SLIPMAP: Fast and Robust Manifold Visualisation for Explainable AI

Anton Björklund(✉), Lauri Seppäläinen, and Kai Puolamäki

University of Helsinki, Helsinki, Finland
{anton.bjorklund,lauri.seppalainen,kai.puolamaki}@helsinki.fi

Abstract. We propose a new supervised manifold visualisation method, SLIPMAP, that finds local explanations for complex black-box supervised learning methods and creates a two-dimensional embedding of the data items such that data items with similar local explanations are embedded nearby. This work extends and improves our earlier algorithm and addresses its shortcomings: poor scalability, inability to make predictions, and a tendency to find patterns in noise. We present our visualisation problem and provide an efficient GPU-optimised library to solve it. We experimentally verify that SLIPMAP is fast and robust to noise, provides explanations that are on the level or better than the other local explanation methods, and are usable in practice.

Keywords: Manifold visualisation · Explainable AI · Local approximation

1 Introduction

The goal of manifold visualisation is to find a low-dimensional visualisation of high-dimensional data. We recently introduced a method that combines manifold visualisation with *explainable artificial intelligence* (XAI), called SLISEMAP [6,7]. SLISEMAP creates an embedding of data points such that points nearby in the embedding have similar explanations. (for a given black box machine learning model). Figure 1 shows an example of an embedding (left) and explanations in the form of linear coefficients (right). SLISEMAP has already been used in studying physical systems [29], for studying molecular properties [4], and to reduce data dimensionality in manufacturing [27].

The practical application of SLISEMAP is hindered by four shortcomings: (i) *Speed.* SLISEMAP scales quadratically with the amount of data, so it is impractical to visualise large datasets (larger than $\sim 10^4$ points). The solution in [7] is subsampling: train on a subset of the data and, if necessary, add the remaining points to the trained SLISEMAP post-hoc. (ii) *New data.* However, adding new data is only possible if the value of the target variable is known [7]. (iii) *No predictive model.* Since there is no principled way of adding points to the embedding, SLISEMAP cannot predict the values of the target variable. (iv) *Behaviour*

I. Miliou et al. (Eds.): IDA 2024, LNCS 14642, pp. 223–235, 2024.
https://doi.org/10.1007/978-3-031-58553-1_18

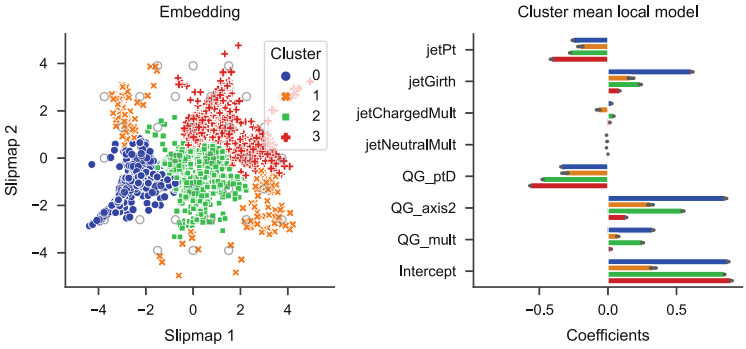

Fig. 1. SLIPMAP embedding of the *Jets* dataset used in a classification task described in Sect. 4 is shown on the left. The local models explaining the black box classifier have been clustered, and the mean coefficients for each cluster are shown on the right.

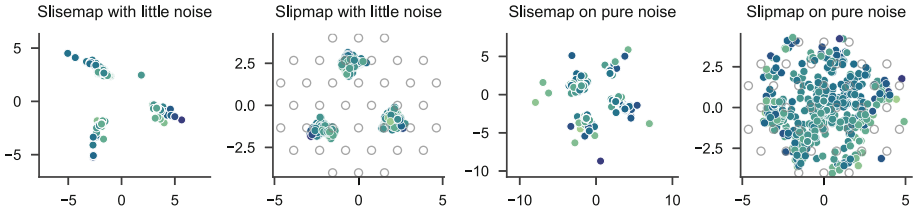

Fig. 2. Both SLISEMAP (left) and SLIPMAP (2nd from the left) correctly find the three modes for a toy data of 500 points constructed as in Fig. 1 of [7] ($y = \max(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) + \mathcal{N}(0, 0.01)$ and $\boldsymbol{x} \sim N(0,1)^4 \in \mathbb{R}^4$), each of the three visual clusters corresponding to a linear model $f_j(\boldsymbol{x}) \approx \boldsymbol{x}_j$, where $j = \arg\max_{i \in \{1,2,3\}} \boldsymbol{x}_i$. However, SLISEMAP outputs visual clusters, even when the target variable y is Gaussian noise (2nd from the right). In contrast, SLIPMAP (right) overfits less due to the equally spaced prototypes and Gaussian kernel (see Sect. 2.1), leading to fewer misleading visual structures; for SLIPMAP, noise looks like noise.

for noisy data. SLISEMAP works well for low-noise data, but in the presence of noise, it tends to cluster data in random clusters, as shown in Fig. 2.

