

Evaluating Architectures and Hyperparameters of Self-supervised Network Projections

Tim Cech^a, Daniel Atzberger, Willy Scheibel^b, Rico Richter and Jürgen Döllner
Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam, Germany

Keywords: Dimensionality Reduction, Hyperparameter Optimization, Autoencoders.

Abstract: Self-Supervised Network Projections (SSNP) are dimensionality reduction algorithms that produce low-dimensional layouts from high-dimensional data. By combining an autoencoder architecture with neighborhood information from a clustering algorithm, SSNP intend to learn an embedding that generates visually separated clusters. In this work, we extend an approach that uses cluster information as pseudo-labels for SSNP by taking outlier information into account. Furthermore, we investigate the influence of different autoencoders on the quality of the generated two-dimensional layouts. We report on two experiments on the autoencoder's architecture and hyperparameters, respectively, measuring nine metrics on eight labeled datasets from different domains, e.g., Natural Language Processing. The results indicate that the model's architecture and the choice of hyperparameter values can influence the layout with statistical significance, but none achieves the best result over all metrics. In addition, we found out that using outlier information for the pseudo-labeling approach can maintain global properties of the two-dimensional layout while trading-off local properties.

1 INTRODUCTION

Dimensionality reduction algorithms (DR) are a class of unsupervised learning methods that aim to find a low-dimensional layout for a high-dimensional dataset. They are used as a basis for the visualization of high-dimensional data in various application domains (Espadoto et al., 2021b). Ideally, local properties, e.g., cluster membership, and global properties, e.g., cluster separation, of the high-dimensional dataset are preserved by a DR. In the case of datasets that carry an intrinsic dimensionality, manifold learning approaches are the preferred DR, as linear approaches, such as *Principal Component Analysis* (PCA), cannot meaningfully represent the data with only two or three dimensions (Jolliffe, 2005). Among the most popular manifold learning approaches are *t-distributed Stochastic Neighbor Embedding* (t-SNE) and *Uniform Manifold Approximation and Projection* (UMAP), as they are known to generate segregated clusters of high visual quality (van der Maaten and Hinton, 2008; McInnes et al., 2020).

However, those methods have limitations that make their application difficult (Espadoto et al.,

2021a). For example, the results are highly susceptible to the choice of parameters and do not allow inverse mapping. Deep Learning methods, such as *autoencoder* or *Neural Network Projections* (NNP), emerged from the field of artificial intelligence and offer alternative dimensionality reduction approaches for creating layouts (Hinton and Salakhutdinov, 2006; Espadoto et al., 2020b). They are compelling due to their ease of use and the possibility of handling data outside the training data. *Self-Supervised Network Projections* (SSNP), presented by Espadoto et al. (2021a), combine the qualities of manifold learning approaches and deep learning methods by incorporating neighborhood information of data points into the architecture of an autoencoder. For it, two kinds of data are combined: the feature data and pseudo-labels. In this work, pseudo-labels are labels determined by another machine learning algorithm. For the original SSNP, the pseudo-labels result from a clustering on the high-dimensional data space. Extending the loss function of an autoencoder, which reflects the preservation of the pseudo-labels, a two-dimensional representation of the data points is learned. Although the specific autoencoder architecture shows convincing results, it is an open question how the choice of pseudo-labels and the hyperparameters of the individual layers influence the results.

^a  <https://orcid.org/0000-0001-8688-2419>

^b  <https://orcid.org/0000-0002-7885-9857>

In this work, we evaluate different architectures and parameters for SSNP. In the work of Espadoto et al. (2021a), the pseudo-labels encode neighborhood information resulting from the cluster membership of a point. However, besides obtaining larger clusters, it is also desirable for a DR to obtain outliers. Therefore, we combine cluster membership and outlier information into an alternative pseudo-label approach. Through a random search, we further test the hyperparameters of the model, e.g., the number of training epochs or the size of the layers, on the results. We evaluate the influence of the parameters in two experiments on eight datasets using nine quality metrics.

