b;c).

To overcome the aforementioned drawbacks of the classical ELA approach, attentions have been focused on developing algorithm selection approaches without the need of classical landscape features. For example, Prager et al. (2022) proposed two feature-free approaches using deep learning method, where optimization landscapes can be represented through either 1) image-based fitness maps or 2) graph-like fitness clouds. In the first approach, convolutional neural networks are employed to project data sets into two-dimensional images, called fitness maps, using different dimensionality reduction techniques. On the other hand, data sets are embedded into point clouds using modified point cloud transformers, which can accurately capture the global landscape characteristics, in the second approach. Nonetheless, both approaches have their own weaknesses, such as the fitness map approach suffers from curse of dimensionality, while the fitness cloud approach is currently limited to fixed training sample size. Additional relevant works can be found in Alissa et al. (2019); Seiler et al. (2020; 2022); Prager et al. (2021). These feature-free approaches are used directly as classifiers in downstream tasks, unlike the proposed approach which results in a feature set that can be combined with other feature sets such as conventional ELA feature sets. In our work we do not propose to replace conventional ELA features, but actually extend them with autoencoder based latent-space features. Since the implementation of both approaches was not available, a direct comparison in terms of classifying high-level properties between them and this work is infeasible. Following this, results from the downstream tasks in this work can partially be compared to the mentioned results in Seiler et al. (2022), taken into account that additional hyperparameter tuning was involved in their classification experiments with ELA features.

Our approach is capable of learning the representations of an optimization landscapes in an automated, generic and unsupervised manner, with the advantage that the learned features are not biased towards any particular landscape characteristic. Unlike previously mentioned approaches, our proposed method is independent of sampling method. By using only fully connected (dense) layers that learn from one-dimensional (flattened) landscapes, an autoencoder (AE) or a VAE is, in theory, capable of learning any number of input-dimensions without scaling difficulties. Furthermore, the fast-to-train (V)AE models can be easily shared in practice.

2.1 AUTOENCODER

A standard AE usually has a symmetrical architecture, consisting of three components: an encoder, a hidden layer, also known as bottleneck, and a decoder (Figure 1). In short, an encoder projects the input space $\mathcal{X}$ to a representative feature space $\mathcal{H}$, such that $f\colon \mathcal{X} \to \mathcal{H}$, while a decoder transforms the feature space back to the input space $g\colon \mathcal{H} \to \hat{\mathcal{X}}$ Charte et al. (2020; 2018). In other words, AE attempts to optimally reconstruct the original input space $\mathcal{X}$, by minimizing the reconstruction error $\mathcal{L}(\mathcal{X}, \hat{\mathcal{X}})$, e.g. mean squared error, during the (unsupervised) training process. Commonly, AE is constructed with a bottleneck layer of lower dimension d, i.e. $d_{\mathcal{H}} < d_{\mathcal{X}}$ to improve its ability in capturing crucial representations of the input data.

Following this, AE has rapidly gained popularity in the field of dimensionality reduction as well as representation learning Bengio et al. (2013); Tschannen et al. (2018). In comparison with the simple principal component analysis (PCA) technique, AE has the advantage that nonlinear representation features can be learned with activation functions, such as sigmoid function.

2.2 VARIATIONAL AUTOENCODER

Originating from the same family as traditional AE, a VAE typically has a similar architecture with some modifications, as shown in Figure 1. Unlike AE, the latent space of a VAE is encoded as

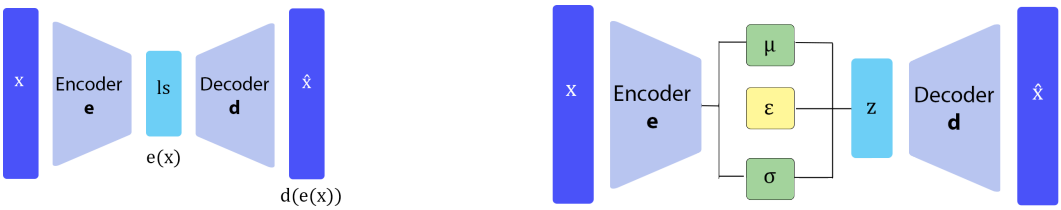

Figure 1: Example of a (standard) AE on the left and VAE architecture on the right used in DoE2Vec. The latent space z is defined by $z = \mu + \sigma \cdot \epsilon$, where ϵ denotes the sampling with mean μ and variance σ.

a distribution by using a mean and variance layer, together with a sampling method. Following this, the latent space can be properly regularized to provide more meaningful features. During the training process, a common loss function (Equation 1) of VAE is to be minimized, consisting of a regularization term (Equation 2), which can be expressed as the Kullback–Leibler (KL) divergence, and a reconstruction error, e.g., mean squared error (Equation 3).