The contributions of this paper are: we introduce a new prototype-based variant, coined SLIPMAP, that solves the scalability issues, define the computational problem, and present a simple modification to SLISEMAP that allows it to be a generative model that makes predictions (and is, therefore, an actual interpretable model). We show that SLIPMAP is fast, the modification results in a predictive model having good fidelity, and the explanations are stable even in the presence of noise, and valuable in practice.

Related Work. Starting at the introduction of ISOMAP in 2000 [30], countless *manifold visualisation* methods have been developed, of which t-SNE [22] and UMAP [23] are currently commonly used with several variants proposed (e.g.,

[15,16]). Manifold visualisations present high-dimensional data in a typically two-dimensional embedding such that neighbouring points in the embedding are similar by some pre-defined criteria. Unlike SLIPMAP and SLISEMAP, none of the prior methods defines the neighbourhood in terms of local explanations. Manifold visualisations are an indispensable tool in various disciplines where understanding of complex datasets is necessary, from genetics [11,18] to astronomy [3] and linguistics [19].

XAI is essential due to the increasing complexity and widespread use of black-box machine learning models. The primary objective in XAI is to understand and explore *black box* supervised learning algorithms [14]. Explanation methods can be divided into *model specific* and *model agnostic*, the latter of which can be applied to any supervised learning model.

XAI methods can further be split into *global* and *local*. Global methods try to explain the global behaviour of the supervised learning model for all data points. The obvious drawback of this approach is that if the black box model is too complicated, it is impossible to find a simple explanation that approximates it with sufficient fidelity. On the other hand, local explanations methods such as LIME [28], SHAP [21], and SLISE [5] produce an explanation that is valid only for individual data items. In this categorisation, SLIPMAP falls into the class of model-agnostic methods, which provide local explanations for all data points. However, combined with the embedding, the local explanations effectively produce a global explanation of the black-box model.

A common approach for local, model-agnostic explanation methods is to locally approximate the black box model with an interpretable model [5,7,21,28]. However, most other methods rely on randomly sampling new data points [21,28]. In contrast, SLIPMAP only uses the training data. As a result, SLIPMAP is especially useful for explaining models where random sampling of new data is not straightforward; e.g., with scientific data, generating random data that obeys all physical constraints is often challenging.

2 Problem Definition

In this section, we define the computational problem we want to solve in Sect. 2.1, and how we can get interpretable predictions for new data items, in Sect. 2.2.

2.1 SLIPMAP

The main difference between SLIPMAP and SLISEMAP is the introductions of "prototypes" in the embedding (the regular grid of circles in Fig. 1). In SLIPMAP, only the prototypes have local models instead of every data item, making the algorithm faster as we only need to optimise the parameters for a smaller number of prototypes, yielding a linear computational complexity (Sect. 3.1).

SLIPMAP also uses a Gaussian kernel instead of an exponential kernel (distances in the exponent are squared). The squared distances and the fixed spacing of the prototypes reduce the tendency of SLIPMAP to form clusters with random data. SLIPMAP solves the following optimisation problem:

Problem 1. (SLIPMAP) Assume you are given a dataset $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i\in[n]}$,[1] prototype vectors $\{\boldsymbol{c}_j\}_{j\in[p]}$, embedding dimensionality $d \in \mathbb{N}$ (typically $d = 2$), and a radius $r \in \mathbb{R}_{>0}$, where $\boldsymbol{x}_i \in \mathbb{R}^m$ are the vectors of features, $\boldsymbol{y}_i \in \mathbb{R}^o$ are the targets, and $\boldsymbol{c}_j \in \mathbb{R}^d$ are embedding coordinates. Find the embedding $\boldsymbol{z}_i \in \mathbb{R}^d$ and the local models $f_j : \mathbb{R}^m \to \mathbb{R}^o$, where $i \in [n]$ and $j \in [p]$, that minimise

$$\mathcal{L}_0 = \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{p} \frac{e^{-\|\boldsymbol{z}_i - \boldsymbol{c}_j\|_2^2}}{\sum_{k=1}^{n} e^{-\|\boldsymbol{z}_k - \boldsymbol{c}_j\|_2^2}} l(f_j(\boldsymbol{x}_i), \boldsymbol{y}_i), \tag{1}$$

where $\|\cdot\|_2$ is the Euclidean distance and $l(\cdot, \cdot)$ is a loss function for the local models under the constraint that

$$\text{radius}(\boldsymbol{Z}) = \left(\sum\nolimits_{i=1}^{n} \sum\nolimits_{k=1}^{d} \boldsymbol{z}_{ik}^2 / n\right)^{1/2} = r. \tag{2}$$