2 RELATED WORK

Widespread DR are computationally intensive and require extensive hyperparameter tuning (Yang and Shami, 2020). Espadoto et al. (2020b) showed that a neural network can approximate a given DR. The neural network can be trained by a test dataset, and the results after applying a DR. Those approaches are called *Neural Network Projections* (NNP). The approximation by the neural network is faster, easier to use, and allows to map data points outside the training basis. The quality of the NNP can be improved by tuning the hyperparameters of the model (Espadoto et al., 2020a), taking neighborhood information, e.g., from a clustering algorithm, into account (Espadoto et al., 2021a), or sharpening the data distribution before applying the DR (Kim et al., 2022). Our work follows the idea of investigating the effect of the underlying architecture and parameters of the model on the results of the DR and mainly builds upon an NNP technique presented by Espadoto et al. (2021a). The authors suspected that Deep Learning algorithms produce worse cluster separation than traditional manifold learning approaches, as they do not consider neighborhood information. To address this issue, SSNP relies on an autoencoder that considers neighborhood information. Among the main advantages of autoencoders are their ease of use and their computational efficiency (Fournier and Aloise, 2019). Each data point is assigned a pseudo-label derived from a clustering algorithm in the first step. By modifying the loss function of an autoencoder, the encoder network is trained to learn a low-dimensional representation of the dataset that separates the clusters well, taking the pseudo-label into account.

Another desirable property of dimensionality reduction is the stability of the results under changes in the model's parameters and small changes in the data basis. Becker et al. (2020) verified the first property

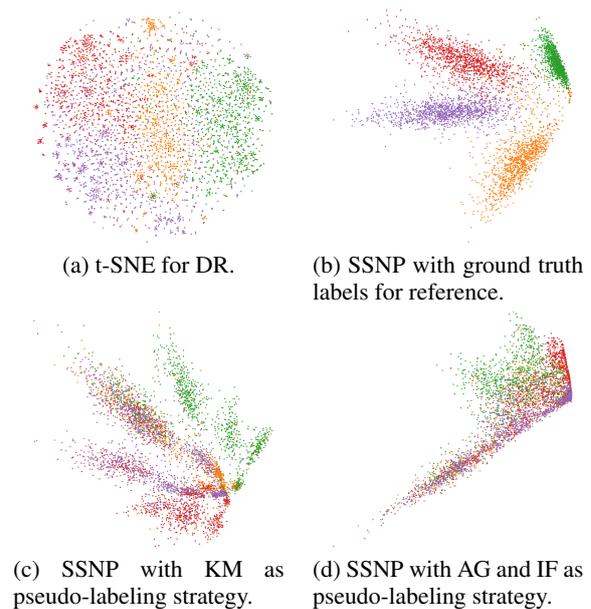


Figure 1: Example layouts for different configurations of the ag-news dataset. The color is derived from the ground truth labels. Figure 1b, and Figure 1a shows two reference images. Figure 1b is the result of SSNP with the ground truth labels followed by t-SNE. Figure 1c uses the simple KM pseudo-label strategy, and Figure 1d the complex KM with both AG and IF pseudo-label strategy. We see that the choice of the pseudo-labeling strategy can influence the layout considerably.

for Deep Learning approaches. Bredius et al. (2022) investigated stability with respect to changes in the data points. Explicitly, the authors evaluated Deep Learning algorithms after the data was undertaken different perturbations, e.g., translations, scaling, or permutations of dimensions of the dataset (Bredius et al., 2022). Their results showed that NNP can adapt to data modifications.

3 GRAY-BOX SSNP

Hyperparameters can influence the layout generated by an autoencoder considerably, as shown in Figures 1c and 1d. Therefore, we investigate, how the choice of hyperparameters influences the result (gray box). For it, we implemented a processing pipeline which enables us to perform experiments for evaluating the influence of several kinds of hyperparameters on specific datasets. In this section, we review the concepts we used in our processing pipeline (Figure 2). For it, we used standard techniques such as *term frequency-inverse document frequency transformation* (tf-idf). Additionally, we extended the pseudo-labeling approach proposed by Espadoto et al. (2021a) by com-

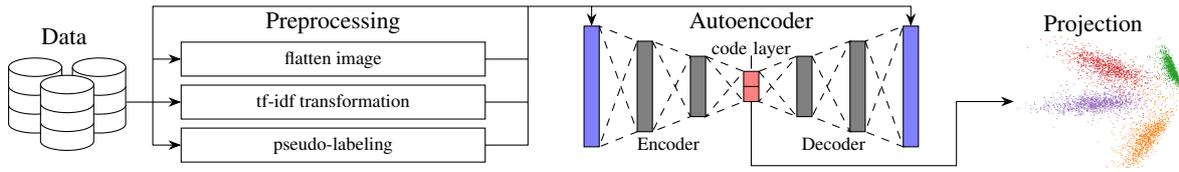


Figure 2: Our processing pipeline shows which preprocessing steps are undertaken before passing it to an autoencoder. The code layer represents the resulting projection.

binning the cluster information with outlier information via the Cantor pairing function.

