UNDER REVIEW

RESCIENCE C

Replication / ML Reproducibility Challenge 2022

[Re] Reproducibility study of "Label-Free Explainability for Unsupervised Models"

Gergely Papp[1,2], Julius Wagenbach[1], Laurens Jans de Vries[1] Niklas Mather[1]

[1]University of Amsterdam, Amsterdam, the Netherlands – [2]Alfréd Rényi Institute of Mathematics, Budapest, Hungary

Edited by
CBYa

Reviewed by
CYnV
nrXi

Received
03 February 2023

Published
–

DOI
–

A replication of crabbe2022label.

Reproducibility Summary

Scope of Reproducibility – In this work, we present our reproducibility study of *Label-Free Explainability for Unsupervised Models* [1], a paper that introduces two post-hoc explanation techniques for neural networks: (1) label-free feature importance and (2) label-free example importance. Our study focuses on the reproducibility of the authors' most important claims: (i) perturbing features with the highest importance scores causes higher latent shift than perturbing random pixels, (ii) label-free example importance scores help to identify training examples that are highly related to a given test example, (iii) unsupervised models trained on different tasks show moderate correlation among the highest scored features and (iv) low correlation in example scores measured on a fixed set of data points, and (v) increasing the disentanglement with β in a β-VAE [2] does not imply that latent units will focus on more different features.

Methodology – The authors uploaded their code when they published the paper. We reviewed the authors' code, checked if the implementation of experiments matched with the paper, and also ran all experiments. Moreover, we extended the codebase in order to run the experiments on more datasets, and to test the claims with other experiments. Our code is available at https://anonymous.4open.science/r/5974660645.

Results – We found that all of the main claims of the paper were reproducible. However, when we repeated the same experiments on two new datasets, we found that there was a much higher correlation in example scores across different tasks (point iv above).

What was easy – The published code was high quality, well-documented and ran the experiments end to end. The paper introduced the relevant theory well.

What was difficult – The code contained a few minor bugs we needed to fix first. Some parts of the code were written specifically for MNIST and therefore we could not extend the experiments easily with new datasets.

Communication with original authors – We contacted the authors to clarify our understanding of some details in the methods. They responded quickly and answered all questions.

Copyright © 2023 G. Papp et al., released under a Creative Commons Attribution 4.0 International license.
Correspondence should be addressed to Gergely Papp (gergely.papp@student.uva.nl)
The authors have declared that no competing interests exists.
Code is available at https://github.com/jwagenbach/FACT/ – DOI 10.5281/zenodo.7895731.
Open peer review is available at https://openreview.net/forum?id=n2qXFXiMsAM.

1 Introduction

An enduring problem in machine learning is explaining why a trained 'black box' model behaves as it does. Knowing this is important because it gives insight into the modelling process, and allows users to trust the model outputs as being fair and sensible. As a result, several techniques have been devised to explain the outputs of supervised models. However, these techniques are only usable when there is a target function to be explained, and therefore cannot be used in unsupervised settings. Crabbé and Schaar [1] introduced methods for extending existing techniques from the supervised setting to the unsupervised one.

This paper attempts to replicate and extend the findings of Crabbé and Schaar [1]. Specifically, we:

- Reproduce the findings of the paper using their provided codebase, and identify which factors (e.g. code quality, clarity of description) hampered or helped this reproduction effort.
- Test whether their proposed methods behave consistently on more complex datasets.
- Extend their investigation by looking at whether the methods they devised for the unsupervised case are consistent with results obtained when supervision signals are available.

1.1 Label-Free Explainability for Unsupervised Models

Crabbé and Schaar [1] introduce three broad classes of methods for explaining unsupervised models.

- **Feature importance:** feature importance methods aim to describe how each input feature contributes to a given prediction. Mathematically, such a method can be described as a scalar function $a_i(f_j, x)$ which assigns a scalar value to the importance of the ith feature of the jth output of a model f for a specific input vector x. There are many existing methods (i.e. possible choices of function a_i) for the supervised case (e.g. Lime [3], Shap [4], Salience [5], Integrated Gradients [6]), but they cannot be used for unsupervised models because there is no obvious choice of output function f_j. Crabbé and Schaar [1] argue that any linear feature importance method can be extended to the unsupervised case by substituting the inner product on the latent space for the output function. That is, the score assigned to the ith feature for the output of a model f and sample x is defined as $b_i(f, x) = a_i(g_x, x)$ where $g_x(\tilde{x}) \coloneqq \langle f(x), f(\tilde{x}) \rangle$.

  Intuitively speaking, this method ascribes high importance to features which induce a large change in position in the latent space.

Example importance methods describe which elements of the training data were most influential to a particular prediction. These can be further divided into two categories.