We use the following matrices: $\boldsymbol{X}_{i\cdot} = \boldsymbol{x}_i$, $\boldsymbol{Y}_{i\cdot} = \boldsymbol{y}_i$, and $\boldsymbol{Z}_{i\cdot} = \boldsymbol{z}_i$ for $i \in [n]$ and $\boldsymbol{C}_{j\cdot} = \boldsymbol{c}_j$ for $j \in [p]$. The rows $\boldsymbol{B}_{j\cdot}$ of matrix $\mathbf{B} \in \mathbb{R}^{p\times q}$ contain the parameters for the local models f_j, where q is the number of parameters in the local models. The loss function in Eq. (1) can be augmented with regularisation terms,

$$\mathcal{L} = \mathcal{L}_0 + \sum\nolimits_{j=1}^{p} \sum\nolimits_{k=1}^{q} (\lambda_{\text{lasso}} |\mathbf{B}_{jk}| + \lambda_{\text{ridge}} \mathbf{B}_{jk}^2), \tag{3}$$

where $\lambda_{\text{lasso}} \in \mathbb{R}_{\geq 0}$ and $\lambda_{\text{ridge}} \in \mathbb{R}_{\geq 0}$ are the parameters for Lasso and Ridge regularisation, respectively.

As local, interpretable models, we use linear models for regression problems and multi-variate logistic regression for classification problems. The loss functions are a quadratic loss for regression and Hellinger loss for classification; see [7] for details and discussion.

2.2 Mapping from Covariates to the Target Variable

Next, we define a mapping from the covariates to the embedding coordinates and the local models. In principle, these mappings could be arbitrary functions. Here, we have chosen the 1-nearest neighbour regression model as the mapping for simplicity and computational efficiency. The simplicity also makes the whole prediction procedure very transparent since we *"use an interpretable model that works well for similar data items"*.

The implied predictive model $f : \mathbb{R}^m \to \mathbb{R}^o$ for SLIPMAP is then the distance-weighted average over the local models in the embedding:

$$f(\boldsymbol{x}) = \sum\nolimits_{j=1}^{p} \frac{e^{-\|\boldsymbol{z}_i - \boldsymbol{c}_j\|_2^2}}{\sum_{k=1}^{p} e^{-\|\boldsymbol{z}_i - \boldsymbol{c}_k\|_2^2}} f_j(\boldsymbol{x}), \tag{4}$$

where $i = \arg\min_{i\in[n]} \|\boldsymbol{x} - \boldsymbol{x}_i\|_2$. We can define an equivalent mapping for SLISEMAP by replacing p with n and $\boldsymbol{c}_j$ by $\boldsymbol{z}_j$ in Eq. (4).

[1] We use shorthand notation $[n] = \{1, \ldots, n\}$.

Algorithm 1: The SLIPMAP algorithm, where $\mathcal{L}$ is given in Eq. (1). See the text for discussion.

1 **Function** `Slipmap(`$\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{C}$, r, d`)`
2 $\quad \boldsymbol{Z} \leftarrow \mathrm{PCA}(\mathbf{X})_{\cdot,1:d}$ `// Initialise the embedding`
3 $\quad \boldsymbol{Z} \leftarrow \boldsymbol{Z} \cdot r/\mathrm{radius}(\boldsymbol{Z})$ `// Normalise the embedding`
4 $\quad \boldsymbol{B} \leftarrow \arg\min_{\mathbf{B}}[\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{C}, \boldsymbol{B}, r, d)]$ `// Initialise the local models`
5 $\quad$ **do**
6 $\quad\quad \boldsymbol{Z} \leftarrow$ `Escape(`$\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{C}, \boldsymbol{B}, r$`)`
7 $\quad\quad \boldsymbol{Z}, \boldsymbol{B} \leftarrow \arg\min_{\boldsymbol{Z},\boldsymbol{B}} \mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \cdot r/\mathrm{radius}(\boldsymbol{Z}), \boldsymbol{C}, \boldsymbol{B}, r, d)$
8 $\quad$ **while** *not converged*
$\quad$ **Result:** $\boldsymbol{Z}$, $\boldsymbol{B}$

9 **Function** `Escape(`$\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{C}$, $\boldsymbol{B}$, r`)`
10 $\quad \boldsymbol{W}_{jk} \leftarrow e^{-\|\boldsymbol{c}_j - \boldsymbol{c}_k\|_2^2} / \sum_{l=1}^{p} e^{-\|\boldsymbol{c}_j - \boldsymbol{c}_l\|_2^2}$ for all $j, k \in [p]$
11 $\quad \boldsymbol{L}_{ij} \leftarrow l(f_j(\boldsymbol{X}_{i\cdot}), \boldsymbol{Y}_{i\cdot})$ for all $i \in [n]$ and $j \in [p]$
12 $\quad \boldsymbol{Z}_{i\cdot} \leftarrow \boldsymbol{C}_{k\cdot}$ where $k = \arg\min_k (\boldsymbol{L}\boldsymbol{W})_{ik}$ for all $i \in [n]$
$\quad$ **Result:** $\boldsymbol{Z} \cdot r/\mathrm{radius}(\boldsymbol{Z})$

3 Algorithm

This section discusses how we implement and solve Prob. 1, including the computational complexity in Sect. 3.1.

