RESCIENCE C

Replication / ML Reproducibility Challenge 2022

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

Valentinos Pariza[1, ID], Avik Pal[1, ID], Madhura Pawar[1, ID], and Quim Serra Faber[1, ID]

[1]University of Amsterdam, Amsterdam, The Netherlands – [1]Equal contribution

Edited by
Koustuv Sinha,
Maurits Bleeker,
Samarth Bhargav

Received
04 February 2023

Published
20 July 2023

DOI
10.5281/zenodo.8173674

Reproducibility Summary

Scope of Reproducibility – In this work, we evaluate the reproducibility of the paper *Label-Free Explainability for Unsupervised Models* by Crabbe and van der Schaar [1]. Our goal is to reproduce the paper's four main claims in a label-free setting: (1) feature importance scores determine salient features of a model's input, (2) example importance scores determine salient training examples to explain a test example, (3) interpretability of saliency maps is hard for disentangled VAEs, (4) distinct pretext tasks don't have interchangeable representations.

Methodology – The authors of the paper provide an implementation in PyTorch for their proposed techniques and experiments. We reuse and extend their code for our additional experiments. Our reproducibility study comes at a total computational cost of 110 GPU hours, using an NVIDIA Titan RTX.

Results – We reproduced the original paper's work through our experiments. We find that the main claims of the paper largely hold. We assess the robustness and generalizability of some of the claims, through our additional experiments. In that case, we find that one claim is not generalizable and another is not reproducible for the graph dataset.

What was easy – The original paper is well-structured. The code implementation is well-organized and with clear instructions on how to get started. This was helpful to understand the paper's work and begin experimenting with their proposed methods.

What was difficult – We found it difficult to extrapolate some of the authors' proposed techniques to datasets other than those used by them. Also, we were not able to reproduce the results for one of the experiments. We couldn't find the exact reason for it by running explorative experiments due to time and resource constraints.

Communication with original authors – We reached out to the authors once about our queries regarding one experimental setup and to understand the assumptions and contexts of some sub-claims in the paper. We received a prompt response which satisfied most of our questions.

Correspondence should be addressed to Valentinos Pariza (valentinos.pariza@student.uva.nl)
The authors have declared that no competing interests exist.
Code is available at https://github.com/valentinosPariza/Re-Label-Free-XAI. – SWH swh:1:dir:8aa22d2a6b71c52b0863a06ab40f40ada1ec5355.
Open peer review is available at https://openreview.net/forum?id=qP34dvJpHd.

1 Introduction

Deep learning models are getting more and more advanced, making it difficult for humans to understand and retrace how an algorithm arrives at a specific result. To solve this problem, explanation methods were developed.
Post-Hoc methods separate explanations from models allowing explanation methods to be compatible with a variety of models [2]. They treat these models as "black boxes" due to their increasing complexity. Most of the post-hoc explanation techniques require labels to explain black-box outputs and thus they work only in a supervised setting.
The paper *Label-Free Explainability for Unsupervised Models*, by J. Crabbé and M. van der Schaar [1] 's goal is to explain black-box outputs in a label-free setting. The authors introduce two extensions for the **Feature Importance** and the **Example Importance** that highlight influential features and training examples respectively for a black box to construct representations at inference time.

The contribution of our work is summarized as follows:

1. We reproduce the main experiments by Crabbé and Schaar [1] to reproduce their main claims.
2. We conduct additional experiments to assess the robustness of label-free techniques proposed by the authors. Since they originally experiment on image and time-series datasets, we extend their techniques to find salient features and training samples for graphs and text datasets respectively.
3. We find that one of the authors' claims is model-specific when we introduce a penalty term to the loss function of those models.

The code implementation of our study is publicly available[1].

2 Scope of reproducibility

The original paper provides label-free extensions of feature and example importance methods. We identify the following main claims from the paper:

Claim 1 Label-free **feature importance** scores allow us to determine salient features of a model's input that contribute to an output prediction.

Claim 2 Label-free **example importance** scores allow us to determine salient training examples that explain a test example.

Claim 3 **Interpretability** of saliency maps derived from disentangled VAE latent units **is hard** and is **unrelated to the strength of disentanglement** between those units.

Claim 4 Distinct pretext tasks **don't have interchangeable representations**, and for example importance, pretext tasks with labels have more different representations than label-free pretext tasks.

Claim 1 and **Claim 2** are the hypotheses focusing directly on the proposed extensions. **Claim 3** focuses on the extent of application of **Claim 1**. **Claim 4** addresses practical utility of **Claim 1** and **Claim 2**. Our additional experiments challenge the robustness of **Claim 1** and **Claim 2** and the generalizability of **Claim 3**.

3 Methodology

We use the code provided by the authors, with some minor fixes (missing statements to load libraries, and inconsistent function calls) [3]. The code is well-structured and documented. We extend the authors' work by providing additional experiments and

[1] https://github.com/valentinosPariza/Re-Label-Free-XAI

results. We discuss important concepts pertaining to the original paper in Sections 3.1 and 3.2 before we discuss technical aspects.

3.1 Going from Label to Label-Free Setting

For a supervised setting, feature space $\mathcal{X}$ and label space $\mathcal{Y}$, make a black-box model $f : \mathcal{X} \to \mathcal{Y}$. For label-free setting, we train an autoencoder model with parameters $\theta \in \Theta$ using label-free loss $\mathcal{L} : \mathcal{X} \times \Theta \to \mathbb{R}$ on training set $\mathcal{D}_{train} = \{x^n | n \in \mathbb{N}^*\}$, where sample $x^n \in \mathbb{R}^p$. We then treat the encoder as black box $f : \mathcal{X} \to \mathcal{H}$, that connects $\mathcal{X}$ and latent space $\mathcal{H} \subset \mathbb{R}^{d_H}, d_H \in \mathbb{N}^*$ using encoder parameters θ_e.

Feature Importance — In a supervised setting, we calculate the weighted importance score $b_i(f, x)$ for every input feature $i \in d_x$ by summing the importance scores across every component of f (for classification, where we output j probabilities of j classes, we sum across j components of f). Similarly, for a label-free setting, we calculate $b_i(f, x)$ by summing importance scores of d_H latent components.
To calculate the importance scores, the authors use well-known attribution methods (AMs) - Saliency [4], Integrated Gradients (IG) [5] and Gradient Shap [6] which are implemented as part of the open source library, Captum [7].

Example Importance — A score c^n is assigned to every training example x^n based on it's importance in explaining a test example $x \in \mathcal{X}$. The original paper introduces two families of methods to compute them [1]:

1. **Loss-Based:** This method simulates the shift in loss $\delta^n_\theta \mathcal{L}$ for a test example x when a particular training example x^n is removed. The $\delta^n_\theta \mathcal{L}$ is also defined as the score c^n for x^n. In a supervised setting, we utilize the data label y^n to evaluate the loss for the entire model. But in a label-free setting, we only rely on the representations from the encoder and drop the decoder. Hence, we also decompose overall parameter gradients as relevant (encoder) and irrelevant (decoder). We only consider shifts in the loss by relevant parameters. This is done by differentiating the loss $\mathcal{L}$ with respect to θ_e.
2. **Representation-Based:** This method assigns a score by analyzing latent representations of training examples. In a supervised setting, we consider representations as the output of an intermediate layer $f_e(x)$. The similarity between the representation of test and training set examples can be quantified by reconstructing $f_e(x)$ with $f_e(\mathcal{D}_{train}) : f_e(x) \approx \sum_{n=1}^{N} w^n(x) \cdot f_e(x^n)$. The weight $w^n(x)$ is defined as the score c^n for x^n. For a label-free setting, we use encoder output, $f_e(x)$.

For loss-based, the authors use the Influence Function [8] and TracIn [9] AMs. And for representation-based, the authors introduce two AMs - DKNN and SimplEx [1].

3.2 Additional Experiments: Attribution Priors

Attribution prior encodes domain knowledge into a model [10]. The prior knowledge is defined by a function $\Omega : \mathbb{R}^{p \times n} \to \mathbb{R}$ that gets an attribution matrix Φ. In this work, we use pixel attribution prior which is a penalty function on pixel attributions, that promotes a high level of smoothness in the attributions by minimizing the total variation of neighbouring pixel attributions. It was introduced by Erion et al. [10] and is defined as:

$$\Omega_{pixel}(\Phi(\theta, \mathcal{X})) = \sum_{l} \sum_{i,j} |\phi^l_{i+1,j} - \phi^l_{i,j}| + |\phi^l_{i,j+1} - \phi^l_{i,j}| \tag{1}$$

Here, $\phi^l_{i,j}$ is the attribution of i,j-th pixel for l-th training sample. Encoding prior knowledge of the model is done by adding Ω to the model's loss function:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\theta; \mathcal{X}, y) + \lambda\Omega(\Phi(\theta, \mathcal{X})) \tag{2}$$

3.3 Model descriptions

We use the same models as used by the authors, except for the text and variational graph autoencoder. Table 1 gives an overview of the models used. We train these models and use the trained models during inference to find salient features and training samples.

Model	Input Dataset	# of Params	Evaluated Claims
CNN Denoising Autoencoder*	MNIST Tiny ImageNet	87.1k 416.3k	1, 2, 4 1, 2
LSTM Reconstruction Autoencoder SimCLR	ECG5000 CIFAR-10	249.4k 12481.7k	1, 2
β-VAE* & TC-VAE*	MNIST dSprites	463.7k 502k	3
Text Autoencoder* Variational Graph Autoencoder*	AGNews Cora	4048.1k 46.8k	2 1

Table 1. Overview of models used. * denotes models used for additional experiments. We use MNIST's autoencoder model for Tiny ImageNet with the input size set to 64 and $d_H = 16$.