- **Loss-based example importance:** These methods assess the importance of a given training example to the prediction made for a test example by estimating how the loss function would change if the training example were removed, the model retrained, and new parameters obtained. The authors extend existing methods to measuring the importance of the encoder in an encoder/decoder architecture by only considering loss changes due to the encoder's parameters.

- **Representation-based example importance:**
  For a given test point $\mathbf{x}$, these methods assign an importance score to a training example $\mathbf{x}'$ based on how similar the images $f(\mathbf{x})$ and $f(\mathbf{x}')$ are. The authors argue that these methods do not need any modification to be used in the unsupervised case, since the same procedure can be applied if f maps to a latent space.

2 Scope of reproducibility

Crabbé and Schaar [1] broadly make three types of claims in their paper. Below we describe each category as well as which hypotheses are specifically tested in each one.

Area 1: Consistency — They aimed to show that their suggested feature- and example-importance methods identify features and examples that are consistent with a 'sensible' definition of importance. They make two concrete claims in this area.

- Claim 1.1: Masking features (i.e. replacing them with a baseline value) identified as the most important by their feature importance method will induce a larger representational shift (change in position in the latent space) than masking features selected at random.
- Claim 1.2: For a given test datapoint, ordering a random set of training datapoints by their unsupervised example importance will place training points of the same class first.

Their other claims relate to the use of their new techniques to study the behaviour of latent representations.

Area 2: Correlation — They train separate latent representations for different tasks, and compare the behaviour of feature- and example- importance methods across the latent representations.

- Claim 2.1: The label-free importance scores assigned to features on the MNIST dataset are moderately correlated across tasks, and there is no evidence that this correlation is lower when comparing supervised and unsupervised models.
- Claim 2.2: Label-free example importance scores have low correlation across different tasks and this correlation is slightly lower when comparing supervised and unsupervised models.

Area 3: Disentanglement — They use their method to study whether disentangled β-VAE's (which have a regularisation parameter that penalises correlation between their latent dimensions) see reduced correlations between the importance of a feature to each latent dimension as the regularisation is increased.

- Claim 3: Increasing β does not imply a decrease in correlation between the feature importance scores of separate latent units.

3 Methodology

To replicate the study, we relied on the public repository provided by the authors. This included scripts to run each of the experiments, which we adapted for our use with minor modifications to fix small errors. We also modified the code for improved parallelisation. The scope and impact of these modifications is described in Appendix B. We also conducted some analyses that went beyond their paper. Both the adapted code and our experiments can be found at https://anonymous.4open.science/r/5974660645/README.md.

3.1 Model descriptions

Different models are used depending on the dataset and experiment. The first model is a denoising autoencoder CNN, which is used on MNIST, CIFAR-10 and CIFAR-100. The second model is an LSTM reconstruction autoencoder, which is used on ECG5000. The third model is a SimCLR neural network with a ResNet18 backbone, this model is used for experiments on the CIFAR-10 and CIFAR-100 datasets. The last model is a disentangled β-VAE, this is used for experiments on MNIST dSprites, CIFAR-10 and CIFAR-100.

3.2 Datasets

The experiments in the original paper were conducted using four datasets: MNIST, dSprites, ECG5000 and CIFAR-10. For additional experiments CIFAR-100 was used.
MNIST[7] consists of 60000 training and 10000 testing images. These images are 28x28 and in greyscale with pixel values ranging from 0 to 255. Each image represents a handwritten digit ranging from 0 to 9, which is also labelled correspondingly.
dSprites[8] is a dataset of procedurally generated 2D shapes. These shapes are based on 6 ground truth independent latent factors. The dataset contains 737280 64x64 images with 6 dimensional float64 values of the latent factors. The dataset was split into a training set and a test set, with a ratio of 0.9 for training and 0.1 for the test set.
ECG5000[9] consists of 5000 univariate time series describing the heartbeat of a patient. This is equivalent to around 20 hours of real-time heartbeats. Each time series describes a single heartbeat in 140 time steps. Each heartbeat is also labelled with a 0 or a 1, indicating if the heartbeat is normal(0) or abnormal(1). 4500 time series are used for testing and 400 for training, 100 are used for validation.
CIFAR-10[10] consists of 60000 32x32 colour images with 10 classes, with 6000 images per class. The dataset is split in 5 training batches and one testing batch, with each batch consisting of 10000 images. Each image contains a label in the form of an integer ranging from 0 to 9.
CIFAR-100[10] is similar to CIFAR-10, the difference is that it contains 100 classes with 600 images for each class. These 100 classes can be grouped together in 20 superclasses. Each superclass is a group of 5 classes, for example: maple, oak, palm, pine, willow form the superclass trees.