To optimise Eq. (1), we use the gradient-based quasi-Newton LBFGS optimiser [20]. We combine the optimiser with a heuristic for escaping local optima, just as with SLISEMAP [7]. The pseudocode can be seen in Alg. 1.

The algorithm starts by initialising the embedding for the data items and the local models for the prototypes (lines 2–4 in Alg. 1). Then, it alternates between the escape heuristic and the optimisation until no better solution is found (lines 6–8). The escape heuristic consists of greedily assigning each item the embedding of the prototype that minimises the weighted loss (lines 10–12).

SLIPMAP is implemented using PyTorch [26], which enables GPU acceleration. The source code for our implementation and experiments (Sect. 5) is available under an open-source licence at https://github.com/edahelsinki/slisemap.

3.1 Computational Complexity

The time complexity of Eq. (1) is $\mathcal{O}(np)$, not counting the time for evaluating a local model on one data item. For many simple models and loss functions, including linear and logistic regression, the time complexity increases by a factor of $\mathcal{O}(m+q+o)$. The optimisation contributes an unknown number of iterations, depending on the convergence difficulty. The memory complexity of Eq. (1) is $\mathcal{O}(npo + nm + pq)$, and the LBFGS optimisation only adds a constant factor for the history. The complexities are empirically evaluated in Sect. 5.4.

4 Datasets

We use the following datasets in the experiments (Sect. 5).

Air Quality [25] contains 7355 hourly instances of 12 different air quality measurements. One of the measurements is chosen as a dependent regression variable, and the others are used as covariates.

Covertype [8] is a classification dataset of forest cover types containing over half a million instances with 54 features and seven classes. The instances are various cartographic variables of natural forests.

Gas Turbine [1,17] is a regression dataset with 36,733 instances of 9 sensor measurements from a gas turbine to study gas emissions.

HIGGS [31] is a two-class classification dataset consisting of signal processes that produce Higgs bosons or are background. The dataset contains nearly 100,000 instances with 28 features.

Jets [10] contains simulated LHC proton-proton collisions. The collisions produce quarks and gluons that decay into cascades of stable particles called jets. The classification task is to distinguish between jets generated by quarks and gluons. The dataset consists of 266,421 instances with seven features.

QM9 [9] is a regression dataset comprising 133,766 small organic molecules. As the dependent variable, we use HOMO energies obtained from [12], and create interpretable features with the Mordred molecular description calculator [24].

5 Experiments

In this section, we empirically evaluate SLIPMAP by first comparing predictions on unseen data in Sect. 5.1. Then, we verify the embedding quality in Sect. 5.2 and local explanations in Sect. 5.3. Finally, we validate the claims about improved scaling in Sect. 5.4.

All experiments use normalised data (zero mean and unit variance). The density of the prototype grid is one prototype per unit square, and the regularisation coefficients λ_{lasso} and λ_{ridge} and the radius r have been optimised using Bayesian hyperparameter optimisation. All experiments have been run ten times with different seeds and randomly subsampled datasets. Since SLIPMAP is implemented with PyTorch [26], we run the experiments with GPU acceleration, except for the experiments measuring time.

5.1 Predictions

In this experiment, we measure the predictive performance of SLISEMAP and SLIPMAP, using Eq. (4). We also compare the predictions against the nearest neighbours to verify that the local models improve the predictions. As target values, we try both predictions from various black box models and the ground truth labels, with increasing subsamples of the training data.

In Fig. 3, we see how the losses from the SLIPMAP predictions on unseen test data approach that of the predictions from the black box models. In some cases,

