Finding NEM-U: Explaining unsupervised representation learning through neural network generated explanation masks

Bjørn Leth Møller [1] Christian Igel [1] Kristoffer Knutsen Wickstrøm [2] Jon Sporring [1] Robert Jenssen [1 2 3]
Bulat Ibragimov [1]

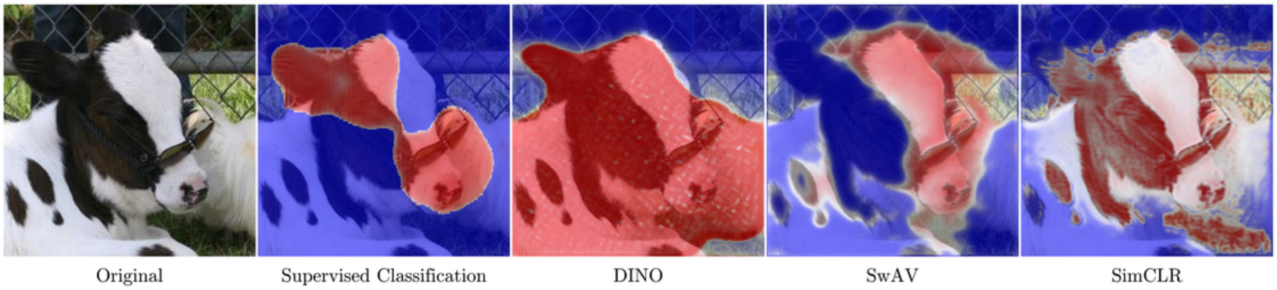


Figure 1. NEM-U explaining how important various parts of an image are for feature extractors trained using different methods. The feature extractor trained using supervised learning focuses on the ears and face of the object. The DINO pretrained feature extractor considers the entirety of all animals. The SwAV and SimCLR feature extractors look at both the focused object and the background, where SwAV is more focused than SimCLR. Explanations are generated without optimization on the image (taken from VOC data set).

Abstract

Unsupervised representation learning has become an important ingredient of today's deep learning systems. However, only a few methods exist that explain a learned vector embedding in the sense of providing information about which parts of an input are the most important for its representation. These methods generate the explanation for a given input after the model has been evaluated and tend to produce either inaccurate explanations or are slow, which limits their practical use. To address these limitations, we introduce the Neural Explanation Masks (NEM) framework, which turns a fixed representation model into a self-explaining system by augmenting it with a masking neural network. This network provides occlusion-based explanations in parallel to computing the representations during inference. We present an instance of this framework, the NEM-U (NEM using U-net structure) architecture, which leverages similarities between segmentation and occlusion-based explanation masks. Our experiments show that NEM-U generates explanations faster and with lower complexity compared to the current state-of-the-art while maintaining high accuracy as measured by locality.

1. Introduction

Explainable AI (XAI) is concerned with explaining the outputs of machine learning (ML) models to improve transparency and safety. In the context of representation learning, we want to know which parts of an input are most important for the representation (i.e., embedding) provided by

[1] Department of Computer Science, University of Copenhagen, Copenhagen, Denmark [2] Department of Physics and Technology, UiT The Arctic University of Norway, Tromsø [3] Norwegian Computing Center, Oslo, Norway. Correspondence to: Bjørn Leth Møller <bjm@di.ku.dk>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

the model. With the increasing importance of unsupervised learning in computer vision (Caron et al., 2021; Chen et al., 2020), speech and natural language processing (Devlin et al., 2019; Brown et al., 2020; Mohamed et al., 2022), and time series modelling (Wickstrøm et al., 2022; Pöppelbaum et al., 2022), there is a big need for XAI approaches that go beyond the supervised setting.

Explaining representations can be regarded as a general approach applicable to models that have been trained in both the supervised and unsupervised setting. A few methods exist that are designed for the task of explaining representations (Wickstrøm et al., 2023a; Crabbé & van der Schaar, 2022), but they suffer from some significant limitations, namely low-quality explanations or slow computational speed. Methods generating low-quality explanations are generally not desirable, whereas slow computational speed limits the methods' usefulness in many real-world scenarios where high inference speed is required, for example video processing and interactive applications.

Our goal is to generate high-quality explanations with low latency. To this end, we introduce the NEM (Neural Explanation Masks) framework, which takes a fixed model Φ and turns it into a self-explaining system by augmenting it with a *masking network* Ψ. The masking network is trained to segment a given input into areas of importance for the representation provided by Φ. Instead of running a post-hoc process to provide the explanation of a new input, we design a self-explaining system that provides the explanation in parallel with the output. In contrast to running an optimization process to find an occlusion mask, we learn to predict the mask. This speeds up the system during inference time at the cost of having to train the masking network once before using the system. Furthermore, connecting Ψ directly with the explained model Φ allows to leverage encodings at different levels of Φ.

The NEM approach is general in the sense that it is in principle not input-specific and applicable to any differentiable model Φ. However, our empirical evaluation focuses on explaining deep neural network embeddings of images. Outputting an occlusion mask for highlighting the important parts of an input can be viewed as a segmentation task. This insight allows us to build on results from the field of semantic segmentation. We do so by introducing the NEM-U (NEM using U-Net structure) architecture. In this instance of the NEM framework, the explained model and masking network can be viewed as the encoder and decoder of a U-Net, arguably one of the most popular neural network architectures for image segmentation (Ronneberger et al., 2015).

The main contributions of this study are the following:

- The NEM framework, which generates representation explanations by augmenting a given representation generating system with a masking network, alongside a loss function to train the masking network;
- The NEM-U architecture, a specific NEM architecture that leverages a U-Net encoder-decoder structure to improve mask generation;
- Experiments showing that NEM-U is considerably faster and provides less complex explanations compared to the state of the art, while still keeping the quality of explanations high.