3.3 Hyperparameters

The reproduction experiments are conducted using the same model hyperparameters as in the original paper. Table 5 in the Appendix D specifies the hyperparameters for each used model.

3.4 Experimental setup and code

Methodology - Claim 1.1 — This experiment measured the impact of replacing the values of the top-ranked most important features with random baselines. For each of three attribution methods (Saliency, Integrated Gradients, and Gradient SHAP), we calculated unsupervised feature importance values for all features. Then, for various values of $M \in \mathbb{N}$, we replaced (or 'masked') the top M features with a fixed baseline value, and measured the change in position in the latent space. This distance is referred to as the *representation shift*, and is calculated as $||f_e(x) - f_e(\mathbf{m}(x))||$ where f_e is our encoder, and $\mathbf{m}$ is our masking function.

The authors hypothesised that if an unsupervised feature attribution method was 'sensible', then masking the top M highest ranked features should always induce a larger representation shift than when the masked features were chosen at random.

Methodology - Claim 1.2 — This experiment measured whether, for a given test example x^*, the top-ranked most similar training examples were more likely to share class labels with x^*. For a given value of $M \in \mathbb{N}$, we computed what proportion of the top M most similar training examples shared the same class label as x^* - this proportion is referred to as the *similarity rate*. We repeated this process for 1000 separate test examples, and averaged the resulting proportions. The authors hypothesised that a 'sensible' example similarity method should see higher values of this proportion at lower values of M.
This process was repeated for each of three datasets (CIFAR-10, MNIST, ECG5000) and five example importance methods (DKNN [1], SimplEx [11], TracIn [12], Influence Functions [13], CosineNN). The 'CosineNN' method was created by us as a drop-in replacement for DKNN. It uses cosine similarity, rather than inverse distance, to define similarity between train and test points.

Methodology - Claim 2.1 — This experiment compared feature importance for latent representations trained for different tasks on the MNIST dataset. A model was trained on each of the three unsupervised tasks of reconstruction, denoising, and inpainting, and the supervised task of classification. Label-free gradient Shap was used to score the importance of each feature to each latent representation. Then, we calculated the Pearson correlation coefficient of the importance scores of matching features between pairs of latent representations. That is, if $\mathbf{f}$ and $\mathbf{g}$ were two encoders corresponding to different latent representations, and b_i the label free gradient Shap of the ith feature, we calculated the correlation between $b_i(f, x)$ and $b_i(g, x)$ across all choices of i.

Methodology - Claim 2.2 — This experiment had the same structure as 2.1, but measured correlations in example importance (measured using the label-free DKNN method) instead. That is, if $c^n(f, x_m)$ refers to the example importance of the nth training point to the mth test point, and $\mathbf{f}$, $\mathbf{g}$ referred to encoders learned for different tasks, then we measured the correlation of $c^n(\mathbf{f}, x_m)$ and $c^n(\mathbf{g}, x_m)$ pooled across all n and m.

Methodology - Claim 3 — This experiment aimed to use label-free feature importance methods to study the behaviour of disentangled VAEs, a VAE architecture where a regularisation parameter (β) penalises correlation between variables in the learned latent space. The authors initially hypothesised that this implied that as the parameter was increased, the correlation of label-free feature importance scores between different units should decrease. Ultimately, they found no evidence to support their initial hypothesis. To test this hypothesis, we compute feature importance scores separately for each latent unit, then computed the Pearson correlation coefficient of corresponding scores between the latent units.
We also attempted to corroborate the authors' explanation for their findings by visualising the behaviour of each latent unit for multiple values of β using Lucid [14], a method which determines the input that would maximise the response from each latent unit.

Additional Experiment - Extending consistency and task comparison to CIFAR-10 and CIFAR-100 — We repeated the experiments described above on two extra datasets: CIFAR-10 (for which 1.1 and 1.2 were already tested in the original paper) and CIFAR-100 datasets (which is entirely new). We had two aims: first, we wanted to check whether the label-free methods would still be consistent on these more complex datasets. Second, we wanted to check whether there would be a larger difference between the latent representations of models that were trained with and without access to labels, since the labels may provide a more important signal for training representations when the data is more complex. Due to the increased complexity of the task, we also added one extra layer both to the encoder and decoder and increased the network's width. The details of the encoder and decoder networks are described in Appendix C.