3.4 Datasets

The original paper's experiments generate latent representations for the black-box models. Table 1 denotes which dataset was used to train which model. Table 2 provides an overview of these datasets. The authors use MNIST, ECG5000, CIFAR-10, and dSprites datasets in their experiments. Tiny ImageNet is a subset of the ImageNet dataset [11]. We use Tiny ImageNet to see whether **Claim 1** and **Claim 2** are satisfied on an image dataset other than the ones used by authors. Cora dataset consists of academic publications as nodes and citations between them as links. We use the Cora dataset to explain graphs in a label-free setting. For text explainability, we take a balanced subset of the AGNews dataset while normalizing and tokenizing the text, keeping a maximum token length of 64 [12].

Dataset	Samples		Classes	Description
	Training	Test		
MNIST*	60K	10K	10	28x28 grayscale images of the digits [13]
ECG5000	4K	1K	2	Time Series - 20 hour long ECG [14]
CIFAR-10*	50K	10K	10	32x32 colour images [15]
dSprites*†	73K	n/a	6	64x64 grayscale images of 2D shapes [16]
Tiny ImageNet*	100K	10K	200	64×64 colour images [17]
AGNews*	10K	1.7K	4	Collection of news articles [18]
	Nodes	Edges		
Cora*	2708	5429	7	Graph Based Citation Network. [19]

Table 2. Overview of datasets used. †- 2D shapes in the dataset were generated from six independent latent factors. ∗ - datasets used for additional experiments.

3.5 Hyperparameters

We use the same hyperparameters for all the experiments as the original paper [1]. The hyperparameters for graph and text explainability are mentioned in the Appendix Sections B.1.1 and C.1.2 respectively.

3.6 Experimental setup and code

We closely follow the setup of the original paper to reproduce their results [1]. We reproduce our claims quantitatively using the metrics described in Table 3. We mask important pixels to find salient features in images. Likewise, we mask edges and isolate important nodes to find salient nodes in graphs. The setup for graph and text explainability is detailed in Appendix Sections B.1 and C.1 respectively.
The authors' **Claim 3** addresses interpretability for disentangled VAE (d-VAE) models. The claim is based on experiments with two d-VAE models, β-VAE and TC-VAE. We extend this original setup to examine whether "interpretability is indeed hard" for these two d-VAE models. Ideally, we want to make each latent unit of our d-VAEs pay attention to distinct parts of the input. We attempt to do that by controlling their SM. Thus, we provide a secondary objective for the SM that helps tune the model's attention. This secondary objective is achieved through the use of attribution priors (Section 3.2). We use the penalty function of pixel attribution prior (Equation 1) for d-VAE (β and TC) models. We keep $d_H = 3$ for MNIST. We train both these d-VAE models with loss $\mathcal{L}$ as in Equation 2, $\beta \in \{1, 5, 10\}$ and regularization parameter, $\lambda \in \{0.001, 0.005, 0.01, 0.1\}$. We select the pixel attribution prior because it works very well [10]. It is simple and easy to compute.

Experiment	Metric(s)	Description
Feature Importance	Latent shift	Shift: $\|\|f(x) - f(m \odot x + (1-m) \odot \bar{x}\|\|_{\mathcal{H}}$ is calculated for each feature x_i while masking M important features with mask m.
Example Importance	Similarity Rate	The mean correct prediction $\sum_{n=1}^{M} \delta_{y,y^n}/M$ of labels of M most important training examples y^n against label of test example y.
Comparison b/w saliency maps	Pearson Correlation	Measures correlation between importance scores by taking covariance between them and dividing by the product of their standard deviations.

Table 3. Overview of metrics used.

3.7 Computational requirements

We carried out our experiments on a cluster of nodes, each had an NVIDIA Titan RTX GPU. Our reproducibility study required a total of 110 GPU hours, 90 for the original and 20 for the additional experiments (see Appendix A for more information).

4 Results

4.1 Results reproducing original paper

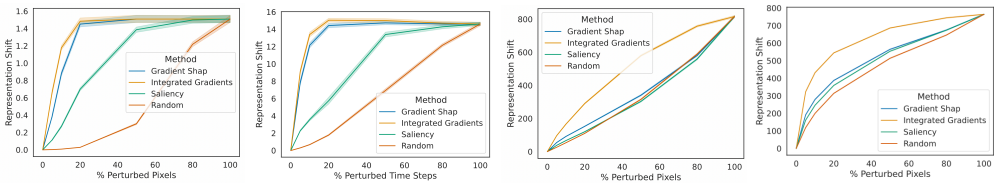
Claim 1: Label-Free Feature Importance — Claim 1 breaks into 4 sub-claims as follows:

1. Latent shift increases sharply for perturbing a few most important features.
2. This increase decreases when we disturb less relevant pixels.
3. Latent shift by feature-attribution methods is higher than that generated by perturbing random pixels.

4. Integrated Gradients (IG) outperforms other methods.

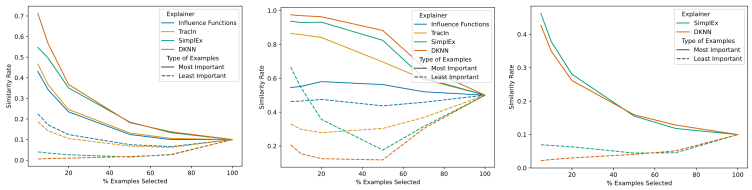
Figure 1 shows the trends of latent shifts by 3 AMs - IG, Saliency, and Gradient Shap for 3 black-box models. **The above 4 sub-claims are reproducible** verifying the reproducibility of **Claim 1** for **MNIST and ECG5000**. For CIFAR-10, sub-claim 3 is not reproducible. We see this discrepancy in Figures 1c and 1d. We discuss why is that so in Section 5.

Claim 2: Label-Free Example Importance — We get the similarity rates for the three datasets by varying the number of selected important training examples, as shown in Figure 2. The downward trend of similarity rates indicates that similar training examples are assigned higher importance scores. We also observe that representation-based example importance methods give much higher similarity measures for important training examples than loss-based methods. These results validate the consistency of the example importance methods and we reproduce **Claim 2**.



(a) MNIST **(b)** ECG5000 **(c)** CIFAR-10 (ours) **(d)** CIFAR-10 (theirs)

Figure 1. Consistency check for label-free feature importance.



(a) MNIST **(b)** ECG5000 **(c)** CIFAR-10

Figure 2. Consistency check for label-free example importance.

Claim 3: Latent Space Interpretability of Disentangled VAEs using saliency maps (SM) — We break **Claim 3** into two sub-claims:

1. Interpretation and association of the d-VAE model's latent units with clear generative factors is hard using their saliency maps (SM).
2. Interpretations with SM cannot be improved with the increase in disentanglement between the latent units.

We examine the SM of each model's encoder's latent units to verify these two sub-claims. **Qualitative Result**: We observe that a latent unit is sensitive to a given image while insensitive to a similar image (Figure 4a, Image 1 versus Image 2 for latent unit 1). The latent unit's focus changes completely between two similar images (Figure 4b, Image 1 versus Image 2 for latent unit 5). Several latent units focus on the same part of the image (Image 3 of Figures 4a and 4b). These observations agree with the authors' observations and support sub-claim 1. Figures 9 to 11 present the full list of SM used for comparisons. **Quantitative Result**: Figure 3a shows that when we increase the disentanglement factor β, it does not lead to a strong increase in decorrelation between the SM of different latent units. Consequently, this suggests that SM are not appropriate for distinguishing generative factors of latent units of d-VAEs, thus proving sub-claims 1 and 2.

Hence, our results agree with those of the authors; thus **Claim 3 is reproducible.**

Claim 4: Comparing the learned Pretext Tasks Representations – We break **Claim 4** into 3 sub-claims and also discuss the respective results of each of the sub-claims as follows:

1. **Distinct pretext tasks do not yield interchangeable explainability representations**. We see in Table 4a that the Pearson Correlation Coefficients (PCCs) for **feature importance** range between 0.32 and 0.44 (**moderate positive correlation**). PCCs range between 0.06 and 0.13 (**weak positive correlation**) for **example importance** (Table 4b). These are in accordance with the original paper and confirm the reproducibility of the first sub-claim.
2. For feature importance, **the correlation between label-free pretext tasks' SM and classification tasks' SM is similar to the correlation between SM of label-free pretext tasks**. We see in Table 4a that the PCCs between SM of classification and the other label-free pretext tasks are in the same range as the correlations between SM of only the label-free pretext tasks. Thus, we confirm the reproducibility of the second sub-claim.
3. For example importance, **the correlation between representations of label-free pretext tasks and the classification task is lower to the correlation between representations of only label-free pretext tasks**. As seen in Table 4b the autoencoder-classifier correlations are the lowest, whilst the autoencoder-autoencoder PCCs are higher, as in the original paper. Hence, the third sub-claim is also reproduced.

The analysis of qualitative results is present in Appendix Section E.1.

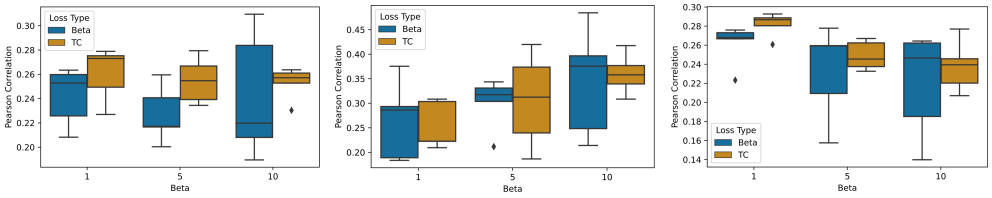
	R	D	I
D	0.41 ± 0.02		
I	0.33 ± 0.03	0.32 ± 0.01	
C	0.44 ± 0.02	0.41 ± 0.02	0.33 ± 0.02

(a) Saliency maps (avg $\pm$ std).