2. Related Work

Explaining vector representations of data, as opposed to scalar predictions, is a new direction within XAI. There are two main approaches in this direction. The first is to adapt existing XAI methods to handle the representation learning setting. Most notably, the label-free XAI framework (Crabbé & van der Schaar, 2022; Chen et al., 2023) offers an auxiliary function that allows XAI methods designed for the supervised setting to be used in the unsupervised representation learning setting. The label-free XAI framework has the advantage that it enables the use of existing methods. However, it has been shown to perform worse compared to alternative approaches (Wickstrøm et al., 2023a). The second direction is to design new XAI methods that are particularly suited for the unsupervised representation learning setting. Notably, the RELAX framework (Wickstrøm et al., 2023a) uses a masking approach to measure similarities between masked and unmasked representations for explainability. RELAX has been extended to explaining relations between representations (Lin et al., 2023) and has been applied in several areas including medicine (Wickstrøm et al., 2023b) and social science (Feng et al., 2023). RELAX has demonstrated excellent performance, but the method requires a computationally demanding randomized optimization procedure for each explained sample. Bertolini et al. (2023) have proposed another method for directly explaining vector representation with a focus on convolutional neural networks. They aggregate the saliency maps of latent representations extracted from different embedding layers of the model to be explained.

Explaining neural networks through occlusions is a longstanding and active area in XAI research. Zeiler & Fergus (2014) introduced one of the earliest occlusion-based XAI methods, where an input image was systematically occluded with a rectangular mask while monitoring the output of the network. Petsiuk et al. (2018) suggested to randomly mask out parts of the image, which reduced computational demand and improved performance. Other lines of work focus on optimizing a mask for each instance (Fong & Vedaldi, 2017; Fong et al., 2019; MacDonald et al., 2019;

Kolek et al., 2020). The recent work of Bhalla et al. (2023) has explored whether simultaneously optimizing masks of the training data used to train a model while training the model results in models which can be better explained with occlusion-based methods. However, apart from the RELAX framework, occlusion-based methods have generally been designed with supervised predictions in mind. An approach, alternative to input occlusions, is to generate an explanation from the model gradients. This can be achieved using Integrated Gradients (Sundararajan et al., 2017), Gradient SHAP (Lundberg & Lee, 2017) and different variants of GradCAM (Selvaraju et al., 2017; Chattopadhay et al., 2018; Jiang et al., 2021).

The idea of explaining one neural network through another is well-established in the XAI literature. Taghanaki et al. (2019) introduced InfoMask, where an encoder-decoder network jointly masks and predicts an input image. Zhmoginov et al. (2021) introduced an information-bottleneck approach to salient region discovery, where a variational autoencoder was used to generate salient regions for a classification model. Schulz et al. (2020) proposed an information-theoretic approach where a readout network would mask out irrelevant regions and only passes relevant regions to the predictor. However, all of these methods are designed for supervised learning and are not applicable in the representation XAI setting.

3. Methodology

In the following, we first present the general NEM (Neural Explanation Masks) framework, which allows for training a masking network that can generate occlusion-based explanations for differentiable representation learning systems. Then we derive an objective function for training NEM models. Finally, we introduce the NEM-U (NEM using U-Net structure) architecture, a specific instance of the NEM framework utilizing a U-Net style encoder-decoder architecture.

3.1. Neural Explanation Masks (NEM)

We decided to follow an occlusion-based explanation paradigm, which has proven to be effective in the representation learning setting (Wickstrøm et al., 2023a). As we wanted a framework that allows for computationally inexpensive generation of explanations, we worked on a self-explaining system. Furthermore, we wished to have a fast method for gauging the quality of a given explanation. This led to the NEM (Neural Explanation Masks) framework. It comprises two key components: the *frozen model* Φ and the *masking network* Ψ. The former is the differentiable model that generates the representations for which explanations are desired. It is frozen in the sense that its weights and therefore its input-output behaviour do not change when using it together with NEM. The masking network generates occlusion masks conditioned on the input to Φ, indicating important regions of the input.

Given Φ, the masking network is trained on a corpus of data $\mathbf{X}$. Let us assume that the inputs are images, $x \in \mathbb{R}^{i \times j}$, and $\Phi : \mathbb{R}^{i \times j} \to \mathbb{R}^d$, for some integers i, j, and d. For each input $x \in \mathbf{X}$, the masking network generates an explanation mask $m = \Psi(\Phi, x) \in [0,1]^{i \times j}$. The mask is used to occlude parts of x. The masked input can, for example, be computed as $x_\mathrm{m} = x \odot m$, where $\odot$ denotes the Hadamard product (see equation (1) below for a more general definition). Both x and x_m are then processed by the frozen model to generate the representation pair $(r, r_\mathrm{m}) = (\Phi(x), \Phi(x_\mathrm{m}))$. Finally, the triplet (r, r_m, m) is used to optimize the masking network according to an objective function $\mathcal{L}$ as we discuss in Section 3.2.

Since network Ψ is trained to generate masks for various inputs, it is possible to predict explanation masks for new observations without additional optimization. The NEM framework allows for quickly measuring the quality of a given explanation mask, both during training the system as well as for evaluating an explanation mask generated by the system after deployment. An estimate of whether a predicted mask removes important information of the input can be computed by measuring the difference between the representation of the original and masked input.

Figure 2 depicts the NEM framework. The framework assumes the differentiability of all components, as this permits gradient-based optimization of the masking network. To exploit information from the encoding process of the frozen model, the masking network can extract input representations generated by the frozen network, although this is not required. This will be used in the NEM-U architecture described in Section 3.3.

3.2. Objective Function

The question arises how to train the masking network Ψ. In the following, we describe the objective function we use for NEM training. The goal is to train a masking network Ψ that predicts an accurate explanation mask for a given input. We assume that the input is drawn from some unknown input distribution and that we have access to training data $\mathbf{X}$ sampled from the same distribution. In the occlusion-based masking paradigm, the task of generating explanation masks can be viewed as segmenting the input into areas of relevance and irrelevance, similar to the signal-distractor framework by Bhalla et al. (2023). In the language of Fong & Vedaldi (2017), we play a preservation game where we trade off information and mask sparsity to find a sparse set of features that are maximally informative for the given model Φ.