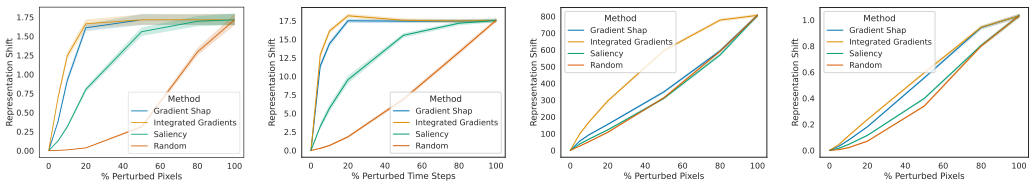
Additional Experiment - Comparison of Unsupervised and Supervised Feature Importance — We ran an additional experiment to corroborate their claims of **consistency** for feature importance. Their experiments showed that features with higher importance induced a higher representational shift when masked. However, it's unclear how this representational shift can be interpreted: there's no guarantee that a large shift in the latent space would correspond to a large shift in a target function if one was learned on the latent space.
To answer this question, we examined whether the feature importance scores calculated for a classifier were similar to those calculated for the layers responsible for the latent representation. That is, if the full classifier can be described as the composition of a latent space encoder f_e and a projection head f_d, we computed for each test point x_j the correlation ρ_j of $\mathbf{b_i(f_e, x_j)}$ and $\mathbf{b_i(f_d \circ f_e, x_j)}$ across all features i. Finally, we averaged ρ_j to obtain a point estimate of the correlation between importance scores of f_e and $f_d \circ f_e$.
We took the classifier that was trained on the MNIST data, and followed the above procedure for three types of feature importance metrics (Gradient Shap [4], Salience [5] and Integrated Gradients [6]). We repeated this process across 5 random reseedings of the model. If the unsupervised and supervised feature importance are consistent with each other, we would expect a high correlation.

3.5 Computational requirements

In our experiments, we used a single NVIDIA A100 GPU and 8 cores of an AMD EPYC 7643 processor. All together, the experiments took approximately 80 hours to run.

4 Results

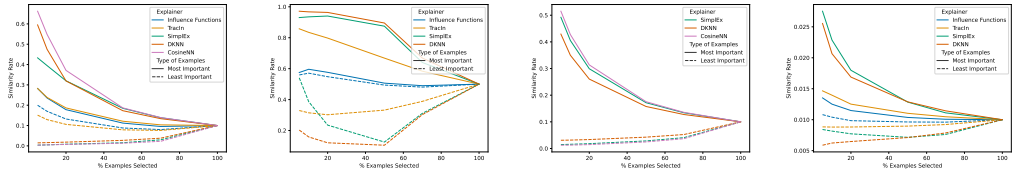
Claim 1.1 - Consistency of feature importance — Our results were closely aligned with those of the original work (Figure 1). We found that for all models, including those trained on the CIFAR-100 dataset, the representation shift induced by masking high-importance features was always greater than in the random baseline, although in the case of CIFAR-10 this was only true for the Integrated Gradients method. In general, integrated gradients and gradient Shap performed best, while saliency performed worst.



(a) MNIST (b) ECG5000 (c) CIFAR-10 (d) CIFAR-100

Figure 1. Consistency check for label-free feature importance. Each curve shows the size of representation shift that is induced as we mask an increasing percentage of the top-ranked most important features. Each line shows the average and 95% standard deviation over 5 random reseedings.

Claim 1.2 - Consistency of example importance — Our experiments replicated the findings of Crabbé and Schaar [1] (Figure 2). Similarity rates were always higher when calculated across examples with higher label-free example importance, including on the CIFAR-100 dataset. The one exception to this was the SimplEx method on ECG5000, where high-importance examples only achieved marginally higher similarity than the lowest importance scores. This is directionally in line with the results of Crabbé and Schaar [1], but it was not remarked on in their paper. Our newly defined 'cosine nearest neighbours' techniques outperformed all other methods.



(a) MNIST (b) ECG5000 (c) CIFAR-10 (d) CIFAR-100

Figure 2. Consistency check for label-free example importance. Each solid line shows how the similarity rate changes as we calculate it over some percentage of the highest-ranked training examples, for each choice of label-free example importance metric. The dotted lines show what happens if we start with the bottom-ranked samples instead.

(a) saliency maps

PEARSON	RECON.	DENOIS.	INPAINT.	CLASSIF.
RECON.				
DENOIS.	0.41 ± 0.02			
INPAINT.	0.34 ± 0.03	0.31 ± 0.02		
CLASSIF.	0.44 ± 0.02	0.39 ± 0.01	0.32 ± 0.02	

(b) example importance

PEARSON	RECON.	DENOIS.	INPAINT.	CLASSIF.
RECON.				
DENOIS.	0.1 ± 0.02			
INPAINT.	0.06 ± 0.02	0.1 ± 0.04		
CLASSIF.	0.06 ± 0.02	0.07 ± 0.01	0.06 ± 0.03	

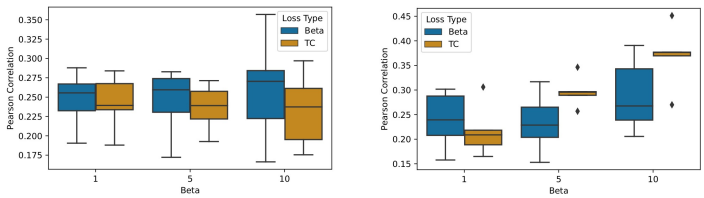
Table 1. Pearson correlation for saliency maps and example importance (avg +/- std) between different pretext tasks on MNIST.