	R	D	I
D	0.09 ± 0.02		
I	0.13 ± 0.03	0.1 ± 0.03	
C	0.09 ± 0.03	0.06 ± 0.03	0.09 ± 0.02

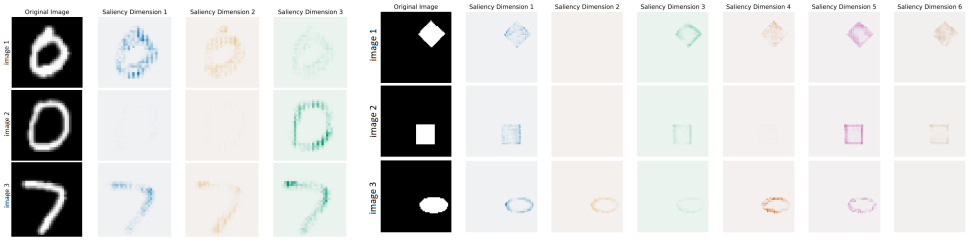
(b) Example Importance (avg $\pm$ std).

Table 4. Pearson Correlation. R-Reconstruction, D-Denoising, I-Inpainting, C-Classification



(a) MNIST **(b)** dSprites **(c)** MNIST ($\lambda = 0.005$)

Figure 3. PCC over pairs of saliency maps (Gradient Shap) from each combination of a model's latent unit for different values of β. Figure 3c uses attribution priors as explained in Section 3.6.



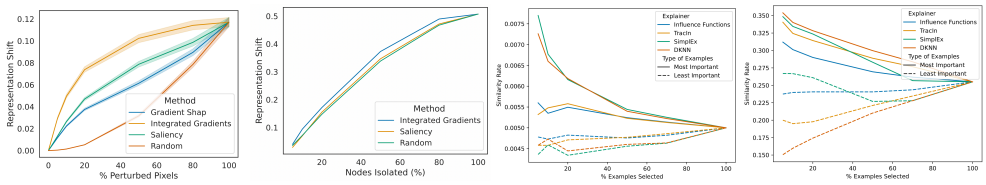
(a) MNIST (β-VAE with β=10) **(b)** dSprites (TC-VAE with β=1)

Figure 4. Saliency maps for each latent unit of the disentangled VAEs. The VAES selected were the ones with the lowest PCC. i^{th} saliency dimension highlights latent unit i.

4.2 Results beyond original paper

Label Free Feature Importance — We verify whether the 4 sub-claims from Section 4.1.1 hold for Tiny ImageNet and Cora datasets. **Tiny Imagenet satisfies Claim 1** as seen in Figure 5a. **For Cora, sub-claims 3 and 4 are not reproducible** as seen in Figure 5b. Graphs have complicated spatial features (nodes, edges, sub-graphs) which aren't well explained by AMs. This is because AMs rely heavily on computing gradients which causes **gradient saturation** on discrete values of the adjacency matrix of graph [20]. From these results, we conclude that the approach of generating latent shifts to compute feature importance works well for images and time-series data. AMs fail to explain graphs well.

Label Free Example Importance — The results obtained for the Tiny Imagenet dataset shown in Figure 5c are consistent and similar to the results obtained in the case of MNIST in Figure 2a and satisfy **Claim 2**. We observe a declining trend for similarity rate in the case of AGNews' text dataset for Figure 5d implying correct assignment of importance scores for most similar training examples. The trends verify that representation-based methods work better than loss-based methods but in the case of a higher percentage of training examples selected, the SimplEx method (representation-based) declines rapidly in performance compared to loss-based methods.



(a) Tiny ImageNet **(b)** Cora **(c)** Tiny ImageNet **(d)** AGNews

Figure 5. Figures 5a and 5b are for Feature Importance and Figures 5c and 5d are for Example Importance on additional datasets.

Challenging the Generalizability of Claim 3 with Attribution Priors — As discussed in Section 3.6, we aim to improve the interpretability of d-VAE models (β and TC) using pixel attribution prior. In Figure 3c, we see the PCCs between SM of different latent units for the two d-VAE models with attribution priors. In this case, with the increase in β, there is a clear decrease in the correlation between the latent units' SM. This denotes that latent units pay attention to different parts of the input image as we increase β. We see the same observations for different values of λ in Figure 12. In conclusion, d-VAEs with attribution prior identify better the role of each latent unit with their SM (**the interpretability is not hard anymore**). Furthermore, the interpretability of SM is related to the strength of the disentanglement between the units. Thus, given our experiments, **Claim 3 is not reproducible for these two d-VAE models when we use attribution priors**. We present a qualitative analysis with attribution priors in Appendix Section D.4.

5 Discussion

We reproduce the same results for the experiments as the original paper except for the result of CIFAR-10's experiment (Section 4.1.1). We communicated with the authors about this discrepancy. They hinted that it may be due to the difference in model weights for SimCLR. However, due to time and resource constraints, we could not investigate further by running more explorative experiments. We conclude that **Claim 1** is partially reproducible while others are completely reproducible.

We now discuss the reproducibility of claims for our additional experiments. **Claim 1** is not reproducible for graphs, discussed in Section 4.2.1. **Claim 2** is reproducible for Tiny ImageNet and text-based AGNews dataset, discussed in Section 4.2.2. We find that when we use attribution priors, **Claim 3** is not generalizable to all d-VAE models since it is not reproducible for the two d-VAE models (β and TC), as discussed in Section 4.2.3.

Through our additional experiments, we realize that there is scope for future work in the following areas. The first one is the extension of feature-importance methods to explain spatial attributes of graphs (edges, subgraphs). The second one is improving the interpretability of saliency maps for d-VAE models other than β-VAE and TC-VAE, by the use of attribution prior. Lastly, we note that running the Influence function (Section 3.1.2) to obtain an estimation of a shift in loss is time-consuming when we have large amounts of training samples (18 hours for $N = 1000$ time-series training samples). We also observe that the DKNN method gives a better (if not the best) measure of importance scores while being computationally fast and we suggest using the same method for further downstream tasks in future works. Overall, the experiments in the original paper are reproduced, and their main claims seem reasonably substantiated but could benefit from additional evidence in future research.

5.1 What was easy

The authors have significantly helped us to understand the problem, the suggested solution for the problem, and their claims, by providing a well-structured paper that includes detailed appendices consisting of mathematical proofs, implementation details, and experiments. It was easy to begin experimenting with their proposed methods, since the implementations provided are well-organized and fairly documented, with clear instructions on how to get started.

5.2 What was difficult

We found it difficult to extrapolate feature AMs to additional datasets. For example, the majority of the methodologies for explaining graphs are (i) for supervised setting [20] and (ii) the unsupervised setting of them is currently in development [21]. While computing feature importance scores for Text Explainability, we encountered an issue where the use of the embedding layer in our Text Autoencoder (see Appendix C.1) was incompatible with available AMs within Captum [7]. Also, we couldn't identify the right configurations for the CIFAR-10 experiment to reproduce similar trends as the original paper for feature importance.

5.3 Communication with original authors

We reached out to the authors once about our queries regarding the discrepancy in results for feature importance for the CIFAR-10 experiment and to understand the assumptions and contexts of some sub-claims in the paper. We received a prompt response which satisfied most of our questions.

References

1. J. Crabbé and M. van der Schaar. **Label-Free Explainability for Unsupervised Models**. 2022. DOI: 10.48550/ARXIV.2203.01928. URL: https://arxiv.org/abs/2203.01928.
2. S. R. Islam, W. Eberle, S. K. Ghafoor, and M. Ahmed. **Explainable Artificial Intelligence Approaches: A Survey**. 2021. DOI: 10.48550/ARXIV.2101.09429. URL: https://arxiv.org/abs/2101.09429.

3. **JonathanCrabbe/Label-Free-XAI: This repository contains the implementation of Label-Free XAI, a new framework to adapt explanation methods to unsupervised models. For more details, please read our ICML 2022 paper: 'Label-Free Explainability for Unsupervised Models'.** https://github.com/JonathanCrabbe/Label-Free-XAI.
4. K. Simonyan, A. Vedaldi, and A. Zisserman. **Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps**. 2013. DOI: 10.48550/ARXIV.1312.6034. URL: https://arxiv.org/abs/1312.6034.
5. M. Sundararajan, A. Taly, and Q. Yan. **Axiomatic Attribution for Deep Networks**. 2017. DOI: 10.48550/ARXIV.1703.01365. URL: https://arxiv.org/abs/1703.01365.
6. S. Lundberg and S.-I. Lee. **A Unified Approach to Interpreting Model Predictions**. 2017. DOI: 10.48550/ARXIV.1705.07874. URL: https://arxiv.org/abs/1705.07874.
7. N. Kokhlikyan et al. **Captum: A unified and generic model interpretability library for PyTorch**. 2020. arXiv:2009.07896 [cs.LG].
8. P. W. Koh and P. Liang. **Understanding Black-box Predictions via Influence Functions**. 2017. DOI: 10.48550/ARXIV.1703.04730. URL: https://arxiv.org/abs/1703.04730.
9. G. Pruthi, F. Liu, M. Sundararajan, and S. Kale. **Estimating Training Data Influence by Tracing Gradient Descent**. 2020. DOI: 10.48550/ARXIV.2002.08484. URL: https://arxiv.org/abs/2002.08484.
10. G. G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, and S. Lee. "Learning Explainable Models Using Attribution Priors." In: **CoRR** abs/1906.10670 (2019). arXiv:1906.10670. URL: http://arxiv.org/abs/1906.10670.
11. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A large-scale hierarchical image database." In: **2009 IEEE Conference on Computer Vision and Pattern Recognition** (2009), pp. 248–255.
12. X. Zhang, J. Zhao, and Y. LeCun. **Character-level Convolutional Networks for Text Classification**. 2015. DOI: 10.48550/ARXIV.1509.01626. URL: https://arxiv.org/abs/1509.01626.
13. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." In: **Proc. IEEE** 86 (1998), pp. 2278–2324.
14. A. L. Goldberger, L. A. N. 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).
15. A. Krizhevsky. "Learning Multiple Layers of Features from Tiny Images." In: 2009.
16. L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. **dSprites: Disentanglement testing Sprites dataset**. https://github.com/deepmind/dsprites-dataset/. 2017.
17. Y. Le and X. S. Yang. "Tiny ImageNet Visual Recognition Challenge." In: 2015.
18. X. Zhang, J. Zhao, and Y. LeCun. **Character-level Convolutional Networks for Text Classification**. 2016. arXiv:1509.01626 [cs.LG].
19. A. McCallum, K. Nigam, J. D. M. Rennie, and K. Seymore. "Automating the Construction of Internet Portals with Machine Learning." In: **Information Retrieval** 3 (2000), pp. 127–163.
20. R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. **GNNExplainer: Generating Explanations for Graph Neural Networks**. 2019. DOI: 10.48550/ARXIV.1903.03894. URL: https://arxiv.org/abs/1903.03894.
21. **torch_geometric.explain — pytorch_geometric documentation**. https://pytorch-geometric.readthedocs.io/en/latest/modules/explain.html.
22. T. N. Kipf and M. Welling. **Variational Graph Auto-Encoders**. 2016. DOI: 10.48550/ARXIV.1611.07308. URL: https://arxiv.org/abs/1611.07308.
23. R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. "GNNExplainer: Generating Explanations for Graph Neural Networks." In: **Advances in neural information processing systems** 32 (2019).
24. **Tokenizers**. https://github.com/huggingface/tokenizers.
25. Robertson. **NLP from scratch: translation with a sequence to sequence network and attention**. https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html.