Autoencoders. Autoencoders are a neural network architecture belonging to the self-supervised learning algorithms class, first presented by Hinton and Salakhutdinov (2006). They comprise three parts: an input layer, a set of hidden layers usually smaller than the input layer, and an output layer of the same size as the input layer. The inner state of the hidden layer is called the code. Figure 2 shows the general architecture of an autoencoder. The map that maps the input layer to the code is called the encoder, whereas the map that maps the hidden layer to the output layer is called the decoder. Given a training set $\mathcal{D} = \{x_1, \dots, x_N\} \subseteq \mathbb{R}^n$, the parameters of the model need to be adjusted in such a way that the composition $g \circ f$ of the encoder $f: \mathbb{R}^n \rightarrow \mathbb{R}^k$ and the decoder $g: \mathbb{R}^k \rightarrow \mathbb{R}^n$ approximates the identity map on \mathbb{R}^n . This adjustment is usually made by applying the backpropagation algorithm to the loss function

$$\sum_{i=1}^N (x_i - (g \circ f)(x_i))^2 \quad (1)$$

Usually, the dimension k of the code is much smaller than the dimension n of the input layer. Therefore the image $f(x)$ of a data point $x \in \mathbb{R}^n$ can be seen as a lower-dimensional representation of x . We choose $k = 2$ and interpret the encoder function results as a projection. For $k = 2$, the code can be visualized as a scatter plot to explore high-dimensional datasets.

Hyperparameter Tuning. We investigate the influence of hyperparameters on our researched datasets. We restrict our considerations to the hyperparameters which were explicitly set in the initial work of Espadoto et al. (2021a). We consider three kinds of hyperparameters: (1) hyperparameters that only influence the training of the autoencoder, such as patience, minimum delta, the number of training epochs and the pseudo-labeling strategy, (2) backpropagation-related hyperparameters, e.g., layer activation functions or optimizers, and (3) architecture-related hyperparameters, e.g., the number of layers. We used a grid search for the pseudo-labeling strategy and the model archi-

tecture. For all other hyperparameters, we used a random search.

Clustering and Outlier Mining Techniques. Clustering describes finding structures of dense data points in unlabeled data that are well separated. Specifically, a clustering algorithm learns a discrete function that maps similar data points to the same category (Espadoto et al., 2021a). Espadoto et al. (2021a) used the *k-Means algorithm* (KM) and *agglomerative clustering* (AG) for their pseudo-labeling approach. Besides pure clustering, we propose using labels from an outlier mining technique. In contrast to classical clustering, which measures the similarity between samples, outlier mining techniques find samples that are considered very unusual for the remaining data distribution (Liu et al., 2008).

We consider two outlier mining algorithms: *Isolation Forests* (IF) and the *Local Outlier Factor* (LOF). The LOF is an outlier mining technique focused on the sample’s environment (Breunig et al., 2000). If a sample is very dissimilar to its k nearest neighbors, it is classified as an anomaly. An IF is an outlier mining technique presented by Liu et al. (2008) that is similar to a Random Forest. For it, an attribute is repeatedly randomly selected and split so that, if possible, a sample is isolated. Samples that could already be isolated with a relatively shallow depth of the tree are considered outliers. In contrast to the LOF, the global properties of the dataset are also considered (Liu et al., 2008).

Pseudo-labels and Cantor Pairings. To provide the autoencoder with neighborhood information, Espadoto et al. (2021a) provided the data points with pseudo-labels resulting from the application of a clustering algorithm. We extend this idea by sampling different approaches to pseudo-labeling that emerge from clustering or outlier mining algorithms. We consider up to three different views on our data combined in one pseudo-label: The top-down *k-Means* view, the bottom-up *agglomerative clustering* view, and the view of an outlier mining technique. All possible combinations are listed in Table 1. By combining different techniques, the pseudo-labels describe a more comprehensive view of the data. For reference, we

Table 1: List of attributes with all possible values. We only show attributes that are subject to the random search. The initializer values use the following abbreviations: U for uniform distributed and N for normal distributed.