Claim 2.1 - Correlation of feature importance across tasks — Using the MNIST dataset, we found nearly identical results to Crabbé and Schaar [1] when measuring the correlations of feature importance across different tasks. All model pairs showed a modest correlation and had a small standard error across runs (Table 1a). We found very similar results when we repeated the experiment on CIFAR-10 and CIFAR-100.

Claim 2.2 - Correlation of example importance across tasks — We also found nearly identical results to Crabbé and Schaar [1] when measuring correlations of example importance across different tasks using the MNIST data (Table 1b). All model pairs showed a low correlation in example importance (0.06 to 0.1) and a low standard deviation across runs.

However, we found substantially different trends when we repeated the experiment on CIFAR-10 and CIFAR-100, see Table 2 and 3. On these datasets, latent representations trained for separate unsupervised tasks had a high correlation (~ 0.9) to each other, but all had low correlation to the representations trained with the supervised task (~ 0.13).

Claim 3 - disentanglement of VAE — Our results supported their claim that increasing the strength of the disentanglement regularisation parameter (β) does not decrease correlation across feature maps in either $\beta-$ or TC-VAEs. Although our results did not match theirs perfectly, we were satisfied that they exhibited the same trends, and the same rough distribution of the Pearson statistic (Figure3). While our Lucid visualisations showed some interesting patters, they did not provide any insight into the relationship between increased β and the Pearson correlations (Figure 4).



(a) MNIST (b) dSprites

Figure 3. Pearson correlation between feature importance scores of pairs of latent units for different values of β.

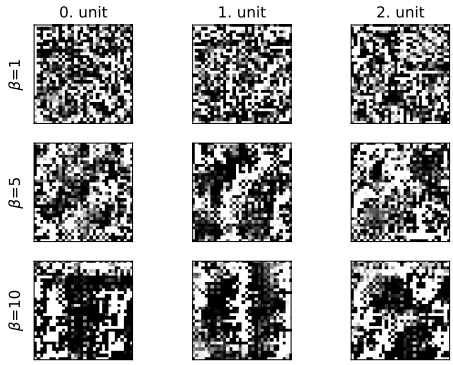


Figure 4. Lucid visualization of the 3 latent unit across the different β-VAE networks.

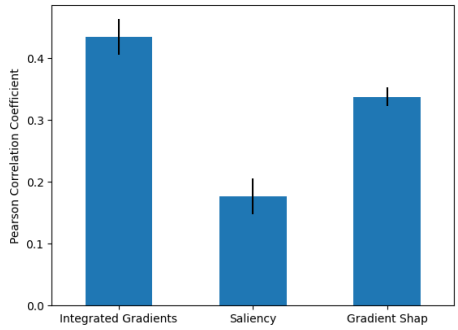


Figure 5. Pearson correlation between feature importance values of a full classifier and its constituent latent encoder. Each value is the mean across 5 runs, shown with a 95% confidence interval.

4.1 Results beyond original paper

(a) CIFAR-10

Pearson	Recon.	Denois.	Inpaint.	Classif.
Recon.				
Denois.	0.88 ± 0.07			
Inpaint.	0.92 ± 0.03	0.9 ± 0.03		
Classif.	0.16 ± 0.02	0.15 ± 0.01	0.15 ± 0.02	

(b) CIFAR-100

Pearson	Recon.	Denois.	Inpaint.	Classif.
Recon.				
Denois.	0.91 ± 0.07			
Inpaint.	0.89 ± 0.03	0.93 ± 0.02		
Classif.	0.2 ± 0.02	0.2 ± 0.02	0.19 ± 0.02	

Table 2. Pearson correlation of example importance (avg +/- std) across different pretext tasks.