A Computational Costs of Experiments

Experiment Type	Experiment Name	Target Claim	Section	GPU Hours
Reproducibility Study	rs-exp 1	Claim 1	4.1.1	9
	rs-exp2 1	Claim 2	4.1.2	3
	rs-exp2 2	Claim 2	4.1.2	22
	rs-exp2 3	Claim 2	4.1.2	2
	rs-exp3 1 (GS)	Claim 3	4.1.3	5
	rs-exp3 1 (IG)	Claim 3	4.1.3	8
	rs-exp3 2 (GS)	Claim 3	4.1.3	10
	rs-exp3 2 (IG)	Claim 3	4.1.3	28
	rs-exp 4	Claim 4	4.1.4	3
Additional Experiments	a-exp 1	Claim 1	4.2.1	4
	a-exp 2	Claim 2	4.2.2	8
	a-exp 3	Claim 3	4.2.3	8

Table 5. Computational Cost for each experiment, measured in GPU hours.

The conducted **Reproducibility Study Experiments (rs-exp)** of this work are:

rs-exp 1 Consistency Check of label-Free Feature Importance.

rs-exp 2 Consistency Check of label-free Example Importance.
- **rs-exp2 1** MNIST.
- **rs-exp2 2** ECG5000.
- **rs-exp2 3** CIFAR10.

rs-exp 3 Challenging authors' assumptions with disentangled VAEs using Integrated Gradients (IG) & Gradient Shap (GS).
- **rs-exp3 1** MNIST.
- **rs-exp3 2** dSprites.

rs-exp 4 Comparing the Explainability Representations from Different Pretext Tasks.

The conducted **Additional Experiments (a-exp)** of this work are:

a-exp 1 Label Free Feature Importance

a-exp 2 Label Free Example Importance

a-exp 3 Challenging the Generalizability of **Claim 3** with Attribution Priors on MNIST

B Label Free Feature Importance

B.1 Graph Explainability

Model — Since we want to explain the graph in a label-free unsupervised setting, we choose to use a Variational Graph Autoencoder (VGAE) [22]. Table 6 describes the architecture of the model. We perform an edge-link prediction task on the Cora citation dataset. We train the VGAE to minimize the objective:

$$\mathcal{L} = \mathbb{E}_{q(Z|X,A)}[\log p(A|Z)] - KL[q(Z|X,A)||p(Z)]$$

where A is the adjacency matrix, $q(Z|X,A)$ is the distribution parameterized by the inference model of 2-layer GCN. $p(Z)$ is the Gaussian prior associated with $\mathcal{N}(0,I)$ and KL is the KL-divergence. We train for 200 epochs with patience 10 by using Pytorch's Adam optimizer with a learning rate = .01.

Feature Importance — As a baseline, we use an unconnected graph, $A = 0$. In the case of images for the MNIST experiment, we mask important pixels by blackening them. In the case of a graph, we mask important nodes, by isolating them. Algorithm 1 summarises steps to compute the feature importance of nodes.

Component	Layer Type	Hyperparameters	Act. Func
Encoder	GraphConv	I/P Feats:1433 ; O/P Feats:32 I/P Feats:32 ; O/P Feats:16	ReLU ReLU
Reparameterisation Trick	The output of encoder contains μ and logσ. The latent representation is then generated via $h = \mu(x) + \sigma(x) \odot \epsilon, \epsilon \sim \mathcal{N}(0, 1)$		
Decoder		Inner Product Decoder	Sigmoid

Table 6. Variational Graph Autoencoder Architecture. The last column denotes the activation function used. I/P Feats means Input Features. O/P Feats means output features.

Algorithm 1: Label Free Feature Importance for Graph

Data: Undirected Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and its adjacency matrix A with $N = |\mathcal{V}|$ nodes.
Black-box $f : \mathcal{X} \to \mathcal{H}$, Feature importance method $a_i(\cdot, \cdot) : \mathcal{A}(\mathcal{H}^{\mathcal{G}}) \times \mathcal{G} \to \mathbb{R}^N$.
Result: Label-Free Feature Importance for nodes $b_i(f, \mathcal{G})$.

1 Convert black-box model f to a model that is usable for captum attribution method [23] ;
2 Compute & normalize node attributions of size $|\mathcal{V}|$;
3 Apply the Mask ;
4 Calculate latent shift ;
5 Repeat steps 1-3 for different rates of perturbation to induce while masking ;

Result Discussion — The work [20] corroborates our finding that the feature attribution methods are not the right indicator for explaining graphs. The issue of gradient saturation worsens for discrete inputs such as graph adjacency matrices. Future work in this direction can address two issues: (1) Currently, the focus is on explaining graphs for tasks such as node classification. We need a more robust, generalized solution that explains a particular latent representation of the graph, and which graph's attributes (nodes, edges, or subgraphs) are influential. (2) Development of a framework for the previous issue. Appendix section A of [20] also addresses the issues we faced while computing important nodes for a particular latent representation. They discuss a 'workaround' to use single-instance explanations to explain a multi-instance explanation (latent representation in our case). We implemented that in our code base.

B.2 Changes in masking and baseline inputs

We redefine the masking and baseline inputs for the experiments on MNIST using the following approaches:

Change in masking — In the original experiment, the variable m is set to 0 or 1 depending on whether the feature x_i to mask is important (x_i is important when its score is in the top M values as calculated by attribution methods). In the new case, considering a feature pixel x_i and a feature map x, we do the masking as follows:

$m \leftarrow 0$
if x_i is an important pixel **then**
  $m_i \leftarrow (1 - x_i)/\max(x)$
else
  $m_i \leftarrow 1$
end if

Instead of completely zeroing out an important pixel, we set the mask to a value inversely proportional to its importance.

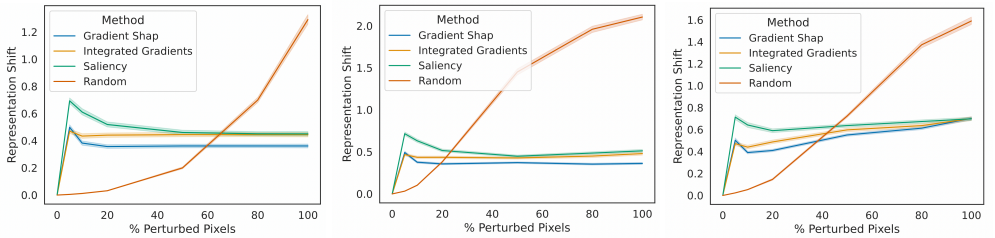
Change in baseline — When we assign blame to a certain cause, we consider the absence of the cause as a baseline for comparing outcomes. In the original experiment, a black image is used as the baseline $\bar{x}$. Instead of setting all pixels of $\bar{x}$ to 0, we set the pixel $\bar{x_i}$ as follows (consider x to be the input image with x_i being its feature pixel):

- We do masking as Appendix Section B.2.1 and set the baseline image pixels to:

$$\bar{x_i} = (1 - x_i)/\max(x) \tag{3}$$

- To the above extension, we also add noise to the baseline image:

$$\bar{x_i} = \mathcal{N}(0, 1) * (1 - x_i)/\max(x) \tag{4}$$



(a) Change in mask variable **(b)** Change in baseline (Equation 3) **(c)** Change in baseline (Equation 4)

Figure 6. Consistency check for label-free feature importance (Changes in baseline and masking)

Result Discussion — None of the trends (Section 4.1.1) in Figure 6 are satisfied. The shift remains constant for Figures 6a and 6b after 10% of perturbations, which suggests that our changes lead to gradient saturation. This leads us to understand why the mask and baselines are assigned values that are independent of the input.

C Label Free Example Importance

C.1 Text Explainability

Tokenizer — We train a **BertWordPieceTokenizer** model from Huggingface's Tokenizers [24] library using our training texts. We keep the vocabulary size at 10000 while also keeping a minimum token frequency of 2. We then use this trained tokenizer model to tokenize our train and test dataset.

Model — We adapt the architecture for text autoencoder from [25]. Figure 7 shows an illustration of the final architecture. The layers and their corresponding hyperparameters we use are summarized in Table 7. For our experiments, we keep the latent dimensions (also referred to as context vector) at 128. We use the negative log-likelihood loss (NLL-Loss) to train on a number of classes equal to vocabulary size. We also make use of the "Teacher Forcing" concept of using the actual target as inputs to the decoder instead of the last prediction of the decoder, 50% of the time. We train the model for 8 epochs with a learning rate of 0.01 and a Stochastic Gradient Descent optimizer.