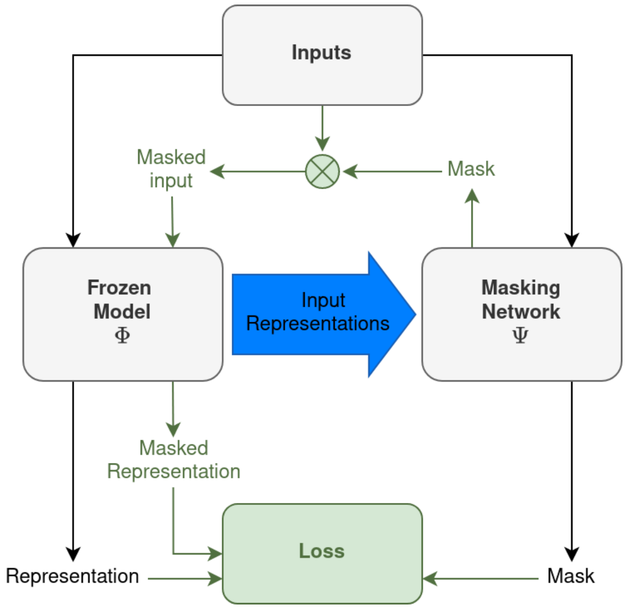


Figure 2. Overview of the NEM framework. The black parts of the diagram indicate paths that will be used during training and after deployment. The green parts indicate additional information flow during training. The blue arrow symbolizes latent representations extracted from the frozen model to inform the masking network. In the case of NEM-U, skip connections are used to extract different levels of input representations. NEM architectures such as NEM-U can be run in parallel during inference.

Similar to Wickstrøm et al. (2023a), we define a subset of informative features (i.e., a subset of pixels) as a set of features that results in a latent representation which is close to the representation of the full set of features (i.e., the full image). When computing a representation for a subset of features, the features not in the subset are replaced by random values drawn i.i.d. from some perturbation distribution $\mathcal{V}$ as suggested by MacDonald et al. (2019). Thus, for an input x and a mask $m = \Psi(\Phi, x) \in [0,1]^{i\times j}$ generated by the masking network, we compute the masked image as

$$x_{\mathrm{m}} = \Psi(\Phi, x) \odot x + (1 - \Psi(\Phi, x)) \odot v \qquad (1)$$

with $v \sim \mathcal{V}$. The representation of the masked image should be close to the representation of the original image. Therefore, we minimize

$$P_d(\Phi, x) = d\big(\Phi(x), \Phi(x_{\mathrm{m}})\big) \qquad (2)$$

for some distance measure d. At the same time, we would like to minimize the conflicting objective of having a sparse occlusion mask as measured by the 1-norm $\|\Psi(\Phi, x)\|_1$. Since our goal is to segment the image into areas of information (foreground/signal) and non-information (background/distractor), we include a general regularization term $B(x)$ that penalizes masking solutions where the results are non-binary. This penalization term is needed since Ψ outputs continuous masks. Putting it all together, we arrive at the general loss function

$$\mathcal{L}(\mathbf{X}) = \sum_{x\in\mathbf{X}} \mathop{\mathbb{E}}_{v\sim\mathcal{V}}[\lambda_1 P_d(\Phi, x)] + \lambda_2\|\Psi(\Phi, x)\|_1 + B(x), \qquad (3)$$

where the trade-offs between the three objectives are controlled by weighting parameters $\lambda_1, \lambda_2 > 0$.

By setting $\lambda_1 = 1$, $B(x) = 0$, $\mathbf{X} = \{x\}$ and replacing $\Psi(\Phi, x)$ by a single mask to be optimized, we recover the objective function of the rate-distortion explanation framework as introduced by MacDonald et al. (2019). Alternatively, by setting $\lambda_1 = 1$, $B(x) = 0$, $d = \|\cdot\|_1$, and replacing $\Psi(\Phi, x)$ with a mask for each image to be optimized, we recover the data distillation loss $\mathcal{L}_{\mathrm{QFA}}$ of the signal-distractor framework proposed by Bhalla et al. (2023).

3.3. NEM using U-net (NEM-U)

In this section, we introduce a specific instance of the NEM architecture termed NEM-U. As the masking network essentially segments the input into areas that are important or unimportant for the frozen model, we can exploit ideas from the segmentation literature, more specifically the widely used U-net encoder-decoder architecture (Ronneberger et al., 2015). The U-Net is commonly used for 2D and 3D image segmentation but has also been applied to other tasks such as time series segmentation (Perslev et al., 2019).

In the NEM-U (NEM using U-net) architecture, the masking network Ψ is a U-Net, where the frozen model Φ acts as the U-Net encoder. Only the decoder part is trained when learning explanations in the NEM-U setup while the encoder is frozen.

In the U-Net architecture, the decoder generates a segmentation mask based on several representations computed by the encoder at different depths. In NEM-U, this latent representation extraction mechanism, indicated in blue in Figure 2, allows the masking model to benefit from multiple levels of representations generated by the explained model, potentially improving performance. Furthermore, removing the need for an unique encoder for Ψ reduces the overall number of model parameters. Our conjecture is that direct access to different layers of representations in the explained model Φ facilitates faster training and better performance of the masking network Ψ. Therefore, we focus on the NEM-U architecture in the experimental evaluation.

4. Experiments and Results

This section describes our empirical evaluation of the NEM-U architecture for explaining computer vision models. We start by describing how we set up the NEM-U models for the experiments. Then we outline the overall experimental protocol used for model evaluation. Finally, we show the results of our experiments

4.1. Implementation

When implementing NEM-U, design choices include the network architecture, masking method, distance function, and objective function weighting coefficients. In the following experiments, the NEM-U models utilized untrained standard U-net decoders for the masking network. The number of trainable parameters of the masking networks varied from 9M to 10.1M, depending on the specific model that was explained. To ensure that the output masks stayed within the range of 0 to 1, a sigmoid activation function was applied in the final layer. Pixel scaling was employed as the masking methodology. That is, each pixel of an image is simply multiplied with its corresponding mask, i.e., $\mathcal{V} = \delta(0)$ implying $\Psi(\Phi, x) \odot x + (1 - \Psi(\Phi, x)) \odot v = \Psi(\Phi, x) \odot x$. For the distance measure d we selected the negative cosine similarity.