$$\mathcal{L}(\mathcal{X}, \hat{\mathcal{X}}) = \mathcal{L}_{\text{MSE}}(\mathcal{X}, \hat{\mathcal{X}}) + \beta \cdot \mathcal{L}_{\text{KL}}(\mathcal{X}, \hat{\mathcal{X}}) \tag{1}$$

$$\mathcal{L}_{\text{KL}}(\mathcal{X}, \hat{\mathcal{X}}) = \frac{1}{2} \sum_{i=1}^{|\mathcal{X}|} (\exp(\sigma_i) - (1 + \sigma_i) + \mu_i^2) \tag{2}$$

$$\mathcal{L}_{\text{MSE}}(\mathcal{X}, \hat{\mathcal{X}}) = \sum_{x \in \mathcal{X}, \hat{x} \in \hat{\mathcal{X}}} (x - \hat{x})^2 \tag{3}$$

where a weighting factor β is included to ensure a good trade-off between regularization and reconstruction error, σ and μ are the variance and mean latent layers of the VAE and $\hat{\mathcal{X}}$ is the reconstruction of the sampled latent space vectors $\mathcal{Z}$, where $z \in \mathcal{Z}$. Detailed explanations regarding VAE can be found in Kingma & Welling (2013; 2019).

3 DOE2VEC

Generally, our proposed method uses a VAE with similar design as described in Section 2.2. Precisely, our VAE model has an architecture of altogether seven fully connected layers, where rectified linear unit (ReLU) activation functions are assigned to the hidden layers, while a sigmoid activation function for the final output layer of the decoder. The encoder is composed of four fully connected layers with sizes depending on the input size (n), starting with the input layer n, hidden layers with sizes $n/2$ and $n/4$ and the latent size ls for the mean and log variance of the latent space. The decoder is composed of three fully connected layers with sizes $n/4$, $n/2$ and n for the final output layer. The focus of our approach lies on VAE, rather than AE, because it has the additional benefits of regularizing the latent space without loss of generalisation. For comparison, we consider a standard AE model as well (with the same number of hidden layers, except that the latent space is now a single dense layer, instead of a mean and log variance layer with a sampling scheme). Full specifications of the different models are available in our Github repository for Review (2022), while pre-trained models are also available on Huggingface.

In the first step of DoE2Vec, a data set of function landscapes is generated using the random function generator from Long et al. (2022), which was originally proposed by Tian et al. (2020). The main advantage of using this function generator is that a large training set can be easily generated, covering a wide range of function landscape of different complexity. Next, all the random functions are computed based on a fixed DoE with a sample size 2^m, where m is a user defined parameter. By default, a Sobol sequence by Sobol' (1967) is used as sampling scheme, but any sampling scheme or even a custom DoE can be utilized in practice. Initially, the DoE samples are created within domain [0,1], which can be re-scaled to any desired function boundaries, as the DoE2Vec method is scale-invariant by design. Afterward, the evaluated response surfaces f are re-scaled to [0,1], before being fed into (V)AE models as input vectors. Once the training process is finished, the latent space of the trained models can be used as feature vectors for further downstream classification tasks, such as optimization algorithm selection and configuration.

In next section, we will show that the learned feature representations have attractive characteristics and they are indeed useful for downstream classification and meta-learning tasks.

4 EXPERIMENTAL RESULTS AND DISCUSSIONS

In this work, we have conducted altogether four experiments for different research purposes. Basically, the first two experiments are to investigate the quality of VAE models trained and to validate the latent space representation learned, while the last two experiments are to verify the potential of the DoE2Vec approach for downstream meta-learning tasks.

1. Analyzing the reconstruction of function landscapes and the impact of the latent space size and other model parameters.
2. Investigating the latent space differences between a standard AE and a VAE model.
3. Identifying functions with similar landscapes based on the latent representations.
4. A downstream multiclass classification experiment to show the potential of proposed approach in practice, using the latent feature vectors as input.

In our experiments, we fix the sampling scheme to Sobol sequence and m to eight, ending up with a DoE of 256 samples. Unless otherwise specified, all AE and VAE models are trained on a set of $250,000$ five-dimensional ($5d$) random functions.

4.1 RECONSTRUCTION OF FUNCTION LANDSCAPES

In our first experiment, we are interested in investigating the impact of two model parameters, including the KL loss weight and the latent size, on both the KL loss and reconstruction error. For this purpose, a total of 30 VAE models are trained using one-factor-at-a-time method and a combination of six different latent dimensions $(4, 8, 16, 24, 32)$ and five different KL loss weights $(0.0001, 0.0002, 0.001, 0.005, 0.01)$. Results in the left subplot of Figure 2 clearly show that latent sizes have a positive effect on the total validation loss, with the improvement diminishes after 16.