Attribute	Search	Investigated parameter values
Number of epochs	Random	10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000
Patience	Random	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Min. delta	Random	0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1
Clusters per class	Random	1, 2, 3, 4, 5
Initializers	Random	Glorot U, Glorot N, random U, random N, truncated N, He U
Layer activation function	Random	linear, tanh, sigmoid, softmax, softplus, softsign, elu, exponential
Code layer activation function	Random	linear, tanh, sigmoid, softmax, softplus, softsign
Optimizer	Random	Adam, SGD, RMSprop, Adadelta, Adagrad, Adamax, Nadam
L1 regularizer	Random	0.0, 0.01, 0.1
L2 regularizer	Random	0.0, 0.01, 0.1
Pseudo-label	Grid	KM, AG, KM/AG with IF/LOF, KM with AG and IF/LOF

capture the result of a basic autoencoder without any pseudo-labeling. For it, given the class labels x and y from two algorithms, we create a new pseudo-label by using the bijective *Cantor pairing function* (Lisi, 2007), given by:

$$\text{Pair} : \mathbb{N}^2 \rightarrow \mathbb{N}, (x, y) \mapsto \frac{(x+y)^2 + 3x + y}{2} \quad (2)$$

In the case of three labels, x, y , and z , we apply the Cantor pairing function first on x and y and then analogously pair the result with z , i.e., $\text{Pair}(\text{Pair}(x, y), z)$.

4 EXPERIMENTS

We investigate how the hyperparameters, model architectures, and pseudo-labels influence the result according to our evaluation metrics. We performed two experiments. First, we focused on how the pseudo-labels and the training duration influenced our result. For it, we investigated only hyperparameters that influence the training but not the model definition. Our second experiment investigates different model architectures and their related hyperparameters. We used the SSNP implementation most recently published by Kim et al. (2022). For more implementation details, we refer to our auxiliary material.

Training-related Hyperparameters. In this first experiment, we study training-related parameters. In

detail, we will sample from four hyperparameters with the values shown in Table 1. The *number of epochs* describes how often the training data is fed to the model. Too few epochs lead to underfitting while too many epochs can result in the overfitting problem without proper regularization. *Patience* describes a parameter that – together with the *minimum delta* – influences if the training is stopped early. If the model cannot improve over minimum delta accuracy over patience number of epochs the training is stopped. The *cluster per class* parameter times the number of classes determines how many cluster labels are present for our cluster-based pseudo-labeling strategy. Additionally, we perform a grid search on the pseudo-labeling strategy. In summary, we tested 100 different parameter configurations with a random search strategy (Bergstra and Bengio, 2012).

Backpropagation-related and Architecture-related Hyperparameters. We test seven model architectures with a grid search and sample backpropagation-related hyperparameter configurations with values as shown in Table 1 in this second experiment. In detail, we consider the following hyperparameters: (1) *number of layers* and their (2) *number of nodes*, (3) *initializer*, (4) *layer activation function*, (5) *optimizer*, and (6) *regularizers*. A larger number of layers offers more abstraction potential but is harder to train. The *initializer* sets the initial weights and biases for each node in the model. The *layer activation function* depends on the current state of the signal from the previous layer is fed into the next layer. For the code layer, we may use another activation function similar to the output layer of a neural network (Espadoto et al., 2021a). The *optimizer* determines how the model weights and biases are updated dependent on the old state of each node and the activation function. It guides the backpropagation of the model. *Regularizers* also influence the backpropagation and systematically force the network to not use information in order to avoid overfitting. We differentiate between an l1 (linear) and l2 (quadratic) regularization. The details of the seven model architectures can be found in the auxiliary material. We tested ten randomly selected parameter configurations according to the random search strategy (Bergstra and Bengio, 2012).

Evaluation Datasets. In our experiments, we extend the datasets provided by Espadoto et al. (2021a) by datasets provided by Atzberger et al. (2022) to validate the former results and shift the focus to natural language processing (NLP). In detail, the datasets are given by Table 2.

Table 2: Details of the evaluation datasets. We use the abbreviation FGI for “flatten grayscale image”.