The regularizer $B(x)$ considered the representation of both the signal represented by $\Psi(\Phi, x)$ and its distractor defined by the inverse $\overline{x}_{\mathrm{m}} = (1 - \Psi(\Phi, x))$. To encourage solutions with binary masks, we penalize solutions where the masked image and the inverse masked image are close in latent space. This distance is defined as $N_d(\Phi, x)$. The intuition is that if an object is only partially occluded by the mask, it will influence both latent representations. Furthermore, we scale $N_d(\Phi, x)$ with $P_d(\Phi, x)$, since we only care about binary masks when our masked image is already close to the original image in latent space. Thus we set

$$B(x) = -N_d(\Phi, x) P_d(\Phi, x). \tag{4}$$

Plugging these choices into (3) yields the loss function

$$\mathcal{L}(\mathbf{X}) = \sum_{x \in \mathbf{X}} \big(\lambda_1 - N_d(\Phi, x)\big) P_d(\Phi, x) + \lambda_2 \|\Psi(\Phi, x)\|_1 \tag{5}$$

with

$$x_{\mathrm{m}} = \Psi(\Phi, x) \odot x, \tag{6}$$

$$\overline{x}_{\mathrm{m}} = (1 - \Psi(\Phi, x)) \odot x, \tag{7}$$

$$P_d(\Phi, x) = -\cos\big(\Phi(x), \Phi(x_{\mathrm{m}})\big), \tag{8}$$

$$N_d(\Phi, x) = -\cos\big(\Phi(x_{\mathrm{m}}), \Phi(\overline{x}_{\mathrm{m}})\big). \tag{9}$$

We set $\lambda_1 = 1.5$ and $\lambda_2 = 1$ since this choice yielded sparse coherent masks for all explained models. These values were determined via visual inspection of masks generated on the validation split of the training data. An evaluation of different implementation choices is summarized in Appendix B.

4.2. Experimental Evaluation

In this section, we present our empirical evaluation of the NEM architecture, including descriptions of evaluation metrics, datasets, and baseline methods.

Metrics. To evaluate the proposed methodology and compare it with existing methods, we considered several established metrics from prior studies on quantitative evaluation of XAI. First, we compute *locality* metrics (Zhang et al., 2018; Arras et al., 2022), which measure the overlap between an explanation and a reference segmentation mask or bounding box provided by human annotators. Locality acts as a proxy for how much the explanation agrees with how a human would explain what the important content of an image is. There exist numerous versions of the locality metric, and here we considered *relevance rank* accuracy (Arras et al., 2022) as used by Wickstrøm et al. (2023a) and *relevance mass* accuracy (Arras et al., 2022). For relevance rank, pixels are sorted according to their importance score, and we measure how many of the top-k pixels are within the ground truth mask, where k is set to be the number of pixels in the ground truth mask. A high relevance rank score indicates that the explanation aligns well with the human annotation. For relevance mass, the ratio of positive attributions within the ground truth mask to the sum of all positive attributions is calculated. A high relevance mass indicates that the explanation puts a lot of attention on the same region as the human annotation and little on other regions. Second, we computed *complexity* metrics (Bhatt et al., 2021; Chalasani et al., 2020), which measure how sparse and therefore comprehensible an explanation is. The rationale is that a good explanation should highlight small regions with few pixels such that it is easy for a human to interpret the output. We considered two popular metrics from the complexity family, namely *complexity*, measured as the entropy of the explanation mask (Bhatt et al., 2021), and *sparseness* (Chalasani et al., 2020), measured as the Gini index of the absolute values of the explanation mask. Furthermore, we determined the computational speed of each method by measuring the average time per image needed to generate explanations for 1000 random images. The final speed score was calculated by averaging the results for each individual explained model. A standard evaluation measure in explainability research is *faithfullness*, which has originally been defined for supervised tasks. We adapted *faithfullness* for the unsupervised setting, see Appendix A for details. Apart from the speed score, all metrics were adapted from the Quantus library (Hedström et al., 2023).

Datasets. Following prior works (Wickstrøm et al., 2023a; Petsiuk et al., 2018), we used the COCO dataset (Lin et al., 2014) and the VOC dataset (Everingham et al., 2010). For the COCO dataset, we randomly sampled 1000 images and bounding boxes from the COCO validation set for evaluation purposes and randomly sampled 10000 images from the COCO train set for training NEM-U models. Similarly, we randomly sampled 1000 images and bounding boxes from the VOC test set for evaluation and used the entire VOC train set (2501 images) for NEM-U model training. Like prior works on unsupervised XAI (Wickstrøm et al., 2023a), we combined bounding boxes for each class into a single collective annotation for the entire image.

Baseline methods. We compared NEM-U to established methods from the representation learning explainablity literature. We considered the RELAX method (Wickstrøm et al., 2023a) and its associated U-RELAX method, where explanations from RELAX are thresholded based on uncertainty estimates. We also looked at the label-free explainability method by Crabbé & van der Schaar (2022), which allows XAI methods designed for the supervised setting to be used in the unsupervised representation learning setting. Using this approach, we considered the following well-known explainability methods: Integrated Gradients (Sundararajan et al., 2017), Gradient SHAP (Lundberg & Lee, 2017) and Saliency (Simonyan et al., 2013). All explanation masks were normalized to the range $[0, 1]$ to ensure fair comparisons when calculating complexity. That is, we considered the normalized mask $m' = (\Psi(\Phi, x) - \min(\Psi(\Phi, x)))/(\max(\Psi(\Phi, x)) - \min(\Psi(\Phi, x)))$, where minimum and maximum are computed pixel-wise.