On the other side, the reconstruction loss can be improved with smaller weights for the KL loss, as shown in the right subplot of Figure 2. The combined effects of these parameters on the KL loss and

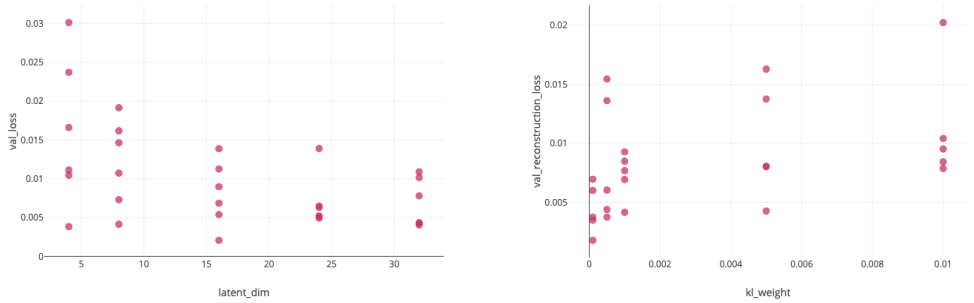

Figure 2: Comparison of different latent sizes and KL weights. Left: Latent sizes against total validation loss. Right: KL loss weight against the validation reconstruction loss.

reconstruction error can be better visualized through a parallel coordinates plot (Figure 3), where a slight conflict between these two loss components can be observed. In the remaining experiments,

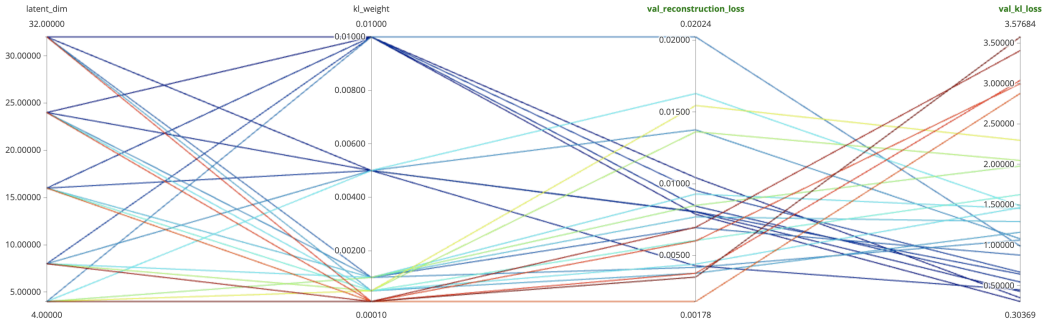

Figure 3: Parallel coordinates plot of different latent sizes and weights for the KL loss in relation with the reconstruction validation and KL validation losses.

we use a latent space size of either 24 or 32 (expressed as (V)AE-24 or (V)AE-32) and a KL loss of 0.001 as a good compromise between the two loss terms. It is to be noted that, while the (V)AE architecture could be further improved by applying neural architecture search, we leave it for future work, since the landscape reconstruction is not the ultimate purpose of our approach. In fact, the reconstruction here is meant to be a way in quantifying the quality of representation learning. Alternatively, we reconstruct the 24 BBOB functions (first problem instance) using a VAE model with 24 latent dimensions to verify its capability in reconstructing a large variety of functions (Figure 4).

4.2 Latent Space Representation

Next, we project the latent space of an AE and a VAE from our DoE2Vec approach onto a $2d$ visualization using multidimensional scaling (MDS) method. Results in Figure 5 show the latent feature vectors of 100 problem instances of all 24 BBOB functions (noise-free function set from the BBOB 2009 workshop Hansen et al. (2009)), where each dot represents a feature vector of a BBOB function and the color denotes the corresponding BBOB function. For a comprehensive comparison, a similar MDS visualization for the feature vectors with all the (considered) classical ELA is included as well. Interestingly, the latent space of ELA has a better looking cluster-structure, while the latent space of AE and VAE have a clear structure. As expected, the latent space of AE contains a few outliers and is in general more widespread than the latent space of VAE.