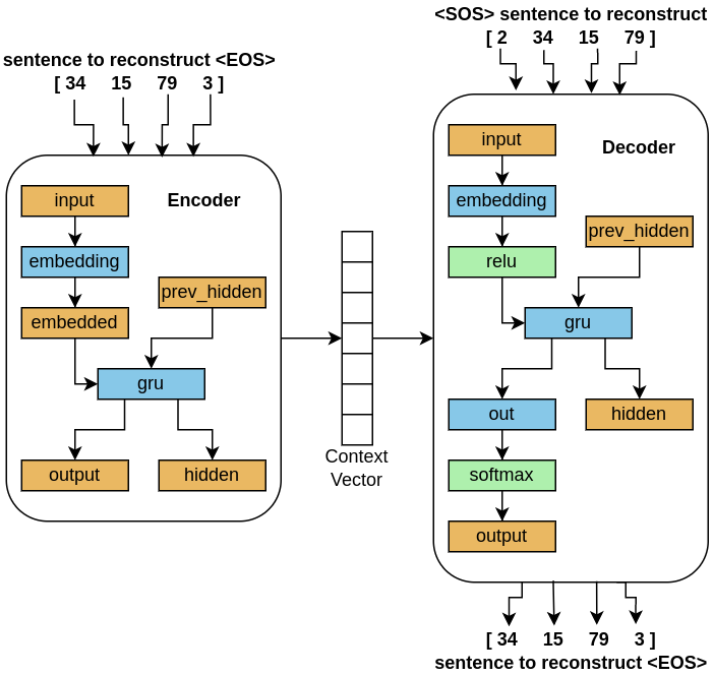


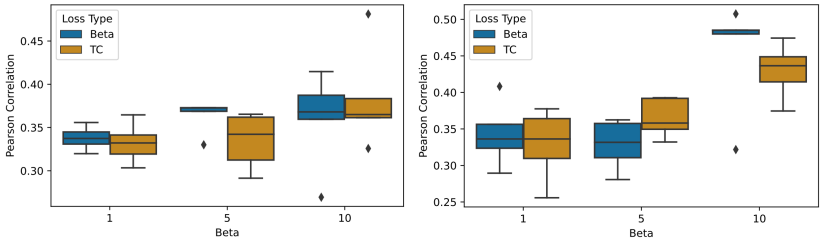
Figure 7. Text autoencoder architecture used for text explainability.

Component	Layer Type	Hyperparameters	Act. Func
Encoder	Embedding	I/P Feats:10000 ; O/P Feats:128	
	GRU	I/P Feats:128 ; O/P Feats:128	
Decoder	Embedding	I/P Feats:10000 ; O/P Feats:128	ReLU
	GRU	I/P Feats:128 ; O/P Feats:128	
	Linear	I/P Feats:128 ; O/P Feats:10000	LogSoftmax

Table 7. Layers and hyperparameters for text autoencoder model. The last column denotes the activation function used. I/P Feats means Input Features. O/P Feats means output features.

D Interpreting the latent units of VAEs with saliency maps

D.1 Additional Quantitative Results for Result on Claim 3 - Section 4.1.3



(a) MNIST (b) dSprites

Figure 8. Pearson Correlations over pairs of SM of distinct latent units computed with Integrated Gradients for different values of β.

D.2 Additional Qualitative Results for Result on Claim 3 - Section 4.1.3

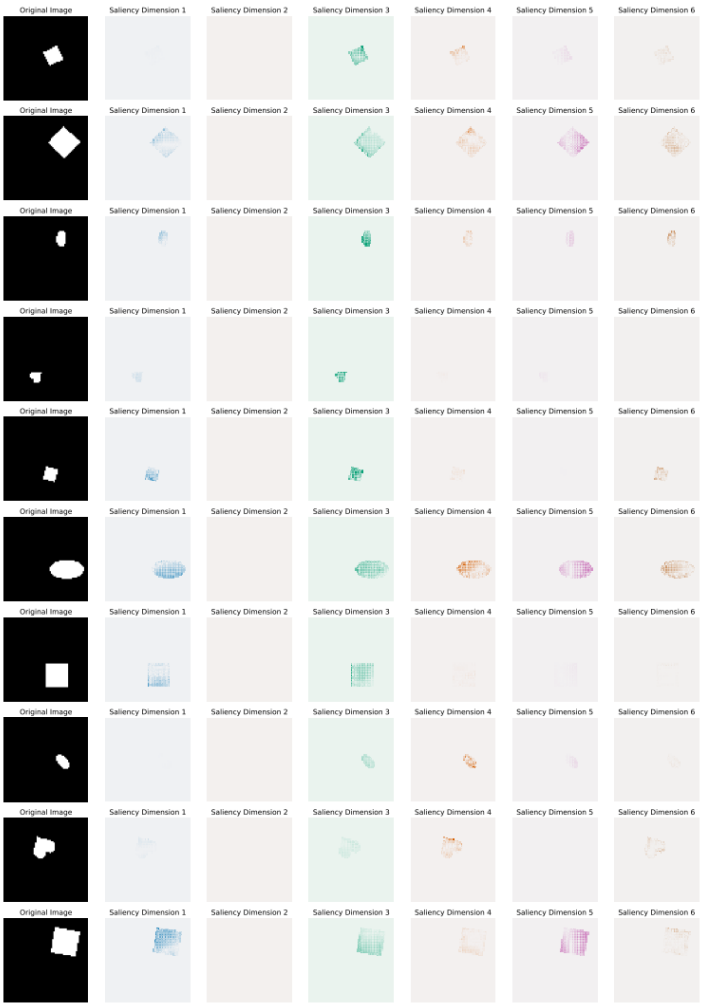


Figure 9. Saliency maps computed with Gradient Shap from the TC-VAE with β=1 on dSprites.

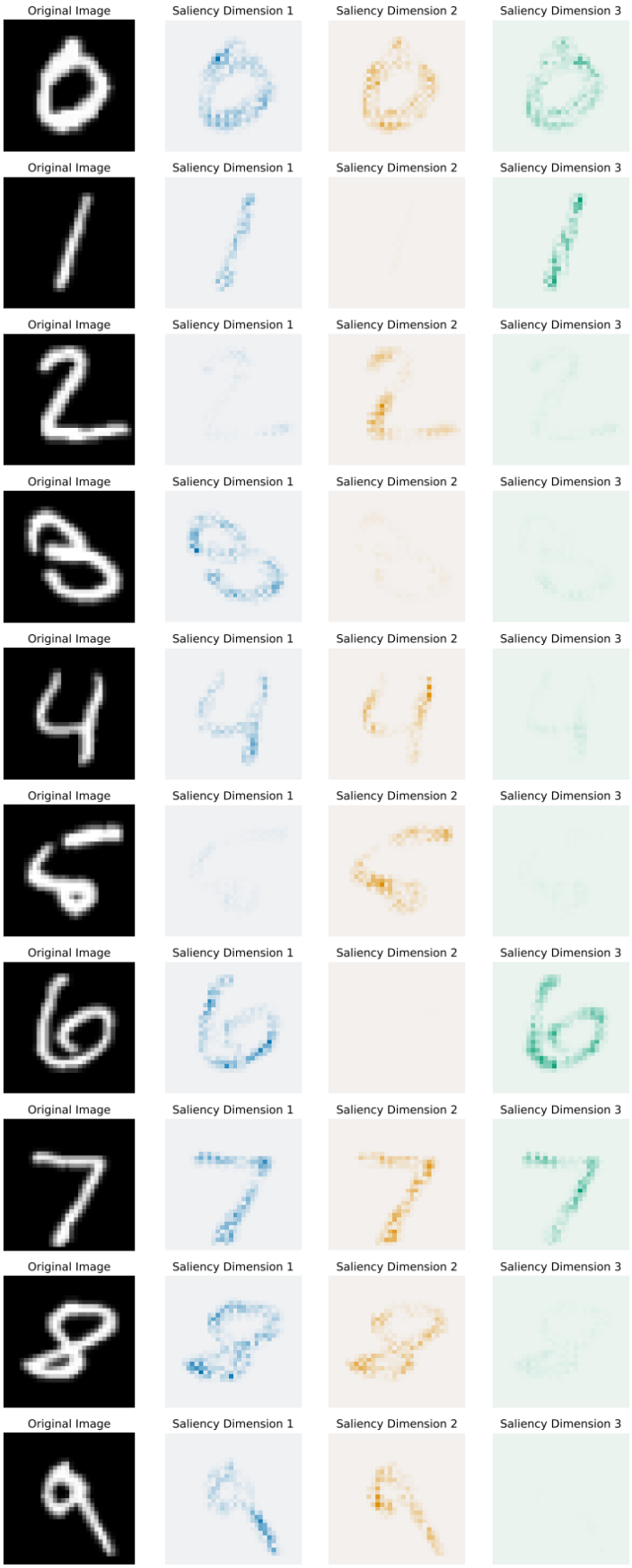


Figure 10. Saliency maps computed with Gradient Shap for β-VAE with β=10 on MNIST (Part 1).