Dataset	Data Points	Dimensions	Classes	Preprocessing
20-newsgroups tf-idf	16 695	23 959	20	tf-idf
ag-news tf-idf	19 175	20 860	4	tf-idf
fashion-MNIST	60 000	784	10	FGI
har	10 299	561	6	–
hatespeech tf-idf	24 783	8 176	2	tf-idf
imdb tf-idf	13 177	30 354	2	tf-idf
MNIST	60 000	784	10	FGI
reuters tf-idf	8 432	5 000	6	tf-idf

Evaluation Metrics. Here and in the following, we always refer to the cosine distance for distance measurement, as not otherwise mentioned, because – especially for very high-dimensional data – it usually captures the similarity between data samples better than euclidean distances (Atzberger et al., 2022). Espadoto et al. (2021a) used the following four metrics for evaluating a DR:

- *Trustworthiness* measures the number of points that are close to each other in the original dataset and after projection (Espadoto et al., 2021a).
- *Continuity* measures the number of points that are close to each other after projection, and which are also close to each other in the original dataset (Espadoto et al., 2021a).
- The *7-Neighborhood-Hit* counts how much data points the closest seven data points in the projection have the same label as the data point weighted by the overall presence of the label in the dataset (Kim et al., 2022).
- The *Shepard Diagram Correlation* measures how well, the dissimilarity matrix is preserved by the DR (Joia et al., 2011).

We furthermore capture the *normalized stress*, which approximates the squared error between the dissimilarity matrices in the high and low-dimensional space. Those metrics focus on the neighborhood of data samples and are therefore more concerned with local properties of the projection. In addition, we also measure more global properties of the data, which can be captured by clustering metrics (Kwon et al., 2018), specifically:

- The *Calinski-Harabasz index* measures the ratio of the mean of inter-cluster dispersion and the mean of intra-cluster dispersion (Caliński and Harabasz, 1974). The index requires the usage of euclidean metrics.

- The *Davies-Bouldin index* compares the similarity of each cluster to its most similar cluster (Davies and Bouldin, 1979). The index requires the usage of euclidean metrics.
- The *silhouette coefficient* of a data sample measures the maximal ratio between the mean distance of all data points within its cluster to the mean distance of all data points in the next nearest cluster (Rousseeuw, 1987).
- The *s_{dbw} validity index* takes the cluster compactness, separation, and density of clusters into account (Halkidi and Vazirgiannis, 2001).

We measured how well the data was clustered in the original data space using their ground truth labels and if the projection could preserve this clustering. We normalize and invert the measurements to the $[0, 1]$ interval with 1 as the best possible result.

Statistical Tests. The choice of a statistical test is dependent on the data distribution. For it, we have to verify whether or not the quality metrics are normally distributed. For it, we used the Quantile-Quantile-Plot (QQ-Plot), which can be found in our auxiliary material. The QQ-Plot reveals that the normal assumption is invalid. We, therefore, choose statistical tests that do not make any assumption on the underlying distribution. We test two different kinds of null hypotheses:

H_{0_1} : The pair of metric values and the increase of a parameter occurs at random.

To verify this null hypothesis, we apply the *Spearman correlation test* (Myers and Sirois, 2004). We can argue that the parameter significantly influences a metric by rejecting the null hypothesis.

H_{0_2} : The underlying distribution of metric values and parameter distribution is the same.

For the second null hypothesis, we use the *Wilcoxon* (Gehan, 1965), the *U-test* (MacFarland and Yates, 2016) and the *sign test* (Hodges, 1955). In this case, we aim to fail to reject the null hypothesis. In general, this does not imply that the null hypothesis is true but only implies insufficient evidence to reject it (Saxena et al., 2011). But in cases, where we already rejected the first null hypothesis and have a large sample size for each of our evaluation datasets, failing to reject the null hypothesis consistently may provide additional verification that the two samples are likely to originate from the same distribution (Makuch and Johnson, 1986).

5 RESULTS

Our results reveal that the choice of hyperparameters, especially the pseudo-labeling strategy, number of clusters, and regularizers, can significantly impact the layout. Therefore, they should be chosen carefully. In our first experiment, a more complex pseudo-labeling strategy that considered outlier information improved global metrics but decreased metrics related to neighborhood information. In our second experiment, the pseudo-labeling strategy had a weak negative correlation with the s_{dbw} validity index. The number of clusters had a modest impact on evaluation metrics, while regularization terms were mostly negatively correlated with evaluation metrics, suggesting that no regularization is needed. The best model architecture was found to be the one previously proposed by Espadoto et al. (2021a). For the full evaluation material we refer to our auxiliary material.