To further analyze the differences in latent space representation between the AE and VAE model, we use the generator part of the models to iteratively generate new function landscapes. To be precise,

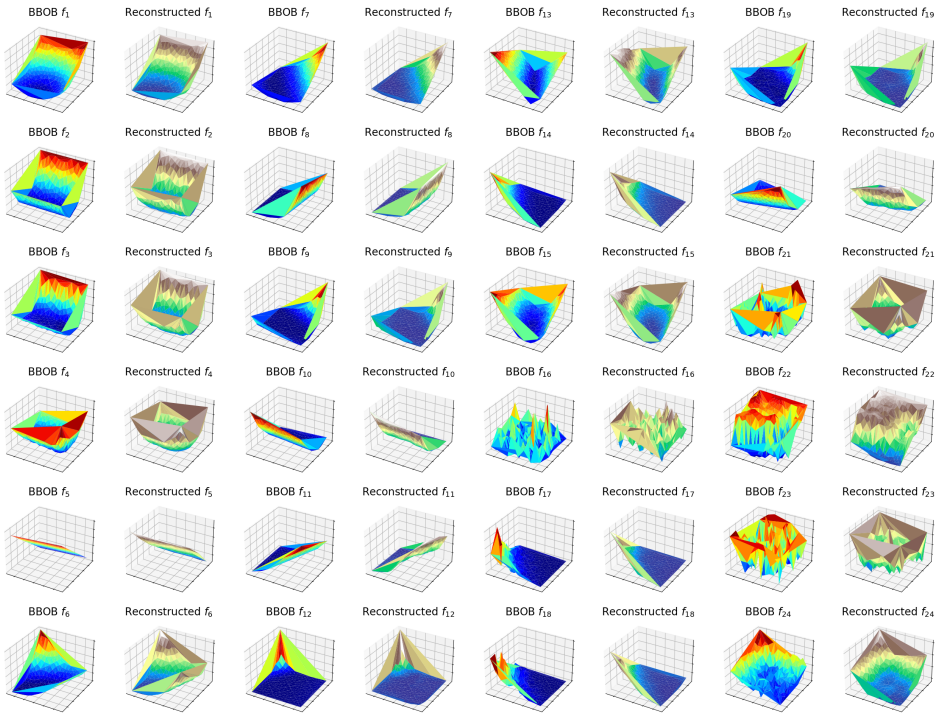

Figure 4: Reconstructions of the 24 $2d$ BBOB funcions (labelled from f_1 to f_{24}) using a VAE-24 model. The $3d$ surface plots are generated based on a DOE with 256 samples using triangulation.

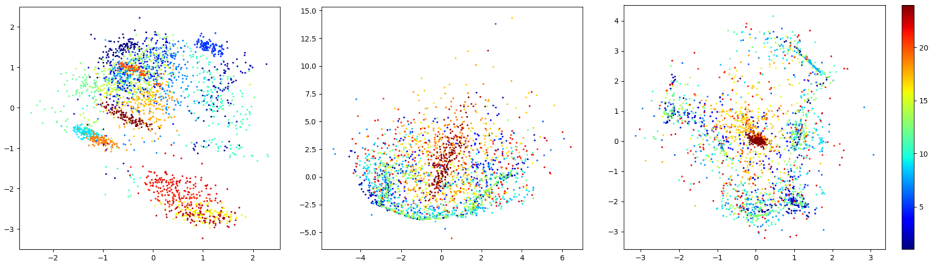

Figure 5: $2d$ MDS visualization of the (normalized) ELA features on the left, the latent space of an AE-32 model in the center and the latent space of a VAE-32 model on the right for the first 100 instances of 24 BBOB functions (one color represents a BBOB function).

we select one latent representation as starting point and individually vary each latent variable with small steps between -1 and $+1$ of its original value. In Table 1, the generated landscapes are shown only for the extreme cases of an AE-4 and a VAE-4 model. Based on this result, two important conclusions can be drawn.

1. The interpolation for both models is very smooth, showing that the models can learns the landscape representations well and are able to understand the structure, even though the dimensionality of the function landscapes is not known.

2. The VAE model utilizes a more compact feature representation, where its output variance is greater than those of the AE model for the same variation of latent features.

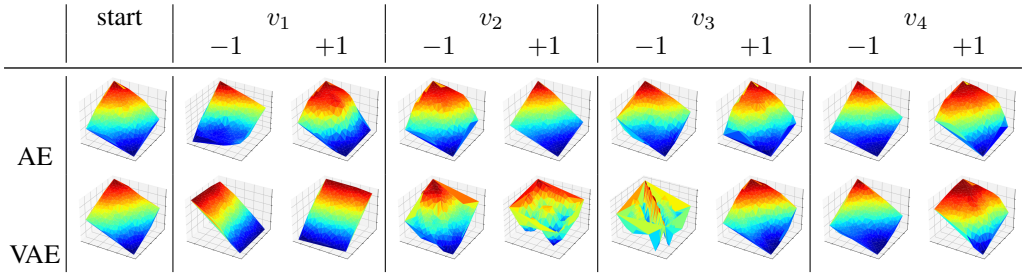

Table 1: Comparison of function landscapes generated by an AE-4 and a VAE-4 model for different latent variables, using the BBOB f_{22} encoding as starting point (v_1, v_2, v_3, v_4). In each column, the latent features are separately varied by -1 or $+1$. Complete variation of the latent features can be found in videos available in our Github repository.