(a) CIFAR-10

Pearson	Recon.	Denois.	Inpaint.	Classif.
Recon.				
Denois.	0.29 ± 0.11			
Inpaint.	0.3 ± 0.04	0.28 ± 0.03		
Classif.	0.19 ± 0.01	0.21 ± 0.02	0.18 ± 0.02	

(b) CIFAR-100

Pearson	Recon.	Denois.	Inpaint.	Classif.
Recon.				
Denois.	0.35 ± 0.08			
Inpaint.	0.31 ± 0.05	0.42 ± 0.07		
Classif.	0.21 ± 0.01	0.22 ± 0.01	0.2 ± 0.01	

Table 3. Pearson correlation of feature importance (mean $\pm$ standard deviation) across different pretext tasks for CIFAR10 and CIFAR100.

Comparison of Unsupervised and Supervised Feature Importance — We found that there was at best a moderate correlation between feature importance for the latent space encoder and the full model (see figure 5). The Pearson correlation coefficients were much higher for Integrated gradients (~0.45) and gradient SHAP (~0.35) than for Saliency (~0.2). There was very little variation in the strength of correlation between different runs.

5 Discussion

5.1 Reproducibility

Overall, our reproducibility study shows that all the claims of *Label-Free Explainability for Unsupervised Models* hold. Although we could not match all their results exactly, we were satisfied that we replicated the major trends for all claims and datsasets. We found

that their claims about the consistency of their methods (1.1 and 1.2) generalised well to the CIFAR-100 dataset. This indicates that the author's proposed 'label-free' methods can be useful on more complex datasets than those that they originally used.
However, we found substantially different results when testing the claim (2.2) about correlations between example importance scores between latent representations using CIFAR-10 and CIFAR-100. On these more complex datasets, there was a much clearer divide between latent representations trained with and without label information. We conjecture that supervision signals have a larger impact on latent representations on more complex data, although we leave it to future work to test this more rigorously.

5.2 Are unsupervised methods consistent with supervised ones?

We found that there was at most a moderate correlation between the feature importance values assigned to a model and its constituent latent space encoder. While their initial experiments showed that their methods were 'consistent', this was only defined based on properties of the latent space, and not in terms of the application of that latent representation to a downstream task. Hence, ours is an important finding, because it shows that the 'label-free' feature importance values do not necessarily align with a semantically meaningful definition of importance, even when the encoder and decoder were trained together. Therefore, we recommend that in applications where an accurate representation of feature importance is essential, the label-free methods should not be substituted for label-based ones. An obvious avenue for future work is to see whether this same problem afflicts label-free example importance methods.

5.3 What was easy

The authors created a publicly available repository which contained code for training models, running experiments, and generating the figures and tables. We found it straightforward to use this code to replicate their experiments. Overall, the authors did an excellent job of making their work reproducible.

Additionally, the paper was very precise in describing how their approach extended on the existing literature. We found there was no ambiguity in the mathematical details of their proposed method, and could have easily implemented it ourselves if that had been required.

5.4 What was difficult

Although their codebase was easy to use end-to-end, it was occasionally difficult to read and validate that the code was bug free. We believe that the readability could have been improved by consolidating the experiment code into classes that were shared between experiments for the different datasets.

5.5 Communication with original authors

We contacted the authors of the paper via email, to ask about the motivation behind their choice of kernel in DKNN (to which they responded they used it because it is a standard choice), and why in the computation of the saliency maps the features are multiplied with their latent values in the code (to which they responded that it was to preserve consistency with earlier methods, despite not being strictly necessary).

6 Acknowledgment

This work was partially supported by the Hungarian Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program.

References

1. J. Crabbé and M. van der Schaar. "Label-Free Explainability for Unsupervised Models." In: **Proceedings of the 39th International Conference on Machine Learning**. Ed. by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 4391–4420.
2. I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework." In: **International Conference on Learning Representations**. 2016.
3. M. T. Ribeiro, S. Singh, and C. Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." In: **Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining**. 2016, pp. 1135–1144.
4. S. M. Lundberg and S.-I. Lee. "A unified approach to interpreting model predictions." In: **Advances in neural information processing systems** 30 (2017).
5. K. Simonyan, A. Vedaldi, and A. Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps." In: **arXiv preprint arXiv:1312.6034** (2013).
6. M. Sundararajan, A. Taly, and Q. Yan. "Axiomatic attribution for deep networks." In: **International conference on machine learning**. PMLR. 2017, pp. 3319–3328.
7. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." In: **Proceedings of the IEEE** 86.11 (1998), pp. 2278–2324.
8. L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. **dSprites: Disentanglement testing Sprites dataset**. https://github.com/deepmind/dsprites-dataset/. 2017.
9. A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals." In: **circulation** 101.23 (2000), e215–e220.
10. A. Krizhevsky, G. Hinton, et al. "Learning multiple layers of features from tiny images." In: (2009).
11. J. Crabbe, Z. Qian, F. Imrie, and M. van der Schaar. "Explaining Latent Representations with a Corpus of Examples." In: **Advances in Neural Information Processing Systems**. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 12154–12166.
12. G. Pruthi, F. Liu, M. Sundararajan, and S. Kale. **Estimating Training Data Influence by Tracing Gradient Descent**. 2020. arXiv: 2002.08484 [cs.LG].
13. P. W. Koh and P. Liang. **Understanding Black-box Predictions via Influence Functions**. 2020. arXiv: 1703.04730 [stat.ML].
14. C. Olah, A. Mordvintsev, and L. Schubert. "Feature Visualization." In: **Distill** (2017). https://distill.pub/2017/feature-visualization.
15. K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition." In: **Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition**. CVPR '16. Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778.