Training-related Hyperparameters. First, considering H_{0_1} , we observe that different pseudo-labeling strategies influence the projections, as shown in Figures 1c, and 1d. The autoencoder was trained 400 epochs, using 20 clusters per class with seven epochs and a minimum delta of 0.02 for early stopping. The deep learning approaches differ more from non-deep learning approaches as t-SNE, as shown in Figure 1a. Our metrics indicate that, in this specific case, the best pseudo-labeling strategy (besides using t-SNE) was SSNP with KM and IF.

In the first experiment, the number of clusters per class and the pseudo-labeling strategy were the most influential parameters, as shown in Table 3 (top). Increasing the number of clusters was correlated significantly with 8 out of 9 of our evaluation metrics ($p < 0.1\%$). The one metric that was correlated without significance was the Calinski-Harabasz index. The correlation was not positive in each case, meaning that a higher number of clusters positively influences the trustworthiness, continuity, Shephard diagram correlation, silhouette coefficient, Davies-Bouldin index, and s_{dbw} validity index while negatively impacting the 7-neighborhood hit and the normalized stress. Making the pseudo-labeling strategy more complex was significantly correlated to 5 out of 9 of our evaluation metrics at a significance level of 0.1%. Again, the correlation was not positive in each case. The pseudo-labeling strategy was positively correlated with the Shephard diagram correlation, the Calinski-Harabasz index, and the s_{dbw} validity index. Therefore, those metrics are positively influenced by choosing a more complex pseudo-labeling strategy that also considers outlier information. The

Table 3: Full list of significantly correlated parameter-metric pairs at significance level 0.1% (top) and excerpt from the significantly correlated parameter-metric list of our second experiment (bottom). Abbreviations: The parameter value n_cluster refers to the number of clusters per class, l1 refers to l1 regularization, l2 to l2 regularization, and label to our pseudo-labeling strategy. The metric C refers to the continuity metric, D to the Davies-Bouldin index, H to the Calinski-Harabasz index, N to the 7-Neighborhood hit, S to the normalized stress, SDC to the the Shephard diagram correlation, SC to the silhouette coefficient, s_dbw to the s_{dbw} validity index and T to trustworthiness.

Parameter	Metric	Significance Level				
		Wilcoxon p	U test p	Sign test p	Spearman statistic	Spearman p
Experiment 1						
n_cluster	C	0.00	0.00	0.00	0.36	0.00
n_cluster	D	0.00	0.00	0.00	0.24	0.00
n_cluster	N	0.00	0.00	0.00	-0.13	0.00
n_cluster	S	0.00	0.00	0.00	-0.27	0.00
n_cluster	SC	0.00	0.00	0.00	0.18	0.00
n_cluster	SDC	0.00	0.00	0.00	0.27	0.00
n_cluster	s_dbw	0.00	0.00	0.00	0.30	0.00
n_cluster	T	0.00	0.00	0.01	0.25	0.00
label	H	0.03	0.01	0.00	0.15	0.00
label	N	0.13	0.85	0.00	-0.19	0.00
label	S	0.00	0.04	1.00	-0.10	0.00
label	SDC	0.04	0.86	0.77	0.06	0.00
label	s_dbw	0.78	0.02	0.06	0.04	0.00
Experiment 2						
l1	C	0.31	0.80	0.61	-0.26	0.00
l1	N	0.72	0.39	0.71	-0.18	0.00
l1	s_dbw	0.01	0.25	0.03	-0.13	0.00
l2	C	0.95	0.14	0.29	-0.12	0.00
l2	N	0.99	0.97	0.73	-0.12	0.00
l2	s_dbw	0.97	0.14	0.51	-0.06	0.00
l2	SDC	0.58	0.60	0.71	-0.07	0.00
l2	T	0.62	0.04	0.13	-0.10	0.00
label	C	0.75	0.55	0.94	-0.10	0.00
label	N	0.75	0.49	0.24	-0.13	0.00
label	H	0.01	0.00	0.00	0.08	0.00
label	SDC	0.79	0.68	0.09	-0.08	0.00
label	T	0.94	0.86	0.96	-0.09	0.00

s_{dbw} validity index was maximized when using the complex SSNP with KM and both AG and IF or SSNP with KM, AG, and LOF pseudo-labeling strategy. In contrast, the other positively correlated metrics were mainly maximized by using the strategy that involves KM and LOF. In contrast, the normalized stress and the 7-neighborhood hit were negatively impacted by choosing a more complex pseudo-labeling strategy. The 7-neighborhood hit was optimized using a simple pseudo-labeling strategy, and the normalized stress

was optimized using no pseudo-labeling strategy.