4.3 Similar Representative Functions

The first main application of DoE2Vec is to identify cheap-to-evaluate functions with similar latent space representation for a given DoE, which can be very useful when dealing with real-world expensive black box optimization problems, as also described in Long et al. (2022). In this way, large experiments can now be conducted on a "similar" function group at a much lower computational cost. Since a direct analytical verification of the results is very challenging, we instead use a visual inspection based on $2d$ functions to showcase the potential of the proposed method. In Table 2, the 24 BBOB functions are paired with their respective "nearest" random function, where *nearest* is defined as the random function that has the closest feature representation in terms of Euclidean distance. For most BBOB functions, a very well fitting (almost identical) random function can be identified by the DoE2Vec model, where random functions of equal complexity are suggested for the more complex BBOB functions.

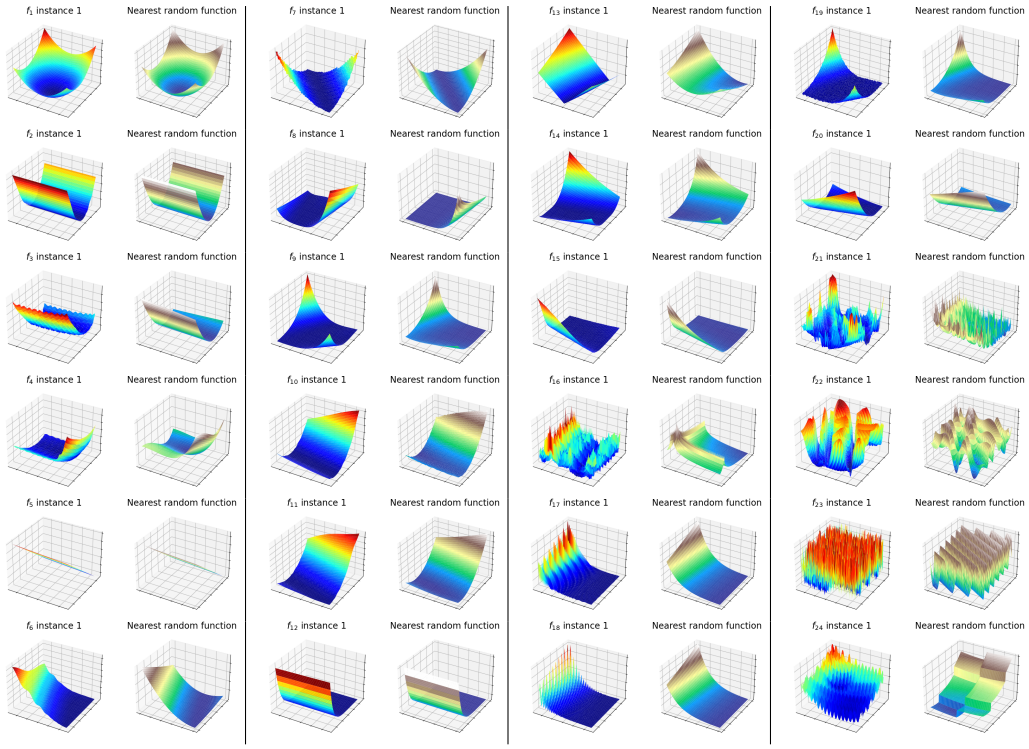

Table 2: Four columns with pairs of BBOB function and their respective nearest random function identified based on latent feature vectors using the VAE-24 model.

4.4 CLASSIFICATION TASKS

Secondly, the DoE2Vec approach is designed to learn characteristic representations of an optimization landscape, which can be verified through a classical classification task of function high level properties. These high level properties, such as multimodality, global structure and funnel structure, are important for ASP, as they often determine the difficulty of an optimization problem. Table 3 illustrates the BBOB functions and their associated high level properties.