A Top examples

Similarly to the original paper, we plot those training images that closest the highest example importance score with the test image. As it can be seen on Figure 6, VAE networks emphasize images that share spatial structure or colour, while the classifier only focuses on the semantic similarity.

B Necessary changes to the code

We had to make a number of small modifications to the code to replicate the results of the experiment. Each one is described below.

B.1 SimCLR

The most substantial differences between our study and the original were found when reproducing the consistency experiments for CIFAR10. We obtained a substantially different (though directionally similar graph) for each of these experiments.

We discovered that the original plot can be reproduced when using the same ResNet18[15] architecture with random weights. We hypothesize the authors accidentally missed the loading of pretrained weights, because there is a bug in the relevant part of the code which causes model loading to fail silently. The plot in our paper was obtained with this bug fixed. Nevertheless, it does not affect their claim.

Also, when describing this experiment, the authors noted in their paper that they sampled 1000 train examples $x^n \in D_{train}$ and matched latent representation of test images against these (Section 4.1 in the original paper). Accidentally, however, they sampled every data point from the test set. While this is technically a mistake, we fixed the issue before replicating the experiment, and found that it made no difference to the results.

B.2 Distance function

In claim 2.2, the authors measured the Pearson correlation between DKNN-based example importance scores. However, when determining these scores, they used inverted squared Euclidean distances as their kernel function $k(x, y) = 1/(x - y)^2$. We thought that using a kernel that was more linear in distance would reduce noise in correlation analyses, and so replaced this function with the negative of the distance $k(x, y) = -\sqrt{(x - y)^2}$. However, we found this change made no major difference in the results, and therefore we chose to keep the original distance function in our final code.

B.3 Data prefetching

Originally, the authors' code ran the data prefetching synchronously, with one worker. For this reason, we made changes in the authors' code in order to parallelize this process, which resulted in a significant speed-up.