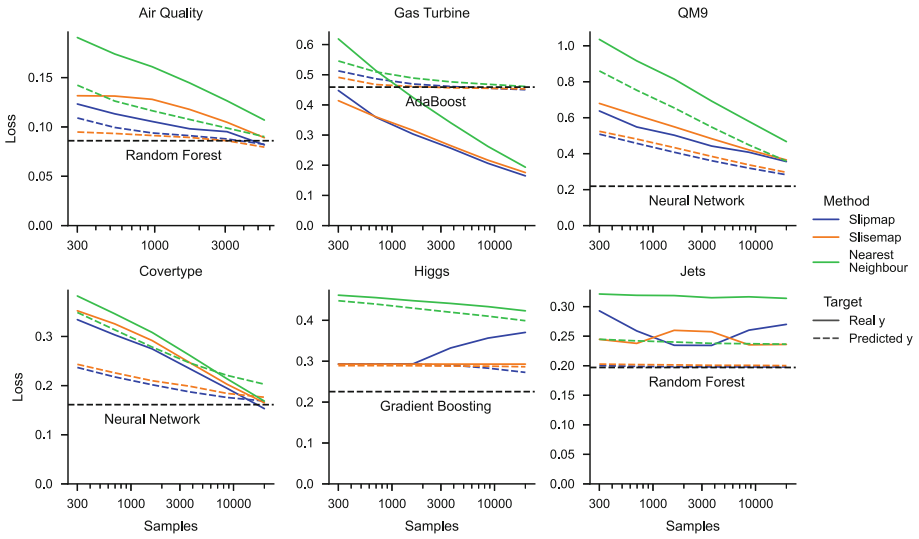

Fig. 3. Loss curves for SLIPMAP, SLISEMAP, and nearest neighbour models trained on predicted y:s and ground truth y:s compared to various black box models. The loss for regression (top row) is mean squared error; for classification (bottom row), the loss is Hellinger loss. Lower is better.

such as the *Jets* dataset, only very little data is needed. Predictions from black box models provide smoothing, especially for discrete class labels. However, with sufficient data, SLIPMAP trained on ground truth labels often converge to similar losses. The AdaBoost regressor is non-optimal for the *Gas Turbine* dataset since SLIPMAP and SLISEMAP, trained directly on the ground truth, actually provide better predictions. Generally, SLIPMAP performs slightly better than SLISEMAP and clearly better than the nearest neighbour.

5.2 Robustness

Explanations are only helpful if they are consistent. If, for example, slightly changing the training dataset causes a significant explanation shift, the explanations are less trustworthy.

Local model consistency [29] measures how stable the set of local models is with respect to resampling the data. If the local models are inconsistent, the local models are not trustworthy as explanations. To measure local model consistency, we train two models on subsamples taken from a dataset such that there is no overlap between the samples. This yields two sets of local models $\{f_1, f_2, ...f_p\}$ and $\{f'_1, f'_2, ..., f'_p\}$. We then match each local model to its most similar counterpart and calculate the average distance between the models:

$$\mathcal{M}_{\mathrm{B}} = 1 - \min_{\pi} \frac{\sum_{i=1}^{p} D(f_i, f'_{\pi(i)})}{\frac{1}{n} \sum_{i=1}^{p} \sum_{j=1}^{p} D(f_i, f'_j)}, \tag{5}$$

Table 1. Comparing local model consistency and neighbourhood stability. Here, we consider ten samples of 10^4 items for each dataset, using predictions from the black box models as labels. As the *Air Quality* dataset has less than 10^4 items, the missing items are generated by resampling the data. SLISEMAP and SLIPMAP show similar performance, and the best (highest) results are highlighted in bold.

Data	Local model consistency ↑		Neighbourhood stability ↑	
	SLIPMAP	SLISEMAP	SLIPMAP	SLISEMAP
Air Quality	0.460 ± 0.097	**0.530 ± 0.252**	**0.393 ± 0.062**	0.263 ± 0.061
Gas Turbine	**0.762 ± 0.051**	0.682 ± 0.190	**0.641 ± 0.039**	0.433 ± 0.103
QM9	0.328 ± 0.106	**0.443 ± 0.272**	**0.369 ± 0.086**	0.164 ± 0.036
Covertype	**0.540 ± 0.260**	0.348 ± 0.380	**0.301 ± 0.062**	0.276 ± 0.082
Higgs	**0.515 ± 0.193**	0.167 ± 0.376	0.604 ± 0.206	**0.771 ± 0.183**
Jets	0.662 ± 0.061	**0.865 ± 0.075**	0.382 ± 0.075	**0.523 ± 0.132**

where $D(f_i, f'_j) = \|\mathbf{B}_{i\cdot} - \mathbf{B}'_{j\cdot}\|_2$ is the Euclidean distance (similarity) between the local model parameters and π is the permutation minimising the distance between the local models.

Neigbourhood stability measures the stability of the embedding with respect to resampling. It measures how well models trained on partly overlapping data retain the relative locations of the data items in the embedding, i.e., whether or not the neighbouring relations between the items are preserved. To measure neighbourhood stability, we train models on datasets sampled such that half of the items overlap. Let $\mathcal{S}$ be the set of overlapping points. Then, for each shared item, we form the set of neighbours in both learned embeddings (denoted as $N(i) = \{j \in \mathcal{S} | \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 < 1\}$ and $N'(i) = \{j \in \mathcal{S} | \|\boldsymbol{z}'_i - \boldsymbol{z}'_j\|_2 < 1\}$) and calculate the Jaccard similarity between the neighbour sets:

$$\mathcal{M}_{neighbourhood} = |\mathcal{S}|^{-1} \sum\nolimits_{i \in \mathcal{S}} |N(i) \cap N'(i)| / |N(i) \cup N'(i)| \tag{6}$$

Table 1 shows a comparison between the explanation robustness of SLIPMAP and SLISEMAP. SLIPMAP performs comparably to SLISEMAP with respect to local model concistency and neighbourhood stability. As discussed in [7], local explanations have inherent ambiguity; a given data item can have multiple local explanations with comparable performance. The neighbourhood stability results show SLIPMAP also exhibits this behaviour.