BBOB function	Multimodal	Global Structure	Funnel
1: Sphere	none	none	yes
2: Ellipsoidal separable	none	none	yes
3: Rastrigin separable	high	strong	yes
4: Büche-Rastrigin	high	strong	yes
5: Linear Slope	none	none	yes
6: Attractive Sector	none	none	yes
7: Step Ellipsoidal	none	none	yes
8: Rosenbrock	low	none	yes
9: Rosenbrock rotated	low	none	yes
10: Ellipsoidal high conditioned	none	none	yes
11: Discus	none	none	yes
12: Bent Cigar	none	none	yes
13: Sharp Ridge	none	none	yes
14: Different Powers	none	none	yes
15: Rastrigin multimodal	high	strong	yes
16: Weierstrass	high	medium	none
17: Schaffer F7	high	medium	yes
18: Schaffer F7 moderately ill-cond.	high	medium	yes
19: Griewank-Rosenbrock	high	strong	yes
20: Schwefel	medium	deceptive	yes
21: Gallagher 101 Peaks	medium	none	none
22: Gallagher 21 Peaks	low	none	none
23: Katsuura	high	none	none
24: Lunacek bi-Rastrigin	high	weak	yes

Table 3: High level properties of the 24 BBOB functions. This table was first introduced in Seiler et al. (2022)

In this experiment, a standard random forest (RF) model (using `sklearn` from Pedregosa et al. (2011)) is implemented for the multiclass classification tasks based on the latent representations learned by four DoE2Vec models with different latent space sizes, consisting of AE-24, AE-32, VAE-24 and VAE-32. In other words, the high level properties of a BBOB function is to be predicted using the latent representations. Again, we compare the DoE2Vec approach against the classical ELA method, which is specifically constructed to excel in exactly this kind of function property classification tasks. Beyond that, a combination of classical ELA features with a VAE-32 model is included to evaluate the complimentary effect of the different feature sets.

The classification results (macro F1 scores) of the different feature sets are summarized in Table 4. It is not surprising that the ELA features generally perform very well and outperform the latent features most of the time, especially in classifying the global structure and multimodal landscapes. Fascinatingly, the classification performances can be significantly improved when the DoE2Vec is combined with the classical ELA method, indicating that both feature sets seem to be complimentary to each other.

Dimension d	Landscape	AE-24	AE-32	VAE-24	VAE-32	ELA	ELA & VAE-32
2	multimodal	0.875	0.849	0.877	0.856	0.984	**0.991**
	global structure	0.903	0.904	0.902	0.889	0.983	**0.998**
	funnel	0.985	0.974	0.956	0.978	**1.000**	**1.000**
5	multimodal	0.908	0.903	0.880	0.889	0.963	**0.998**
	global structure	0.838	0.828	0.810	0.793	**1.000**	**1.000**
	funnel	**1.000**	**1.000**	0.996	0.991	**1.000**	**1.000**
10	multimodal	0.877	0.813	0.844	0.838	**1.000**	**1.000**
	global structure	0.794	0.737	0.783	0.745	0.902	**0.991**
	funnel	**0.998**	0.993	**0.997**	0.993	0.972	**0.997**
20	multimodal	0.726	0.722	0.700	0.694	0.970	**0.991**
	global structure	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**

Table 4: Classification results (macro F1 score) using a standard RF model with 100 trees, trained on the feature representations (from AE, VAE, classical ELA or ELA combined with VAE) of the first 100 instances for each BBOB function and validated on instance 101 to 120.

5 Conclusions and Future Work

In this work we propose *DoE2Vec*, a VAE-based approach to learn the latent representations of an optimization landscape, similar to the classical landscape features. Using a wide set of experiments, we have shown that DoE2Vec is able to reconstruct a large set of functions accurately and that the approach can be used for downstream meta-learning tasks, such as algorithm selection. The proposed methodology can be effectively used next to existing techniques such as classical ELA features to further increase the classification accuracy of certain downstream tasks. In fact, DoE2Vec can learn good feature representations for optimization landscapes and has several advantages over the conventional ELA approach, such as feature engineering or selection knowledge is not required, domain knowledge in ELA is not needed and it is applicable to optimization tasks in a very straightforward manner. Nonetheless, there are a few known limitations to the proposed method, such as

- Our approach is scale-invariant, but not rotation- or translation-invariant. Using a different loss function to train the autoencoders might be able to improve this, however it is unclear yet which loss function that could be,
- If a custom DoE sample is used, the model needs to be trained from scratch (no pre-trained model can be used). This typically takes a few minutes up to an hour, depending on the sample size and number of random functions to train on,
- The learned feature representations are a black-box that are hard to interpret directly, and
- It might not be able to outperform feature engineering approaches, since the method is not tailored to a specific downstream task.

In future work, we plan to improve our approach by tackling some of the challenges mentioned. Apart from that, we would like to verify the usefulness of the random functions with similar latent features w.r.t. the performances of an optimization algorithm.