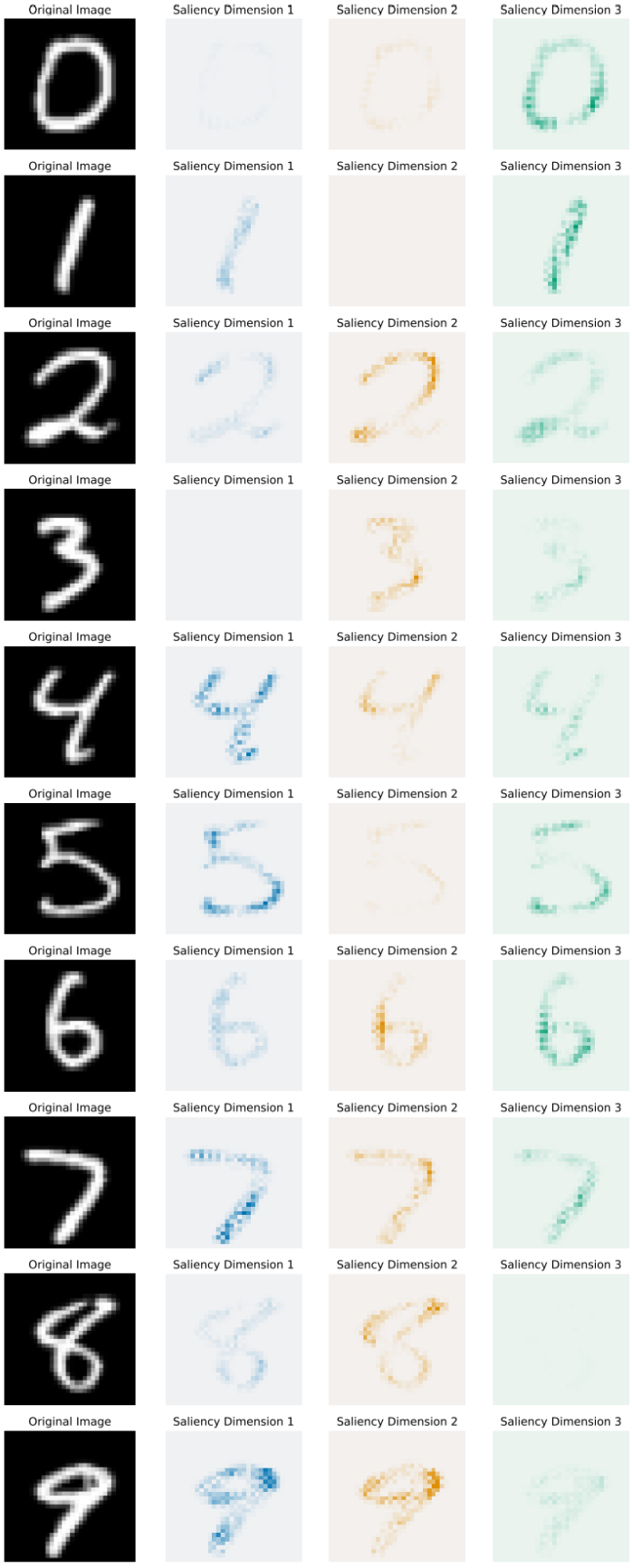
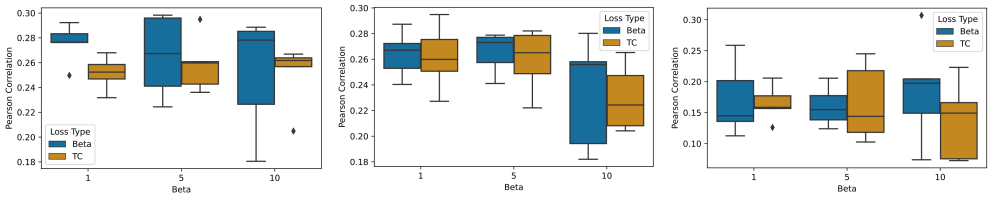


Figure 11. Saliency maps computed with Gradient Shap for β-VAE with β=10 on MNIST (Part 2).

D.3 Additional Quantitative Results for Result Section 4.2.3



(a) MNIST ($\lambda = 0.1$) **(b)** MNIST ($\lambda = 0.01$) **(c)** MNIST ($\lambda = 0.001$)

Figure 12. Pearson Correlations over pairs of SM of distinct latent units with Gradient Shap for different values of β. The models use attribution priors with a regularization parameter, λ.

D.4 Additional Qualitative Results for Result Section 4.2.3

In Figures 13 & 14, we see the saliency maps (SM) of the latent units of TC-VAE with β=10 and λ=0.001 on the MNIST Dataset. We see figures for λ=0.005, in Figures 15 & 16. Comparing Figures 10 & 11, we see that each latent unit of the TC-VAE with attribution prior tends to focus on distinct parts of the input image, with less intersection than the latent units of the β-VAE with no attribution prior. We observe the following cases:

1. For the TC-VAE (β=10 & λ=0.005) in Figure 15, for the digit *6*, latent unit 2 focuses more on the bottom-right part, whereas the latent unit 3 focuses on the top and left part. On the other hand, for the β-VAE (β=10 & no prior) in Figure 10, the latent units 1 and 3 focus on somewhat different parts of the digit *6*, but there is a significant intersection in the regions they focus on.
2. For the TC-VAE (β=10 & λ=0.005) in Figure 15, for the digit *8*, latent unit 1 focuses more on the center and upper part of the digit, whereas the latent unit 3 focuses on the bottom part. On the other hand, for the β-VAE (β=10 & no prior) in Figure 10, the latent units 1 and 2 focus on almost the same regions.

The observations above seem to be even stronger for the TC-VAE with β=10 and smaller λ than before, equal to 0.001. Similar observations for SM of the latter model compared to the SM of the β-VAE with β=10 and no prior can be made:

1. For the TC-VAE (β=10 & λ=0.001) in Figure 13, for the digit *4*, latent unit 1 focuses more on towards the corners and edge-points of the digit whereas the latent unit 3 focuses on the vertical main line of the digit *4*.
2. For the β-VAE (β=10 & no prior) in Figure 10, the latent units 1 and 2 focus on somewhat different parts, but there is a significant intersection in the regions they focus on (e.g., the top-right edge-point).

In conclusion, the prior attribution seems to have the ability to train a model such that its latent units focus on distinct parts with no significant overlapping of the regions they focus on.

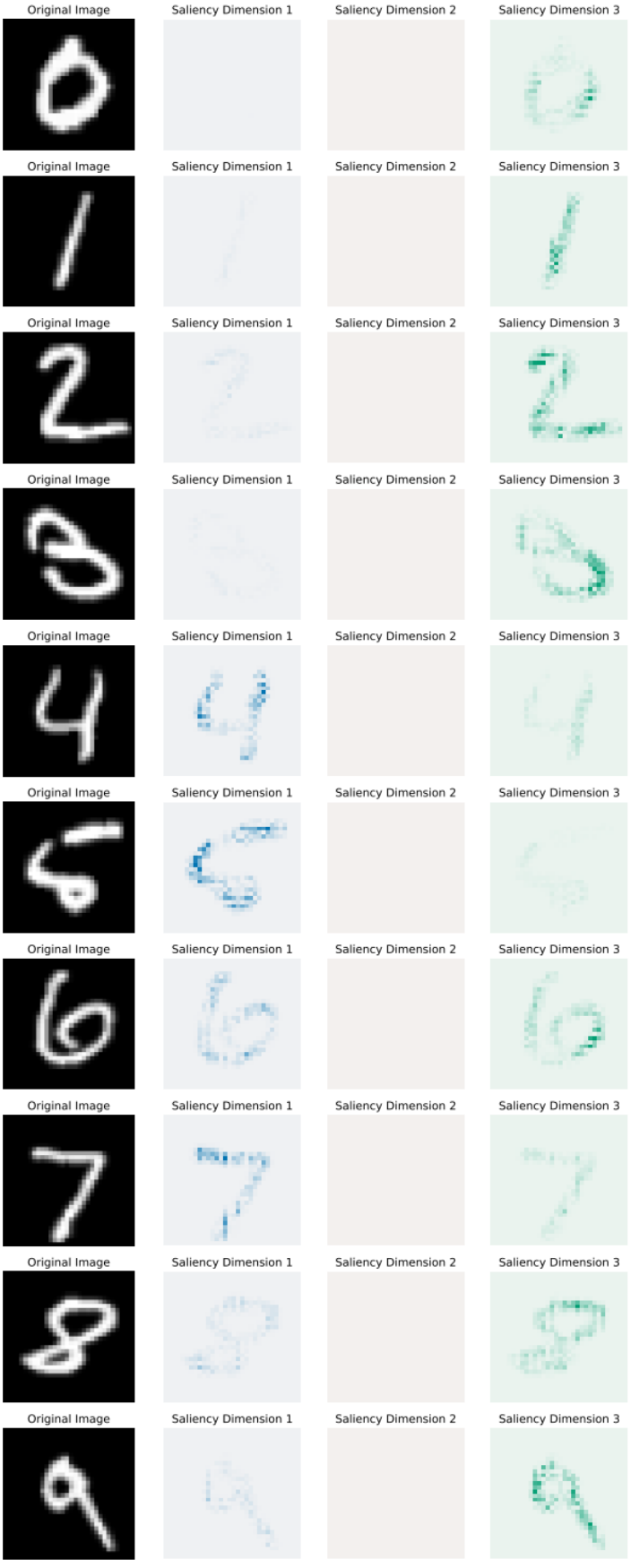


Figure 13. Saliency maps with Gradient Shap for TC-VAE with β=10, λ=0.001 on MNIST (Part 1).