Architecture-related Hyperparameters. Considering H_{0_2} , the pseudo-labeling strategy remains significant at a 0.1% significance level for the Shephard diagram correlation and the s_{dbw} validity index. For higher significance levels, they also agree for patience together with the 7-neighborhood hit or trustworthiness at a significance level of 1%. The other correlation tests, especially the sign test, differ with the remaining tests and would reject the null hypothesis for any correlation between the number of clusters per class and metric.

In our second experiment, we found out that the model architecture did not significantly correlate with any evaluation metric for the tested parameter configuration for a significance level of at least 10%. The best results were achieved with the architecture already proposed by Espadoto et al. (2021a). Furthermore, we found out that other model-related parameters, like the amount of 11 or 12 regularization, are (mostly negatively) significantly correlated with many evaluation metrics, as shown in Table 3 (bottom). The amount of 11 regularization correlated significantly ($p < 0.1\%$) with 7 out of 9 evaluation metrics. The 12 regularization was even for 8 out of 9 evaluation metrics ($p < 0.1\%$). Both regularization terms did not correlate with the Calinski-Harabasz index. The 12 regularization term additionally did not correlate with the silhouette coefficient. In contrast to our first experiment, our second type of hypothesis test agreed with more cases of correlation found. The 11 and 12 regularization still correlated significantly ($p < 0.1\%$) with 5 out of 9 of our evaluation metrics. For one up to three evaluation metrics, the choice of the optimizer, the layer activation function, and the initializers correlated significantly. As before, the pseudo-labeling strategy significantly influenced many evaluation parameters Table 3 (bottom). Notably, the choice of hyperparameters may lead to a degenerated projection where all data points are projected around a single point, which makes the points impossible to differentiate.

Threats to Validity. Several internal threats of validity limit the results presented above. First, we have focused our investigations on hyperparameters that were also present in previous work – mainly from Espadoto et al. (2021a) – and the tested number of hyperparameter configurations was limited by the design of our experiments. In particular, we introduced a bias into our experiments for the chosen hyperparameter values. Second, our results are limited to the tested data sets. Following the no-free-lunch theorem,

our results may not be applicable in another domain for other datasets (Adam et al., 2019). Third, our choice of statistical tests introduced further bias because three out of our four statistical tests aimed at failing to reject the null hypothesis instead of proving the alternative hypothesis. But following Makuch and Johnson (1986), a reasonably large sample size would allow us to deduce that the null hypothesis could be true. We mitigated this bias by establishing a significant correlation according to the Spearman correlation test. Furthermore, we used a reasonably large sample size of over 200 000 samples.

External factors could also threaten our results. First, our implementation and analysis may be subject to software bugs. However, we mitigate this risk by inheriting publicly available source code and software. Second, the used model relies on random number generators. We mitigated this risk by setting the random seed everywhere applicable.

6 CONCLUSIONS

In this work, we reiterated the SSNP approach for dimensionality reduction. For one, we extended the original pseudo-labeling approach by considering outlier labels and pairing them with clustering labels. Furthermore, we designed two experiments testing different hyperparameter configurations, including the extended pseudo-labeling approach. We measured nine evaluation metrics, i.e., five local and four global metrics that consider local neighborhood and global clustering properties, respectively.

Our results indicate that the architecture chosen by Espadoto et al. (2021a) is adequate. Furthermore, the choice of a pseudo-labeling strategy, regularization, and the number of clusters per class influence evaluation metrics significantly. However, the correlation is ambiguous. Most of the time, global metrics are optimized by using a more complex pseudo-labeling strategy while local metrics are traded-off.

We propose that future work investigates further hyperparameter configurations, especially additional compounded pseudo-labeling strategies. Further, we aim to build a visualization that guides the user in choosing a hyperparameter configuration.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their valuable feedback. This work was partially funded by the German Ministry for Education and Research (BMBF)

through grants 01IS20088B (“KnowhowAnalyzer”) and 01IS22062 (“AI research group FFS-AI”).