6 Reproducibility Statement

For anyone interested in our work, we provide an open-source documented implementation of our package at https://github.com/xxxxxxx/doe2vec, with visualizations and video explanations. Pre-trained models (weights) are avaiable on Huggingface https://huggingface.co/xxxxxx. All models are trained using a single T4 GPU and the computations are carbon neutral (CO_2-free) by using solar power. Average training time of a model on $250,000$ random functions was five minutes.

REFERENCES

Mohamad Alissa, Kevin Sim, and Emma Hart. Algorithm selection using deep learning without feature extraction. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, pp. 198–206. Association for Computing Machinery, 2019. ISBN 9781450361118. doi: 10.1145/3321707.3321845.

Yashua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35:1798–1828, 08 2013. doi: 10.1109/TPAMI.2013.50.

Bernd Bischl, Olaf Mersmann, Heike Trautmann, and Mike Preuß. Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pp. 313–320, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311779. doi: 10.1145/2330163.2330209.

David Charte, Francisco Charte, Salvador García, María J. del Jesus, and Francisco Herrera. A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Information Fusion*, 44:78–96, nov 2018. doi: 10.1016/j.inffus.2017.12.007.

David Charte, Francisco Charte, María J. del Jesus, and Francisco Herrera. An analysis on the use of autoencoders for representation learning: Fundamentals, learning task case studies, explainability and challenges. *Neurocomputing*, 404:93–107, sep 2020. doi: 10.1016/j.neucom.2020.04.057.

Johann Dréo, Carola Doerr, and Yann Semet. Coupling the design of benchmark with algorithm in landscape-aware solver design. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '19, pp. 1419–1420, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367486. doi: 10.1145/3319619.3326821.

XXXXX Masked for Review. Xxxxx/doe2vec: General public release, September 2022. URL `https://doi.org/10.5281/zenodo.XXXXX`.

Nikolaus Hansen, Steffen Finck, Raymond Ros, and Anne Auger. Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions. Research Report RR-6829, INRIA, 2009. URL `https://hal.inria.fr/inria-00362633`.

Anja Jankovic and Carola Doerr. Landscape-aware fixed-budget performance regression and algorithm selection for modular cma-es variants. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, GECCO '20, pp. 841–849, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371285. doi: 10.1145/3377930.3390183.

Anja Jankovic, Gorjan Popovski, Tome Eftimov, and Carola Doerr. *The Impact of Hyper-Parameter Tuning for Landscape-Aware Performance Regression and Algorithm Selection*, pp. 687–696. GECCO '21. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450383509. doi: 10.1145/3449639.3459406.

Pascal Kerschke and Heike Trautmann. Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning. *Evolutionary Computation*, 27(1):99–127, 2019a. ISSN 1063-6560. doi: 10.1162/evco_a_00236.

Pascal Kerschke and Heike Trautmann. *Comprehensive Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems Using the R-Package Flacco*, pp. 93–123. Studies in Classification, Data Analysis, and Knowledge Organization. Springer International Publishing, Cham, 2019b. ISBN 978-3-030-25147-5. doi: 10.1007/978-3-030-25147-5_7.

Pascal Kerschke and Heike Trautmann. flacco: Feature-based landscape analysis of continuous and constrained optimization problems, 2019c. URL `https://github.com/kerschke/flacco`. Last accessed 15 January 2022.

Pascal Kerschke, Holger Hoos, Frank Neumann, and Heike Trautmann. Automated algorithm selection: Survey and perspectives. *Evolutionary Computation*, 27(1):3–45, 2019. ISSN 1063-6560. doi: 10.1162/evco_a_00242.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. doi: 10.48550/ARXIV.1312.6114.

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. doi: 10.1561/2200000056.

Fu Xing Long, Bas van Stein, Moritz Frenzel, Peter Krause, Markus Gitterle, and Thomas Bäck. Learning the characteristics of engineering optimization problems with applications in automotive crash. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, pp. 1227–1236, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392372. doi: 10.1145/3512290.3528712.

Katherine Mary Malan. A survey of advances in landscape analysis for optimisation. *Algorithms*, 14(2):40, 2021. ISSN 1999-4893. doi: 10.3390/a14020040.

Olaf Mersmann, Mike Preuss, and Heike Trautmann. Benchmarking evolutionary algorithms: Towards exploratory landscape analysis. In Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph (eds.), *Parallel Problem Solving from Nature, PPSN XI*, pp. 73–82, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15844-5. doi: 10.1007/978-3-642-15844-5_8.

Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. Exploratory landscape analysis. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pp. 829–836, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450305570. doi: 10.1145/2001576.2001690.

Mario A. Muñoz and Kate A. Smith-Miles. Performance analysis of continuous black-box optimization algorithms via footprints in instance space. *Evol. Comput.*, 25(4):529–554, dec 2017. ISSN 1063-6560. doi: 10.1162/evco_a_00194.