Architectures and pre-training methods. We evaluated the different unsupervised explainability methods for explaining features extracted from two widely used deep learning architectures, namely the ResNet50 (He et al., 2016) and the vision transformer (Kolesnikov et al., 2021). We investigated several pre-training methods both with and without labels. For the ResNet50, we studied models trained using supervised learning and self-supervised learning using the SimCLR method (Chen et al., 2020) and the SwAV method (Caron et al., 2020). For the vision transformer, we considered self-supervised training using the DINO methods. The supervised ResNet50 was evaluated as a feature extractor, meaning we explained the output of the last embedding layer of the model. The vision transformer was adapted to the NEM-U framework by reshaping the extracted latent representations to fit the U-net decoders of the masking network. The supervised weights and weights for DINO were obtained from the timm library (Wightman, 2019), whereas the SwAV and SimCLR weights were obtained from the Pytorch Lightning Bolts library (Falcon & The PyTorch Lightning team, 2019).

NEM-U training. All NEM-U models were trained in the same fashion for the evaluations. All models were optimized using the prodigy optimizer (Mishchenko & Defazio, 2023) for 10 epochs on either the COCO or the VOC training data depending on the evaluation dataset. There was no overlap between training data and evaluation data, as we wished to gauge the performance of the NEM-U models in an inference setting. No data augmentations were used during training.

Table 1. Results of the runtime experiments. These experiments have been conducted on a NVIDIA GeForce RTX 4090.

Method	Time in seconds
RELAX	2.048
Integrated Gradients	0.072
Gradient SHAP	0.011
Saliency	0.007
NEM-U (Ours)	**0.002**

4.3. Results

The results for speed are summarized in Table 1, the results for locality and complexity are summarized in Table 2. Examples of explanation masks generated during the evaluation can be seen in Figure 1, Figure 3 and Appendix C. NEM-U achieved the highest speed with an average latency of 0.002 seconds per image, whereas the other methods ranged from 0.007 seconds per image achieved by the Saliency method to 2.050 seconds per image achieved by RELAX. In general, NEM-U yielded the lowest complexity metrics across all models and datasets. When measuring locality using relevancy mass, NEM-U achieved the best values across all datasets and models, whereas it obtained the second or third-best locality score when measuring relevancy rank depending on whether it is evaluated on the VOC or COCO dataset.

5. Discussion

Results. The results of our experiments indicate that the NEM-U model creates a mask with a locality comparable to the RELAX methodology and better than the other methods. Furthermore, the masks have the sparsest explanations (measured in complexity) in general and are generated faster than any other methods during inference. Notable NEM-U is approx. 1000 times faster than the current state-of-the-art approach RELAX with similar locality metrics.

Locality metrics. There was a notable difference between performance in locality across the two locality metrics. While RELAX performed best in terms of relevancy ranking, NEM-U performed best in terms of the relevancy mass. This indicates that most of the mass of NEM-U explanations were located inside the bounding boxes, but its ability

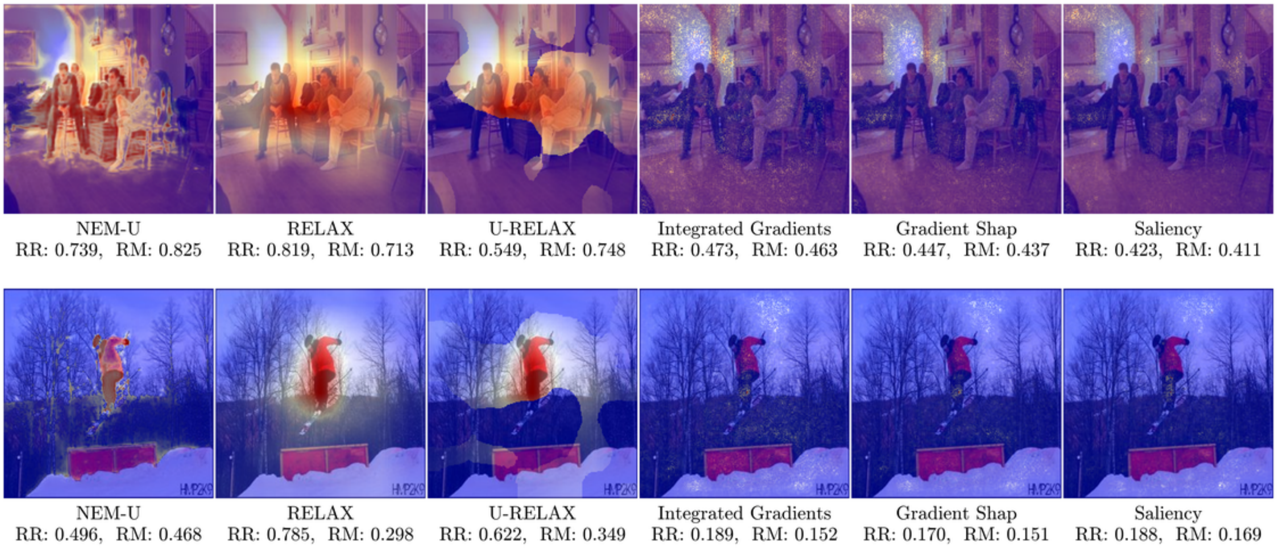


Figure 3. Different XAI methods applied to a ResNet50 model trained using SimCLR on an image from the VOC (top) and the COCO (bottom) validation set, respectively. RR and RM denote the relevancy rank score and the relevancy mass, respectively. Note that the explanation masks of NEM-U varies across datasets due to the differences in training data.

to rank features was worse than that of RELAX. Notably, even in the cases where the complexity of U-RELAX (e.g., DINO on the VOC dataset) was lower than NEM-U, NEM-U still achieved the highest relevancy mass. Thus, the better relevancy mass score was not just a mere result of NEM-U's explanations having lower complexity. The sparsity of the NEM-U explanations probably also reduced its ranking score, indicated by U-RELAX performing worse than both RELAX and NEM-U on the VOC dataset.