5.3 Local Explanation Comparison

In this section we quantitatively compare the local models from SLIPMAP with other model-agnostic, local explanation methods: LIME [28] (with and without discretisation), (partition) SHAP [21], SLISE [5], and SLISEMAP [7]. These all provide explanations in the form of local, linear approximations. As metrics, we consider the following:

Time. How long does it take to get one explanation (dividing the setup time between the data items)?

Table 2. Comparing local explanation methods. We measure how well the approximation predicts the selected data item (local loss), the five nearest neighbours (stability), and the number of other data items with a loss less than a threshold. All explanations are based on 5,000 data items, and the best results are highlighted in bold.

Data	Method	Time (s) ↓	Local loss ↓	Stability ↓	Coverage ↑
Air Quality	LIME	3.648 ± 0.02	0.147 ± 0.07	0.180 ± 0.05	0.079 ± 0.00
	LIME (nd)	0.062 ± 0.02	0.041 ± 0.01	0.046 ± 0.01	0.464 ± 0.04
	SHAP	0.723 ± 0.21	**0.000 ± 0.00**	0.049 ± 0.02	0.217 ± 0.01
	SLISE	13.723 ± 2.36	**0.000 ± 0.00**	**0.019 ± 0.01**	**0.853 ± 0.01**
	SLIPMAP	**0.005 ± 0.00**	0.004 ± 0.00	**0.021 ± 0.01**	0.366 ± 0.01
	SLISEMAP	0.366 ± 0.14	0.001 ± 0.00	**0.018 ± 0.00**	0.759 ± 0.02
Gas Turbine	LIME	2.982 ± 0.02	0.205 ± 0.07	0.259 ± 0.07	0.219 ± 0.02
	LIME (nd)	0.030 ± 0.00	0.186 ± 0.07	0.180 ± 0.07	0.299 ± 0.04
	SHAP	0.693 ± 0.10	**0.000 ± 0.00**	0.116 ± 0.05	0.333 ± 0.03
	SLISE	26.577 ± 4.90	**0.000 ± 0.00**	**0.056 ± 0.02**	**0.407 ± 0.04**
	SLIPMAP	**0.007 ± 0.00**	0.004 ± 0.00	**0.052 ± 0.02**	0.325 ± 0.03
	SLISEMAP	0.482 ± 0.19	0.007 ± 0.00	**0.048 ± 0.01**	0.270 ± 0.04
QM9	LIME	6.256 ± 0.09	0.773 ± 0.23	0.778 ± 0.25	0.153 ± 0.02
	LIME (nd)	0.029 ± 0.00	0.323 ± 0.05	0.366 ± 0.04	0.179 ± 0.01
	SHAP	1.326 ± 0.79	**0.000 ± 0.00**	0.299 ± 0.07	0.207 ± 0.01
	SLISE	28.176 ± 3.71	**0.000 ± 0.00**	0.218 ± 0.04	**0.393 ± 0.01**
	SLIPMAP	**0.013 ± 0.00**	0.011 ± 0.00	**0.158 ± 0.03**	0.303 ± 0.02
	SLISEMAP	0.737 ± 0.22	0.016 ± 0.00	**0.160 ± 0.04**	0.292 ± 0.01
Higgs	LIME	8.425 ± 0.13	0.025 ± 0.00	**0.033 ± 0.00**	0.349 ± 0.01
	LIME (nd)	0.032 ± 0.00	0.034 ± 0.00	0.037 ± 0.00	0.333 ± 0.01
	SHAP	0.561 ± 0.04	**0.000 ± 0.00**	0.038 ± 0.00	0.315 ± 0.00
	SLISE	24.785 ± 2.35	**0.000 ± 0.00**	0.042 ± 0.00	**0.445 ± 0.01**
	SLIPMAP	**0.003 ± 0.00**	0.023 ± 0.01	0.040 ± 0.00	0.337 ± 0.03
	SLISEMAP	1.103 ± 0.35	0.034 ± 0.00	0.041 ± 0.00	0.302 ± 0.00
Jets	LIME	2.346 ± 0.02	0.011 ± 0.00	0.016 ± 0.00	0.106 ± 0.01
	LIME (nd)	0.063 ± 0.01	0.013 ± 0.00	0.014 ± 0.00	0.179 ± 0.01
	SHAP	0.246 ± 0.01	**0.000 ± 0.00**	0.007 ± 0.00	0.163 ± 0.01
	SLISE	10.357 ± 1.34	**0.000 ± 0.00**	**0.006 ± 0.00**	**0.431 ± 0.02**
	SLIPMAP	**0.018 ± 0.01**	0.000 ± 0.00	**0.006 ± 0.00**	0.357 ± 0.02
	SLISEMAP	1.956 ± 0.57	0.000 ± 0.00	**0.006 ± 0.00**	0.308 ± 0.02

Local loss. [5,21] How well does the approximation match the black box model for the selected data item using the losses mentioned in Sect. 2?