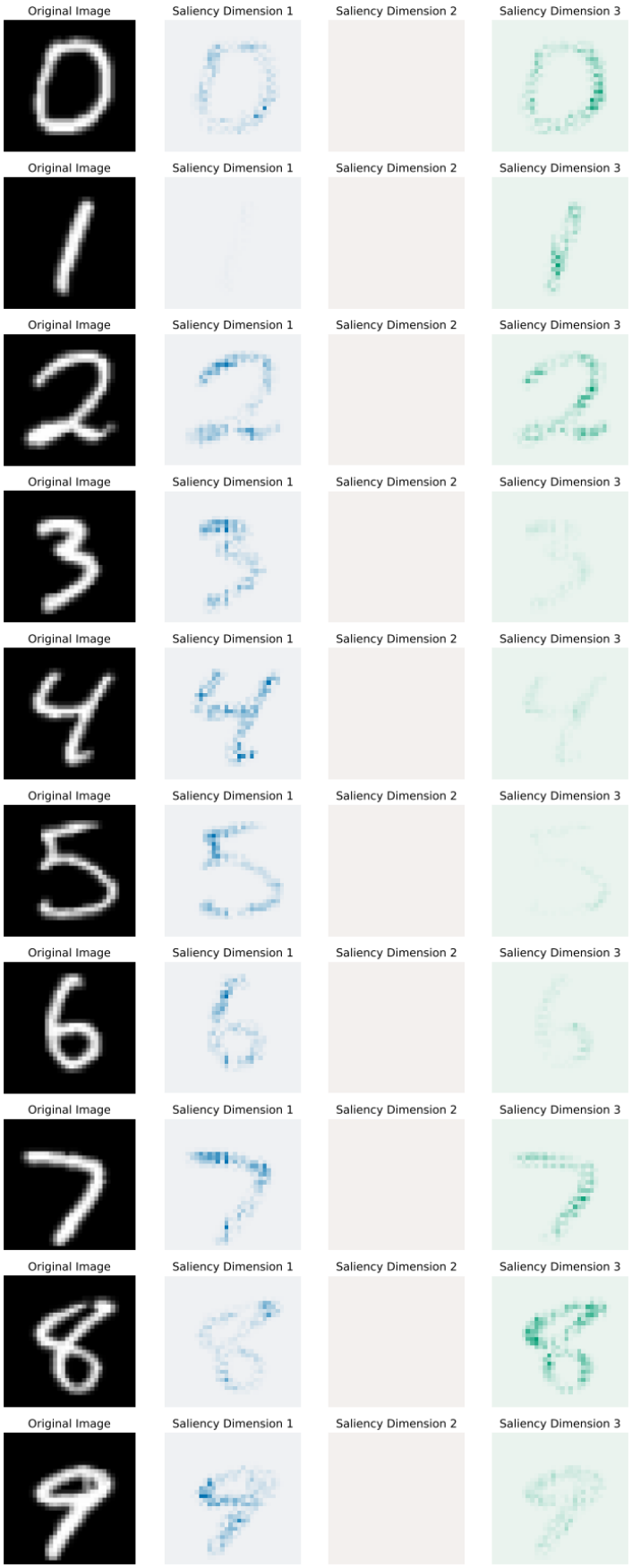


Figure 14. Saliency maps with Gradient Shap for TC-VAE with β=10, λ=0.001 on MNIST (Part 2).

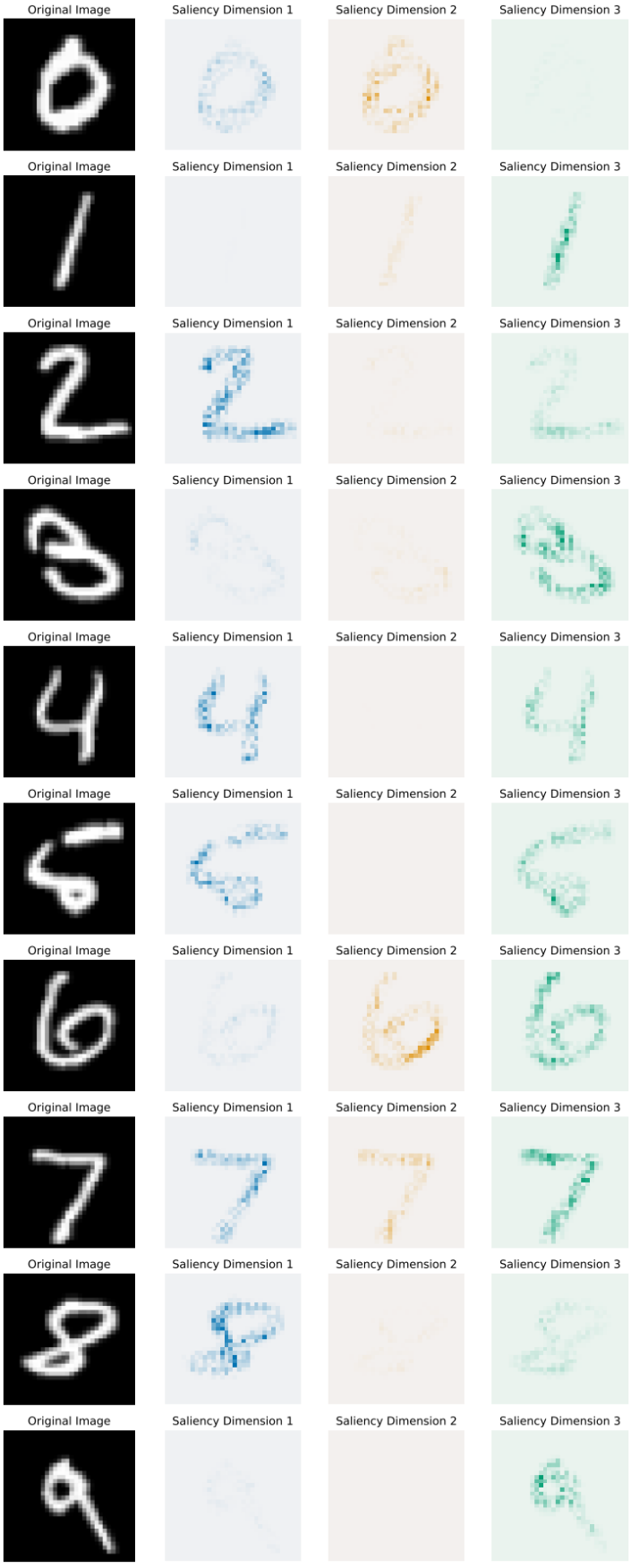


Figure 15. Saliency maps with Gradient Shap for TC-VAE with β=10, λ=0.005 on MNIST (Part 1).

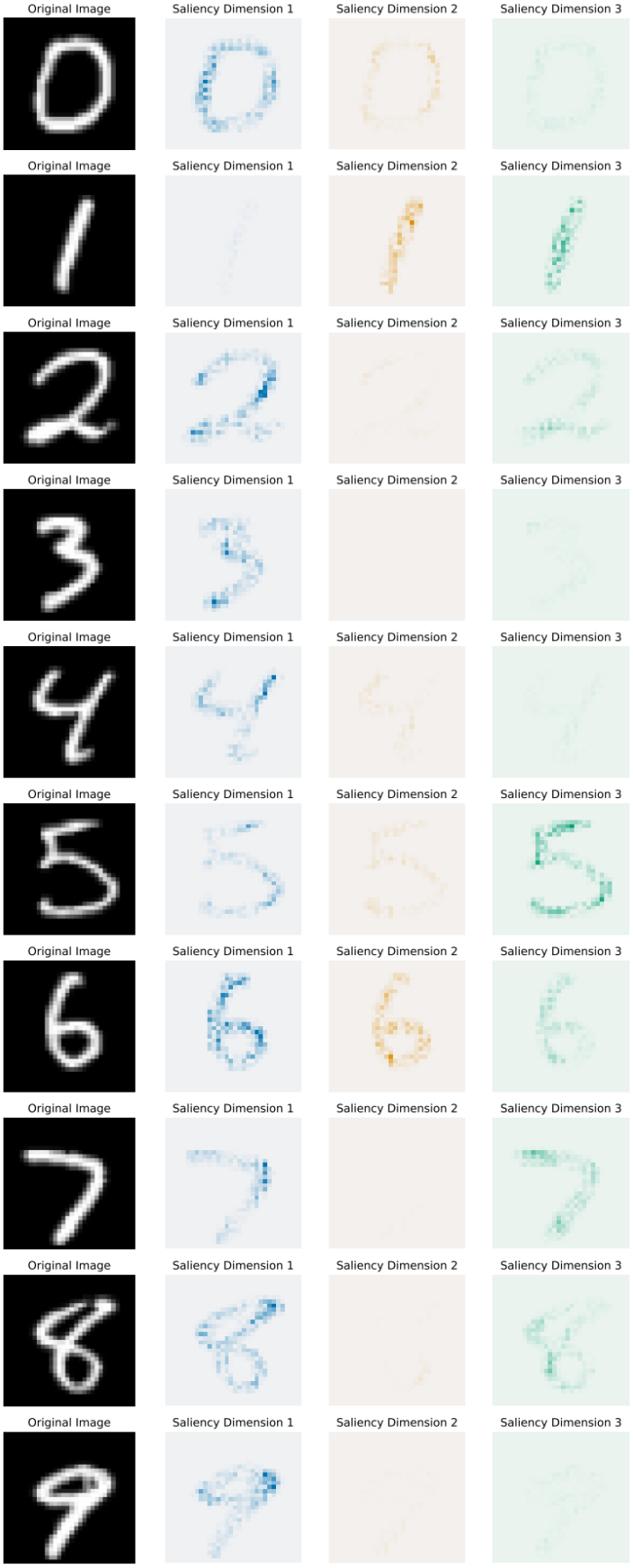


Figure 16. Saliency maps with Gradient Shap for TC-VAE with β=10, λ=0.005 on MNIST (Part 2).

D.5 Use Case: Evaluating Label-Free Feature Importance on the Attribution Prior VAEs

Below you see two tables with the mean (Table 8) and standard deviation (Table 9) of representation shifts of five runs, for different configurations of models (defined by rows) and different percentages of pixel perturbations applied after training those models (columns 5,10,..., 100) on the MNIST Dataset. The last column shows the test error of the trained model derived from the test set of the MNIST Dataset.