Set-of-features vs. additive feature explanations. The loss (3), which is used to train the NEM-U models, does not encourage the models to create an *additive feature explanation*, that is, an explanation mask that ranks all individual feature (in our case pixels) contributions of an input. Similar to the works by Fong et al. (2019) and Kolek et al. (2020; 2023), our loss function (3) encourages set-of-features explanations, induced by $B(x)$ as defined by (4). Therefore, a NEM-U explanation is best viewed as an unordered subset of the input features. These features are important for the model but the individual ranking between features based on the gradual scores is less informative. Comparing additive feature explanations and set-of-feature explanations directly is difficult as pointed out by Fong et al. (2019), which is an issue since all methods we compare against are arguably additive feature explanations. To our knowledge, there are no meaningful metrics that compare the locality of the set-of-feature explanations approach to the additive feature explanation fairly. This is why we opted for both relevance rank, which benefits additive feature explanations, and relevance mass, which benefits set-of-feature explanations.

Whether set-of-features or additive feature explanations are better is an open question, but Kolek et al. (2020) argue that sets of features should be desirable in situations where single features do not carry much meaning, such as individual pixels' contributions to an image classification. Another benefit of the sets of features is that validating the explanation is fast and straightforward since the masked image representing these features can be passed directly through the model and the output can be compared with the result of passing the unmasked input through the network.

Qualitative evaluation across datasets. Figure 3 shows examples contrasting different XAI methods; more examples are presented in Appendix C alongside images depicting the worst NEM-U explanations. The figures demonstrate that the NEM-U models generated visual explanations of low complexity compared to other methods. The Ψ is a trained model and thus will output masks that are biased by the specific dataset it is trained on. An example of this is that NEM-U had the tendency to generate much denser explanations on the COCO dataset. This might be due to the size of the dataset and particular qualities of each dataset, e.g., COCO in general has smaller objects of interest compared to VOC (Lin et al., 2014). These denser, more binary explanations also potentially explain why the relevancy rank of NEM-U dropped on the COCO set.

Latency considerations. NEM-U was faster than all other methods during inference. Our method only requires a

Table 2. Experimental results for the two locality metrics relevancy rank and relevancy mass as well as for the two complexity metrics complexity and sparseness. In case of the locality metrics, higher values indicate better performance. NEM-U performs best in terms relevancy mass, whereas RELAX yields the highest relevancy ranks. NEM-U ranks second when considering the relevancy ranks on VOC, whereas U-RELAX ranks second place in all other settings. Lower values indicate better performance for complexity, whereas higher values indicate better performance for sparseness. NEM-U wins 12 out of 16 games whereas U-RELAX wins 4 of 16 games.

Model	Method	Relevance Rank ↑		Relevance Mass ↑		Complexity ↓		Sparseness ↑	
		VOC	COCO	VOC	COCO	VOC	COCO	VOC	COCO
Supervised	RELAX	**0.719**	**0.689**	0.599	0.556	10.626	10.628	0.336	0.335
	U-RELAX	0.666	0.646	0.679	0.628	9.988	9.987	0.637	0.639
	Integrated Gradients	0.567	0.540	0.569	0.534	10.253	10.277	0.551	0.542
	Gradient SHAP	0.568	0.541	0.571	0.534	10.251	10.275	0.552	0.543
	Saliency	0.555	0.419	0.537	0.427	10.443	10.227	0.460	0.547
	NEM-U (Ours)	0.691	0.644	**0.823**	**0.697**	**9.390**	**9.633**	**0.740**	**0.712**
DINO	RELAX	**0.761**	**0.725**	0.642	0.592	10.559	10.568	0.391	0.385
	U-RELAX	0.714	0.687	0.726	0.669	**9.969**	9.974	**0.649**	0.647
	Integrated Gradients	0.576	0.537	0.584	0.532	10.128	10.161	0.600	0.589
	Gradient SHAP	0.575	0.539	0.585	0.536	10.065	10.100	0.616	0.605
	Saliency	0.565	0.522	0.547	0.503	10.302	10.305	0.526	0.524
	NEM-U (Ours)	0.738	0.638	**0.749**	**0.688**	10.124	**9.847**	0.573	**0.656**
SwAV	RELAX	**0.721**	**0.685**	0.589	0.544	10.665	10.670	0.305	0.300
	U-RELAX	0.656	0.635	0.662	0.615	10.021	10.027	0.621	0.617
	Integrated Gradients	0.513	0.476	0.517	0.476	10.307	10.320	0.531	0.525
	Gradient SHAP	0.511	0.476	0.516	0.476	10.300	10.309	0.535	0.530
	Saliency	0.489	0.456	0.489	0.453	10.460	10.462	0.451	0.449
	NEM-U (Ours)	0.686	0.582	**0.736**	**0.641**	**9.808**	**9.578**	**0.687**	**0.779**
SimCLR	RELAX	**0.709**	**0.665**	0.612	0.561	10.597	10.607	0.368	0.360
	U-RELAX	0.619	0.593	0.661	0.610	**9.949**	9.960	**0.660**	0.655
	Integrated Gradients	0.530	0.479	0.536	0.483	10.260	10.271	0.551	0.546
	Gradient SHAP	0.528	0.478	0.535	0.483	10.246	10.258	0.556	0.551
	Saliency	0.508	0.460	0.505	0.459	10.386	10.371	0.488	0.496
	NEM-U (Ours)	0.692	0.562	**0.710**	**0.656**	9.993	**9.340**	0.649	**0.829**

forward pass through the frozen model and the masking network, whereas the second fastest method, the saliency method, has to do both a forward and a backward pass through the explained model. Since the explained model in all our experiments was larger than the masking model, NEM-U was faster. It is important to stress that our method was evaluated for all metrics in the inference setting. Since our method requires training, it is most efficient when the amount of explanations needed is sufficiently large. If only a few explanations are needed, other methods that do not require training might be more suitable.

Future work. Future work could explore how to further improve the model's performance in general and its ability to rank pixels in particular. One potential avenue would be to build on previous work on the rate-distortion framework. Kolek et al. (2023) have, in the supervised setting, achieved excellent explanations by leveraging shearlet or wavelet image representations. The cost of using these representations was low inference speed, which could potentially be mitigated by our method's high inference speed. Another avenue for improving masking would be to carry over methodology from online learning by allowing the masking network to optimize its explanation mask for a given input. Some implementation decisions should be explored further, for example, more principled ways of choosing $B(x)$ in (3) should be developed. Ideally, we would want a solution based on the $\|\cdot\|_0$ norm to get more binary masks, but such an approach is not well-suited for gradient-based learning. Kolek et al. (2023) have tried to solve this issue via a spatial energy penalty term.