Stability. [2,5] Does a slight change in the input need a significant change in the explanation? Measured by calculating the mean loss of the local models on the five nearest neighbours.

Coverage. [7,13] Does the explanations generalise to other data items? Measured by counting the number of data items with a loss less than a threshold. The threshold is the 0.3 quantile of the losses from a global linear approximation.

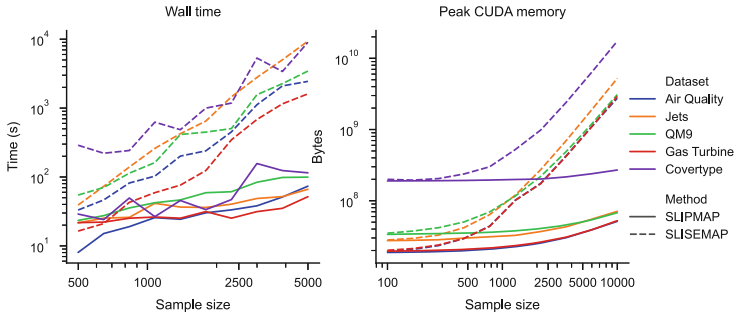

Fig. 4. Comparison of time and memory scaling between SLISEMAP and SLIPMAP. SLIPMAP is consistently faster as sample size increases and needs radically less memory in all six datasets (notice the logarithmic scale). Lower is better.

The results can be seen in Table 2 where SLIPMAP performs comparably or better than SLISEMAP. SLIPMAP does not guarantee a zero local loss like SLISE and SHAP, but they are generally quite small (whereas LIME sometimes have a smaller loss for the synthetic neighbourhood than the item being explained [5]).

5.4 Scaling

This experiment shows that SLIPMAP scales better with the number of data items than SLISEMAP, both in time and in memory. We measure the time on a CPU (to avoid the overhead of GPU communication on small data sizes) and the memory on a GPU, since that is usually the limiting factor SLISEMAP. As the left panel of Fig. 4 demonstrates, for each dataset, SLIPMAP converges faster than SLISEMAP by at least an order of magnitude. The difference is even more dramatic when considering memory complexity (Fig. 4 right panel), as SLIPMAP scales linearly (Sect. 3.1) compared to the quadratic scaling of SLISEMAP [7].

6 Conclusions and Future Work

We propose SLIPMAP, a novel model-agnostic method for explainable AI. SLIPMAP finds all local explanations for a complex black-box regression or classification model and produces an informative embedding where data items with similar explanations (local models) are embedded nearby. We substantially improved our earlier work by making our algorithm fast and robust to noise, leading to fewer false patterns in the embedding (Fig. 2). We have shown that the local explanations produced by SLIPMAP have high fidelity, good stability, and coverage. When trained on the predictions of the black-box model (instead of raw target values), SLIPMAP is, in our use cases, always able to mimic the black-box model with almost perfect fidelity.

Also, even though SLIPMAP is not meant to replace purpose-built classification and regression models, it performs similarly to the state-of-the-art models in real-world use cases.

SLIPMAP allows adding data items to the embedding and making predictions, even when the target variable is unknown, unlike the original SLISEMAP, extending the usage of SLIPMAP from a pure XAI method (which requires a pre-trained regression or classification model to work) to a more general supervised data exploration tool (which finds an interpretable predictive model for the data). In the future, we can still improve on SLIPMAP, e.g., by replacing the simple nearest neighbour model and kernel density estimate for making the predictions with a more general model, such as Gaussian Processes. The improved scalability, especially the GPU memory requirements, also unlocks applications with larger datasets.

Acknowledgement. We acknowledge funding from the Research Council of Finland (decisions 346376 and 345704) and the University of Helsinki.