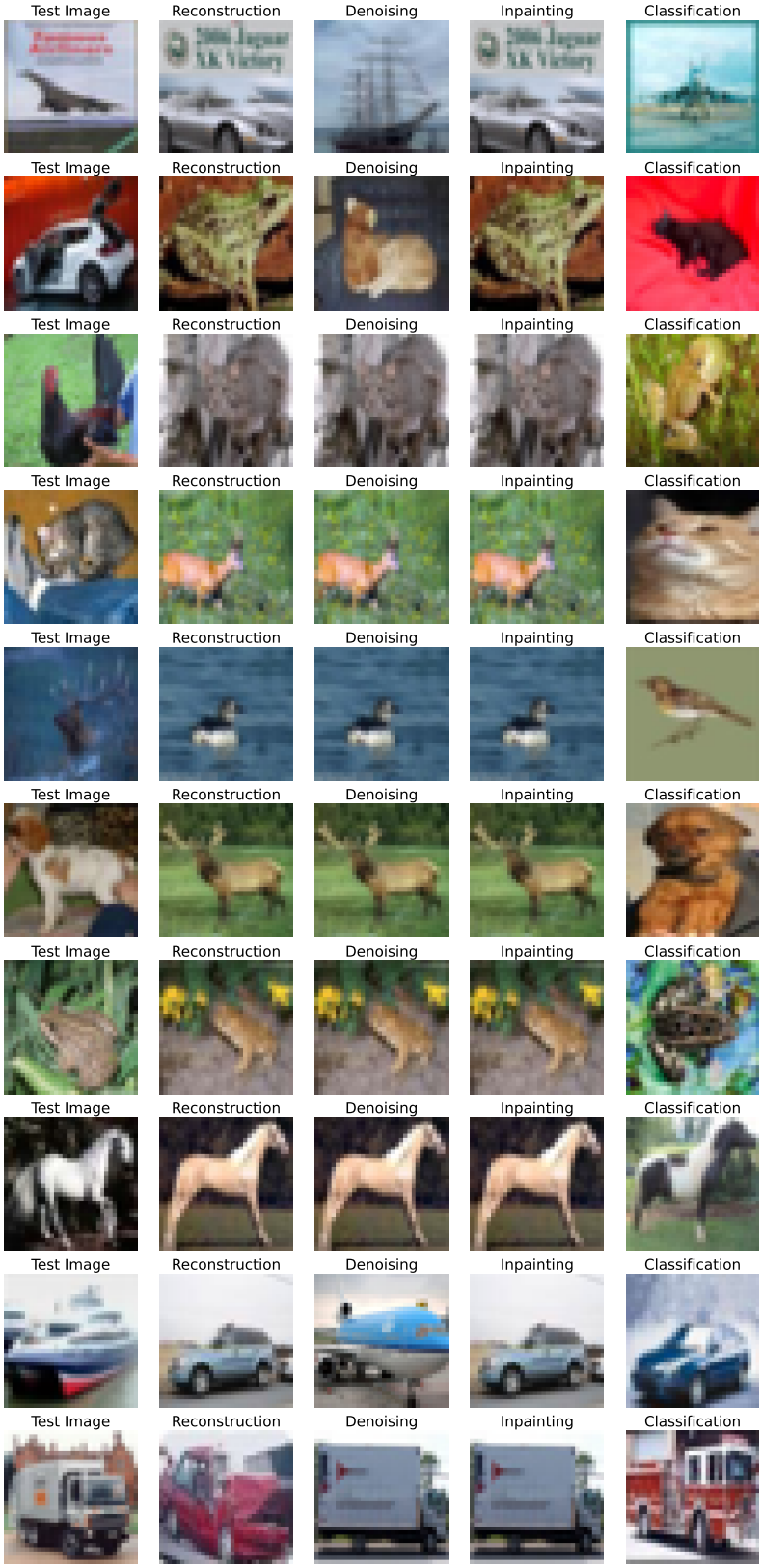


Figure 6. Label-free top examples for VAE networks and the classifier.

C AutoEncoder of CIFAR10 and CIFAR100

For the classification, the same encoder was used but with an extra linear classifier head attached to the end of the network.

Component	Layer Type	Hyperparameters
Encoder	Conv2d	Input Channels:3 ; Output Channels:32 ; Kernel Size:3 ; Stride:1 ; Padding:1
	LayerNorm	
	ReLU	
	MaxPool2D	Stride:2
	Conv2d	Input Channels:32 ; Output Channels:64 ; Kernel Size:3 ; Stride:1 ; Padding:1
	LayerNorm	
	ReLU	
	MaxPool2D	Stride:2
	Conv2d	Input Channels:64 ; Output Channels:128 ; Kernel Size:3 ; Stride:1 ; Padding:1
	LayerNorm	
	ReLU	
	MaxPool2D	Stride:2
	Conv2d	Input Channels:128 ; Output Channels:128 ; Kernel Size:3 ; Stride:1 ; Padding:1
	ReLU	
	Flatten	Output Channels : 2048 (128*4*4)
	Linear	Input Dimension: 2048 ; Output Dimension: 128
	ReLU	
	Linear	Input Dimension: 128 ; Output Dimension: 128
Decoder	Linear	Input Dimension: 128 ; Output Dimension: 128
	Linear	Input Dimension: 128 ; Output Dimension: 2048
	Unflatten	Dimension:1 ; Unflatten Size:(128, 4, 4)
	ConvTranspose2d	Input Channels:128 ; Output Channels:64 ; Kernel Size:3 ; Padding:1 ; Stride:2 ; Output Padding:1
	LayerNorm	
	ReLU	
	ConvTranspose2d	Input Channels:64 ; Output Channels:32 ; Kernel Size:3 ; Padding:1 ; Stride:2 ; Output Padding:1
	LayerNorm	
	ReLU	
	ConvTranspose2d	Input Channels:32 ; Output Channels:3 ; Kernel Size:3 ; Padding:1 ; Stride:2 ; Output Padding:1
	Sigmoid	

Table 4. CIFAR-10 & CIFAR-100 AutoEncoder Architecture.

D Hyperparameters

Table 5. Hyperparameters for each model used trained in conducted experiments.

MODEL	LEARNING RATE	β_1	β_2	ϵ	WEIGHT DECAY	MOMENTUM	EPOCHS	PATIENCE
MNIST AUTOENCODER	.001	.9	.999	10^-8	10^-5		100	10
ECG5000 AUTOENCODER	.001	.9	.999	10^-8	0		150	10
CIFAR-10 SIMCLR	.6				10^-6	.9	100	10
MNIST VAE	.001	.9	.999	10^-8	10^-5		100	10
DSPRITES VAE	.001	.9	.999	10^-8	10^-5		100	10