Mario Andrés Muñoz, Michael Kirley, and Saman K. Halgamuge. Exploratory landscape analysis of continuous space optimization problems using information content. *IEEE Transactions on Evolutionary Computation*, 19(1):74–87, 2015a. doi: 10.1109/TEVC.2014.2302006.

Mario Andrés Muñoz, Yuan Sun, Michael Kirley, and Saman K. Halgamuge. Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. *Information Sciences*, 317:224–245, 2015b. ISSN 0020-0255. doi: 10.1016/j.ins.2015.05.010.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85): 2825–2830, 2011. URL `http://jmlr.org/papers/v12/pedregosa11a.html`.

Maxim Pikalov and Vladimir Mironovich. *Automated Parameter Choice with Exploratory Landscape Analysis and Machine Learning*, pp. 1982–1985. GECCO '21. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450383516. doi: 10.1145/3449726.3463213.

Raphael Patrick Prager, Moritz Vinzent Seiler, Heike Trautmann, and Pascal Kerschke. Towards feature-free automated algorithm selection for single-objective continuous black-box optimization. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, 2021. doi: 10.1109/SSCI50451.2021.9660174.

Raphael Patrick Prager, Moritz Vinzent Seiler, Heike Trautmann, and Pascal Kerschke. Automated algorithm selection in single-objective continuous optimization: A comparative study of deep learning and landscape analysis methods. In Günter Rudolph, Anna V. Kononova, Hernán Aguirre, Pascal Kerschke, Gabriela Ochoa, and Tea Tušar (eds.), *Parallel Problem Solving from Nature – PPSN XVII*, pp. 3–17, Cham, 2022. Springer International Publishing. ISBN 978-3-031-14714-2.

Quentin Renau, Johann Dreo, Carola Doerr, and Benjamin Doerr. Expressiveness and robustness of landscape features. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '19, pp. 2048–2051, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 978-1-4503-6748-6. doi: 10.1145/3319619.3326913.

Quentin Renau, Johann Dreo, Carola Doerr, and Benjamin Doerr. Towards explainable exploratory landscape analysis: Extreme feature selection for classifying bbob functions. In Pedro A. Castillo and Juan Luis Jiménez Laredo (eds.), *Applications of Evolutionary Computation*, pp. 17–33, Cham, 04 2021. Springer International Publishing. ISBN 978-3-030-72698-0. doi: 10.1007/978-3-030-72699-7_2.

John R. Rice. The algorithm selection problem. volume 15 of *Advances in Computers*, pp. 65–118. Elsevier, 1976. doi: 10.1016/S0065-2458(08)60520-3.

Moritz Seiler, Janina Pohl, Jakob Bossek, Pascal Kerschke, and Heike Trautmann. Deep learning as a competitive feature-free approach for automated algorithm selection on the traveling salesperson problem, 2020.

Moritz Vinzent Seiler, Raphael Patrick Prager, Pascal Kerschke, and Heike Trautmann. A collection of deep learning-based feature-free approaches for characterizing single-objective continuous fitness landscapes, 2022.

David Simoncini, Sophie Barbe, Thomas Schiex, and Sébastien Verel. Fitness landscape analysis around the optimum in computational protein design. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '18, pp. 355–362, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356183. doi: 10.1145/3205455.3205626.

Il'ya Meerovich Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967. ISSN 0041-5553. doi: 10.1016/0041-5553(67)90144-9.

Ye Tian, Shichen Peng, Xingyi Zhang, Tobias Rodemann, Kay Chen Tan, and Yaochu Jin. A recommender system for metaheuristic algorithms for continuous optimization based on deep recurrent neural networks. *IEEE Transactions on Artificial Intelligence*, 1(1):5–18, 2020. doi: 10.1109/TAI.2020.3022339.

Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning, 2018.

Bas van Stein, Michael Emmerich, and Zhiwei Yang. Fitness landscape analysis of nk landscapes and vehicle routing problems by expanded barrier trees. In *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, pp. 75–89. Springer, 2013.

Bas van Stein, Hao Wang, and Thomas Bäck. Neural network design: Learning from neural architecture search. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1341–1349. IEEE, 2020. doi: 10.1109/SSCI47803.2020.9308394.

Urban Škvorc, Tome Eftimov, and Peter Korošec. Understanding the problem space in single-objective numerical optimization using exploratory landscape analysis. *Applied Soft Computing*, 90:106138, 2020. ISSN 1568-4946. doi: 10.1016/j.asoc.2020.106138.

David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. doi: 10.1109/4235.585893.