References

1. Gas Turbine CO and NOx Emission Data Set (2019). https://doi.org/10.24432/C5WC95
2. Alvarez-Melis, D., Jaakkola, T.S.: On the Robustness of Interpretability Methods (2018). https://doi.org/10.48550/arXiv.1806.08049
3. Anders, F., et al.: Dissecting stellar chemical abundance space with t-SNE. Astron. Astrophys. **619**, A125 (2018). https://doi.org/10.1051/0004-6361/201833099
4. Besel, V., Todorović, M., Kurtén, T., Rinke, P., Vehkamäki, H.: Curation of high-level molecular atmospheric data for machine learning purposes. Tech. Rep. (2023). https://doi.org/10.5194/egusphere-egu23-1135
5. Björklund, A., Henelius, A., Oikarinen, E., Kallonen, K., Puolamäki, K.: Explaining any black box model using real data. Front. Comput. Sci. **5**, 1143904 (2023). https://doi.org/10.3389/fcomp.2023.1143904
6. Björklund, A., Mäkelä, J., Puolamäki, K.: SLISEMAP: combining supervised dimensionality reduction with local explanations. In: ECML PKDD, vol. 13718, pp. 612–616 (2023). https://doi.org/10.1007/978-3-031-26422-1_41
7. Björklund, A., Mäkelä, J., Puolamäki, K.: SLISEMAP: supervised dimensionality reduction through local explanations. Mach. Learn. **112**(1), 1–43 (2023). https://doi.org/10.1007/s10994-022-06261-1
8. Blackard, J.: Covertype (1998). https://doi.org/10.24432/C50K5N
9. Blum, L.C., Reymond, J.L.: 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. J. Am. Chem. Soc. **131**, 8732 (2009)
10. CMS Collaboration: Simulated dataset QCD_Pt-15to3000_TuneZ2star_Flat_8TeV_pythia6 in AODSIM format for 2012 collision data (2017). https://doi.org/10.7483/OPENDATA.CMS.7Y4S.93A0
11. Diaz-Papkovich, A., Anderson-Trocmé, L., Gravel, S.: A review of UMAP in population genetics. J. Hum. Gene. **66**(1), 85–91 (2021)
12. Ghosh, K.: MBTR_QM9 (2020). https://doi.org/10.5281/zenodo.4035918
13. Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., Giannotti, F.: Local Rule-Based Explanations of Black Box Decision Systems (2018). https://doi.org/10.48550/ARXIV.1805.10820
14. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5), 1–42 (2019). https://doi.org/10.1145/3236009

15. Heiter, E., Kang, B., Seurinck, R., Lijffijt, J.: Revised conditional t-SNE: looking beyond the nearest neighbors. In: IDA, vol. 13876, pp. 169–181 (2023)
16. Kang, B., García García, D., Lijffijt, J., Santos-Rodríguez, R., De Bie, T.: Conditional t-SNE: more informative t-SNE embeddings. Mach. Learn. **110**(10), 2905–2940 (2021). https://doi.org/10.1007/s10994-020-05917-0
17. Kaya, H., Tüfekci, P., Uzun, E.: Predicting CO and NOxemissions from gas turbines: novel data and a benchmark PEMS. Turk. J. Elec. Eng. Comp. Sci. **27**(6), 4783–4796 (2019). https://doi.org/10.3906/elk-1807-87
18. Kobak, D., Berens, P.: The art of using t-SNE for single-cell transcriptomics. Nat. Commun. **10**(1), 5416 (2019). https://doi.org/10.1038/s41467-019-13056-x
19. Levine, Y., et al.: SenseBERT: driving some sense into BERT. In: ACL, pp. 4656–4667 (2020). https://doi.org/10.18653/v1/2020.acl-main.423
20. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**(1–3), 503–528 (1989)
21. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: NeurIPS. vol. 30 (2017)
22. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(86), 2579–2605 (2008). http://jmlr.org/papers/v9/vandermaaten08a.html
23. McInnes, L., Healy, J., Saul, N., Großberger, L.: UMAP: uniform manifold approximation and projection. J. Open Source Softw. **3**(29), 861 (2018)
24. Moriwaki, H., Tian, Y.S., Kawashita, N., Takagi, T.: Mordred: a molecular descriptor calculator. J. Cheminform. **10**(1), 4 (2018)
25. Oikarinen, E., Tiittanen, H., Henelius, A., Puolamäki, K.: Detecting virtual concept drift of regressors without ground truth values. Data Min. Knowl. Discov. **35**(3), 726–747 (2021). https://doi.org/10.1007/s10618-021-00739-7
26. Paszke, A., Gross, S., Massa, F., et al.: PyTorch: an imperative style, high-performance deep learning library. In: NeurIPS. vol. 32 (2019)
27. Peng, G., Cheng, Y., Zhang, Y., Shao, J., Wang, H., Shen, W.: Industrial big data-driven mechanical performance prediction for hot-rolling steel using lower upper bound estimation method. J. Manuf. Syst. **65**, 104–114 (2022)
28. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": explaining the predictions of any classifier. In: ACM SIGKDD, pp. 1135–1144 (2016)
29. Seppäläinen, L., Björklund, A., Besel, V., Puolamäki, K.: Using slisemap to interpret physical data. PLoS ONE **19**(1), e0297714 (2024). https://doi.org/10.1371/journal.pone.0297714
30. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)
31. Whiteson, D.: HIGGS (2014). https://doi.org/10.24432/C5V312