REFERENCES

- Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M., and Vrahatis, M. N. (2019). No free lunch theorem: A review. In *Approximation and Optimization: Algorithms, Complexity and Applications*, pages 57–82. Springer.
- Atzberger, D., Cech, T., Scheibel, W., Limberger, D., Döllner, J., and Trapp, M. (2022). A benchmark for the use of topic models for text visualization tasks. In *Proc. VINCI '22*, pages 17:1–4. ACM.
- Becker, M., Lippel, J., Stuhlsatz, A., and Zielke, T. (2020). Robust dimensionality reduction for data visualization with deep neural networks. *Graphical Models*, 108:101060:1–15.
- Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *JMLR*, 13(10):281–305.
- Bredius, C., Tian, Z., Telea, A., Mulawade, R. N., Garth, C., Wiebel, A., Schlegel, U., Schiegg, S., and Keim, D. A. (2022). Visual exploration of neural network projection stability. In *Proc. MLVIS '22*, pages 1068:1–5. EG.
- Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). LOF: Identifying density-based local outliers. *SIGMOD Record*, 29(2):93–104.
- Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27.
- Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *TPAMI*, 1(2):224–227.
- Espadoto, M., Hirata, N. S., Falcão, A. X., and Telea, A. C. (2020a). Improving neural network-based multidimensional projections. In *Proc. IVAPP '20*, pages 29–41. INSTICC, SciTePress.
- Espadoto, M., Hirata, N. S., and Telea, A. C. (2021a). Self-supervised dimensionality reduction with neural networks and pseudo-labeling. In *Proc. IVAPP '21*, pages 27–37. INSTICC, SciTePress.
- Espadoto, M., Hirata, N. S. T., and Telea, A. C. (2020b). Deep learning multidimensional projections. *Information Visualization*, 19(3):247–269.
- Espadoto, M., Martins, R. M., Kerren, A., Hirata, N. S. T., and Telea, A. C. (2021b). Toward a quantitative survey of dimension reduction techniques. *TVCG*, 27(3):2153–2173.
- Fournier, Q. and Aloise, D. (2019). Empirical comparison between autoencoders and traditional dimensionality reduction methods. In *Proc. AIKE '19*, pages 211–214. IEEE.
- Gehan, E. A. (1965). A generalized Wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika*, 52(1–2):203–224.
- Halkidi, M. and Vazirgiannis, M. (2001). Clustering validity assessment: finding the optimal partitioning of a data set. In *Proc. ICDM '01*, pages 187–194. IEEE.
- Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.
- Hodges, J. L. (1955). A bivariate sign test. *The Annals of Mathematical Statistics*, 26(3):523–527.
- Joia, P., Coimbra, D., Cuminato, J. A., Paulovich, F. V., and Nonato, L. G. (2011). Local affine multidimensional projection. *TVCG*, 17(12):2563–2571.
- Jolliffe, I. (2005). Principal component analysis. In *Encyclopedia of Statistics in Behavioral Science*. John Wiley & Sons, Ltd.
- Kim, Y., Espadoto, M., Trager, S., Roerdink, J. B., and Telea, A. (2022). SDR-NNP: Sharpened dimensionality reduction with neural networks. In *Proc. IVAPP '22*, pages 63–76. INSTICC, SciTePress.
- Kwon, B. C., Eysenbach, B., Verma, J., Ng, K., De Filippi, C., Stewart, W. F., and Perer, A. (2018). Clustervision: Visual supervision of unsupervised clustering. *TVCG*, 24(1):142–151.
- Lisi, M. (2007). Some remarks on the cantor pairing function. *Le Matematiche*, 62(1):55–65.
- Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *Proc. ICDM '08*, pages 413–422. IEEE.
- MacFarland, T. W. and Yates, J. M. (2016). Mann–whitney U test. In *Introduction to Nonparametric Statistics for the Biological Sciences Using R*, pages 103–132. Springer.
- Makuch, R. W. and Johnson, M. F. (1986). Some issues in the design and interpretation of “Negative” clinical studies. *Archives of Internal Medicine*, 146(5):986–989.
- McInnes, L., Healy, J., and Melville, J. (2020). UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv CoRR*, stat.ML(1802.03426). pre-print.
- Myers, L. and Sirois, M. J. (2004). Spearman correlation coefficients, differences between. In *Encyclopedia of Statistical Sciences*. John Wiley & Sons, Ltd.
- Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.
- Saxena, D., Yadav, P., and Kantharia, N. (2011). Nonsignificant P values cannot prove null hypothesis: Absence of evidence is not evidence of absence. *Journal of Pharmacy and Bioallied Sciences*, 3(3):465–466.
- van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *JMLR*, 9(11):2579–2605.
- Yang, L. and Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316.

APPENDIX

The auxiliary material is available under 10.5281/zenodo.7501914 and contains implementation details, the QQ-Plots, and all evaluation results.