Model	β	λ	5	10	20	50	80	100	Test Loss
TC-VAE	1	0	0.58	2.84	15.96	28.35	28.35	28.35	176.4
		0.005	0.65	3.19	15.21	25.4	25.4	25.4	178.8
		0.01	0.7	3.47	18.05	31.66	31.66	31.66	176.7
		0.1	0.64	2.94	12.81	19.24	19.24	19.24	176.9
	5	0	0.83	2.8	9.08	14.11	14.11	14.11	169.8
		0.005	0.77	2.83	11.29	17.89	17.89	17.89	172.7
		0.01	0.84	3.19	13.76	22.8	22.8	22.8	169
		0.1	0.86	2.49	8.7	13.49	13.49	13.49	168.8
	10	0	1.17	3.16	9.73	15.04	15.04	15.04	156.41
		0.005	0.96	2.79	7.91	11.27	11.27	11.27	154.42
		0.01	0.95	2.7	10.22	17.21	17.21	17.21	155.93
		0.1	0.79	2.15	6.9	11.07	11.07	11.07	155.42
β-VAE	1	0	0.86	5.12	24.97	40.81	40.81	40.81	176.4
		0.005	0.67	3.11	14.49	23.22	23.22	23.22	178.8
		0.01	0.6	2.72	14.19	25.28	25.28	25.28	176.7
		0.1	0.5	2.37	11.14	17.37	17.37	17.37	176.9
	5	0	0.85	2.32	6.42	8.92	8.92	8.92	169.8
		0.005	0.8	2.76	12.27	20.62	20.62	20.62	172.7
		0.01	1.07	3.53	11.5	16.82	16.82	16.82	169
		0.1	0.75	2.37	8.15	13.01	13.01	13.01	168.8
	10	0	0.51	1.34	4.55	7.35	7.35	7.35	156.41
		0.005	0.72	1.97	4.93	6.4	6.4	6.4	154.42
		0.01	0.7	1.8	4.4	6.07	6.07	6.07	155.93
		0.1	0.63	1.56	3.72	4.92	4.92	4.92	155.42

Table 8. Mean representation shifts over 5 runs for percentages of pixel perturbations for the models β-VAE, and TC-VAE on the MNIST dataset with different disentanglement factor β and different regularization parameter, λ. The λ is for weighting the importance of the attribution prior to the model's loss function (See Equation 1). $\lambda = 0$ denotes that no attribution prior is used. The Test Loss is the loss of the model reconstructing the MNIST test dataset. The numbers 5,10,..., 100 represent the percentage of the important pixels of each input image perturbed.

Model	β	λ	5	10	20	50	80	100
TC-VAE	1	0	0.35	1.45	3.75	1.03	1.03	1.03
		0.005	0.22	0.81	1.83	1.32	1.32	1.32
		0.01	0.18	0.7	2.19	1.95	1.95	1.95
		0.1	0.16	0.62	1.49	1.52	1.52	1.52
	5	0	0.16	0.35	0.78	0.81	0.81	0.81
		0.005	0.15	0.55	1.78	1.12	1.12	1.12
		0.01	0.25	0.69	1.43	0.97	0.97	0.97
		0.1	0.13	0.37	1.12	0.78	0.78	0.78
	10	0	0.1	0.27	0.7	0.41	0.41	0.41
		0.005	0.08	0.2	0.51	0.46	0.46	0.46
		0.01	0.1	0.23	0.52	0.37	0.37	0.37
		0.1	0.08	0.21	0.39	0.37	0.37	0.37
β-VAE	1	0	0.18	0.8	2.5	1.5	1.5	1.5
		0.005	0.22	0.85	2.18	1.3	1.3	1.3
		0.01	0.22	0.98	2.73	2.01	2.01	2.01
		0.1	0.22	0.82	1.62	1.3	1.3	1.3
	5	0	0.21	0.55	1.21	0.8	0.8	0.8
		0.005	0.22	0.7	1.84	0.88	0.88	0.88
		0.01	0.19	0.72	2.23	1.19	1.19	1.19
		0.1	0.13	0.39	1.05	1.57	1.57	1.57
	10	0	0.29	0.67	1.72	0.95	0.95	0.95
		0.005	0.19	0.46	1	0.64	0.64	0.64
		0.01	0.18	0.56	2.07	1.36	1.36	1.36
		0.1	0.16	0.39	1.24	1.12	1.12	1.12

Table 9. Standard Deviation of the representation shifts over 5 runs for Table 8.

E Representations Learnt on Pretext Tasks

E.1 Qualitative analysis overview

We perform an extensive qualitative analysis by comparing the output saliency maps and top examples. Below in this section, we include figures of some notable cases of top examples and their saliency maps. Multiple groups of images are collected for different runs on the dataset.

In the case of the saliency maps, it is indeed true that for the same image, there are notable differences across different pretext tasks, since the highlighted regions are not the same. On the other hand, in the case of example importance the conclusion that "the top examples are rarely similar across various pretext tasks" does not coincide with the observations of the reproduced outputs. As seen in Figure 17, most of the top examples correspond to the right number, although in some images they do not coincide. Finally, we observe the synergy between the top examples and the feature importance.

Qualitative claim	Status
Saliency maps differences	Confirmed
Top example differences	Unconfirmed
Synergy	Confirmed

Table 10. Qualitative analysis

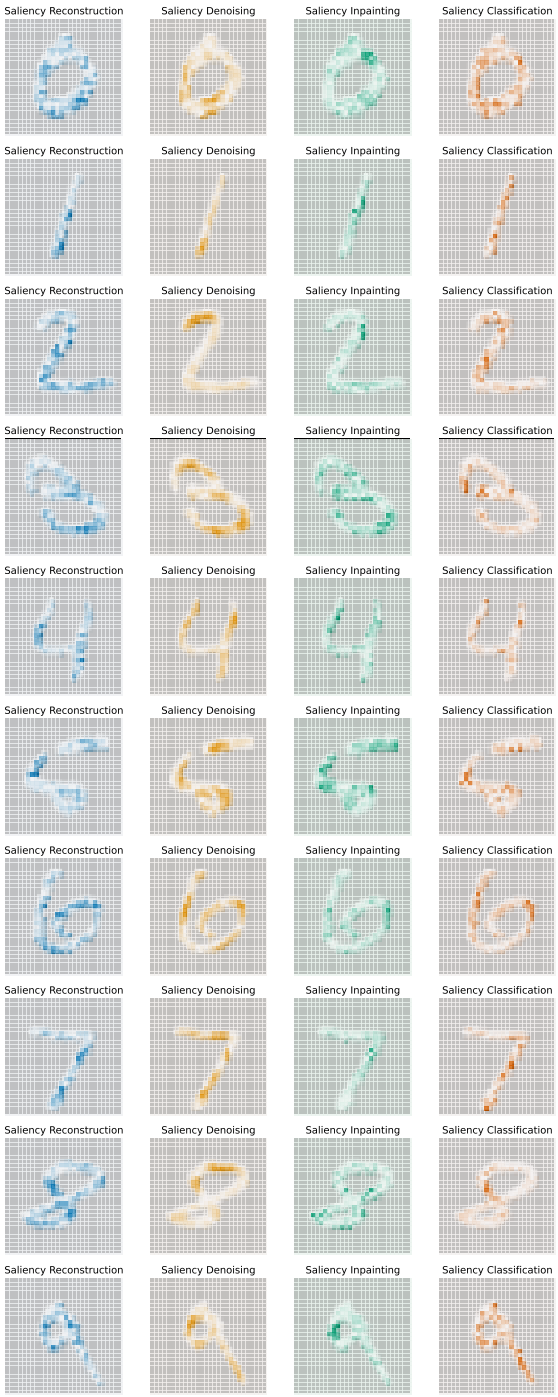


Figure 17. Reproduced label-free saliency maps for various pretext tasks (Run 0)

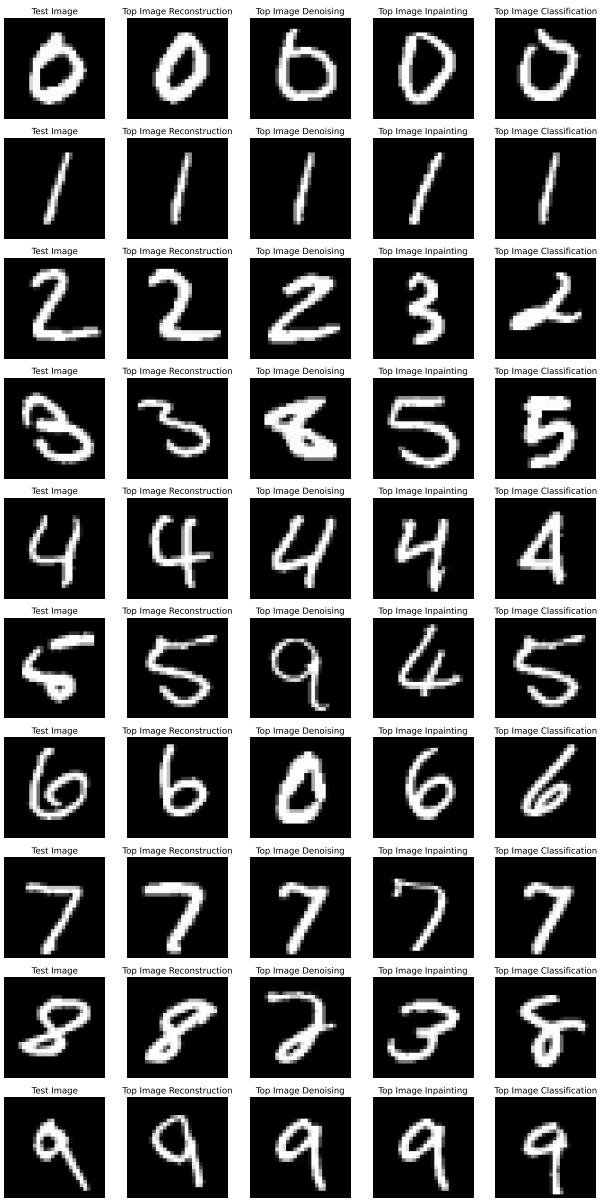


Figure 18. Reproduced label-free top examples for various pretext tasks (Run 0)