Looking at the COCO results (e.g., lower part in Figure 3), we can see that our approach can generate binarized low-complexity explanation masks. Thus, the NEM framework could potentially serve as the basis for a method to translate a given image classification model into a corresponding segmentation model.

6. Conclusion

We have introduced the NEM framework, which allows for transforming a differentiable model into a self-explaining system by augmenting it with a masking network. Furthermore, we have introduced the NEM-U architecture, which leverages a U-Net style encoder-decoder setup to implement the NEM framework. Our experiments indicate that the NEM-U models produce explanations on par with the state of the art while having lower complexity and much lower latency.

Impact Statement

This paper presents work whose goal is to advance the field of explainable AI. Explainable AI is important for understanding and improving fairness, transparency, trust, and accuracy in AI-based decision-making. Applications requiring the generation of many explanations (i.e., online analysis of videos) do not only directly profit from the high accuracy and low latency of the NEM approach, but they also become more resource efficient (Wright et al., 2023).

Acknowledgement

This work was supported by the Novo Nordisk Foundation under grant NFF20OC0062056, by the Research Council of Norway under grant no. 309439, and by the Pioneer Centre for AI under DNRF grant no. P1.

References

Arras, L., Osman, A., and Samek, W. Clevr-xai: A benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40, 2022.

Bertolini, M., Clevert, D.-A., and Montanari, F. Explaining, evaluating and enhancing neural networks' learned representations. In *International Conference on Artificial Neural Networks (ICANN)*, pp. 269–287. Springer, 2023.

Bhalla, U., Srinivas, S., and Lakkaraju, H. Verifiable feature attributions: A bridge between post hoc explainability and inherent interpretability. *arXiv preprint arXiv:2307.15007*, 2023.

Bhatt, U., Weller, A., and Moura, J. M. F. Evaluating and aggregating feature-based model explanations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:9912–9924, 2020.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *International Conference on Computer Vision (ICCV)*, pp. 9650–9660, October 2021.

Chalasani, P., Chen, J., Chowdhury, A. R., Wu, X., and Jha, S. Concise explanations of neural networks using adversarial training. In *International Conference on Machine Learning (ICML)*, pp. 1383–1391. PMLR, 2020.

Chattopadhay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847. IEEE, 2018.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pp. 1597–1607. PMLR, 2020.

Chen, Y., Bijlani, N., Kouchaki, S., and Barnaghi, P. Interpreting differentiable latent states for healthcare timeseries data. In *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*, 2023.

Crabbé, J. and van der Schaar, M. Label-free explainability for unsupervised models. In *International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 4391–4420. PMLR, 17–23 Jul 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019)*, pp. 4171–4186, 2019.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88: 303–338, 2010.

Falcon, W. and The PyTorch Lightning team. PyTorch Lightning, March 2019.

Feng, X. F., Li, C., and Zhang, S. Visual uniqueness in peer-to-peer marketplaces: Machine learning model development, validation, and application. *R&R at Journal of Consumer Research*, (4665286), 2023.

Fong, R., Patrick, M., and Vedaldi, A. Understanding deep networks via extremal perturbations and smooth masks. In *International Conference on Computer Vision (ICCV)*, pp. 2950–2958, 2019.

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *International Conference on Computer Vision (ICCV)*, pp. 3449–3457, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Hedström, A., Weber, L., Krakowczyk, D., Bareeva, D., Motzkus, F., Samek, W., Lapuschkin, S., and Höhne, M. M. M. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research*, 24(34): 1–11, 2023.

Jiang, P.-T., Zhang, C.-B., Hou, Q., Cheng, M.-M., and Wei, Y. LayerCAM: Exploring hierarchical class activation maps for localization. *IEEE Transactions Imgage Processing*, 30:5875–5888, 2021.

Kolek, S., Nguyen, D. A., Levie, R., Bruna, J., and Kutyniok, G. A rate-distortion framework for explaining black-box model decisions. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pp. 91–115. Springer, 2020.

Kolek, S., Nguyen, D. A., Levie, R., Bruna, J., and Kutyniok, G. Cartoon explanations of image classifiers. In *European Conference on Computer Vision (ECCV)*, pp. 443–458. Springer, 2022.

Kolek, S., Windesheim, R., Andrade-Loarca, H., Kutyniok, G., and Levie, R. Explaining image classifiers with multiscale directional image representation. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 18600–18609, 2023.

Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., Heigold, G., Uszkoreit, J., Beyer, L., Minderer, M., Dehghani, M., Houlsby, N., Gelly, S., Unterthiner, T., and Zhai, X. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Lin, C., Chen, H., Kim, C., and Lee, S.-I. Contrastive corpus attribution for explaining representations. In *International Conference on Learning Representations (ICLR)*, 2023.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pp. 740–755. Springer, 2014.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

MacDonald, J., Wäldchen, S., Hauch, S., and Kutyniok, G. A rate-distortion framework for explaining neural network decisions. *arXiv preprint arXiv:1905.11092*, 2019.

Mishchenko, K. and Defazio, A. Prodigy: An expeditiously adaptive parameter-free learner. *arXiv preprint arXiv:2306.06101*, 2023.

Mohamed, A., Lee, H., Borgholt, L., Havtorn, J. D., Edin, J., Igel, C., Kirchhoff, K., Li, S.-W., Livescu, K., Maaløe, L., Sainath, T. N., and Watanabe, S. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1179–1210, 2022.

Nguyen, A.-p. and Martínez, M. R. On quantitative aspects of model interpretability. *arXiv preprint arXiv:2007.07584*, 2020.

Perslev, M., Hejselbak Jensen, M., Darkner, S., Jennum, P. J., and Igel, C. U-Time: A fully convolutional network for time series segmentation applied to sleep staging. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4417–4428, 2019.

Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference*, 2018.

Pöppelbaum, J., Chadha, G. S., and Schwung, A. Contrastive learning based self-supervised time-series analysis. *Applied Soft Computing*, 117:108397, 2022.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.

Schulz, K., Sixt, L., Tombari, F., and Landgraf, T. Restricting the flow: Information bottlenecks for attribution. In *International Conference on Learning Representations (ICLR)*, 2020.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, pp. 3319–3328. PMLR, 2017.

Taghanaki, S. A., Havaei, M., Berthier, T., Dutil, F., Di Jorio, L., Hamarneh, G., and Bengio, Y. InfoMask: Masked variational latent representation to localize chest disease. In *Medical Image Computing and Computer Assisted Intervention (MICCAI 2019)*, pp. 739–747. Springer, 2019.

Wickstrøm, K., Kampffmeyer, M., Mikalsen, K. Ø., and Jenssen, R. Mixing up contrastive learning: Self-supervised representation learning for time series. *Pattern Recognition Letters*, 155:54–61, 2022.

Wickstrøm, K. K., Trosten, D. J., Løkse, S., Boubekki, A., Øyvind Mikalsen, K., Kampffmeyer, M. C., and Jenssen, R. Relax: Representation learning explainability. *International Journal Computer Vision*, 131:1584–1610, 2023a.

Wickstrøm, K. K., Østmo, E. A., Radiya, K., Øyvind Mikalsen, K., Kampffmeyer, M. C., and Jenssen, R. A clinically motivated self-supervised approach for content-based image retrieval of CT liver images. *Computerized Medical Imaging and Graphics*, pp. 102239, 2023b.

Wightman, R. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Wright, D., Igel, C., Samuel, G., and Selvan, R. Efficiency is not enough: A critical perspective of environmentally sustainable AI. *arXiv preprint arXiv:2309.02065*, 2023.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, pp. 818–833. Springer, 2014.

Zhang, J., Bargal, S. A., Lin, Z., Brandt, J., Shen, X., and Sclaroff, S. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126 (10):1084–1102, 2018.

Zhmoginov, A., Fischer, I., and Sandler, M. *Information-Bottleneck Approach to Salient Region Discovery*, pp. 531–546. Springer International Publishing, 2021. doi: 10.1007/978-3-030-67664-3_32.

A. Faithfullness

The question of how to measure faithfullness is not well studied in the literature of unsupervised XAI. One issue is that measuring distances in embedding space is complicated due to the high dimensionality of the space, and thus standard faithfullness concepts such as deletion and insertion curves (Fong et al., 2019) might be misleading. In our case, a comparison of methods using faithfullness may be biased. The current SOTA approach, RELAX, is a pixel-level attribution method, whereas the NEM framework is a set-wise attribution method, which hampers a direct comparison as already pointed out by Kolek et al. (2023). In the following, we consider two metrics to gauge the faithfullness of the NEM framework, which we regard to be less affected by the issues outlined above.

Inspired by Kolek et al. (2023), we define a faithfullness score by measuring the distance of the original image and the masked image using cosine similarity. To account for the sparsity of the mask, we divide the distance score by the L1 norm of the explanation mask, creating our final faithfullness score. To evaluate NEM-U using this score, we compared it with other ways of generating occlusion masks: We considered U-RELAX and additionally defined UMT-RELAX as a method that generates a binary mask by thresholding the uncertainty map of RELAX at its mean. We added the inverse of the NEM-U mask, referred to as NEM-U-INVERSE, to understand the importance of what is discarded by NEM-U. Finally, we included a naive solution randomly removing half of the pixels. The results can be seen in Table 3. We find that NEM-U achieves either the highest or second highest score.

An alternative approach is to evaluate the faithfullness of the different methods on a downstream task by explaining a classifier trained in a supervised manner. This allows for using standard faithfullness metrics. We apply all methods evaluated in the main paper to explain the last embedding layer of a ResNet50 classifier trained on ImageNet. The created attribution map of the embedding layer is used as a proxy explanation for the output layer, meaning we use the attribution map as if it was the attribution map for the output layer. We measure the faithfullness using the monotonicity metric proposed by Nguyen & Martínez (2020). We extract 11000 images from the ImageNet validation set, where 10000 is used to train the NEM-U model and the other 1000 is used for evaluation. The results are shown in Table 4.

Table 3. Unsupervised measure of faithfulness. Accuracy is measured as the cosine similarity between the masked vector representation and the original vector representation. Complexity is the L_1 norm of the explanation mask. The score is an overall measure of mask performance and is calculated as accuracy divided by complexity.

Model	Method	Accuracy ↑		Complexity ↓		Score ↑	
		VOC	COCO	VOC	COCO	VOC	COCO
Supervised	NEM-U	0.87	*0.89*	*0.25*	*0.28*	*3.48*	*3.21*
	NEM-U-INVERSE	0.82	0.80	0.75	0.72	1.09	1.11
	U-RELAX	*0.90*	**0.90**	**0.20**	**0.20**	**4.45**	**4.50**
	UMT-RELAX	**0.91**	0.88	0.40	0.37	2.39	2.46
	Naive Solution	0.70	0.71	0.50	0.50	1.41	1.42
DINO	NEM-U	**0.93**	**0.92**	*0.39*	*0.33*	2.38	*2.77*
	NEM-U-INVERSE	0.68	0.67	0.61	0.67	1.11	1.00
	U-RELAX	0.87	*0.87*	**0.20**	**0.20**	**4.35**	**4.28**
	UMT-RELAX	*0.88*	0.87	0.37	0.37	*2.48*	2.46
	Naive Solution	0.69	0.70	0.50	0.50	1.38	1.40
SwAV	NEM-U	*0.90*	0.87	**0.23**	**0.14**	**3.99**	**6.21**
	NEM-U-INVERSE	**0.95**	**0.95**	0.77	0.86	1.23	1.11
	U-RELAX	0.88	*0.88*	*0.24*	*0.25*	*3.60*	*3.58*
	UMT-RELAX	0.88	0.88	0.44	0.44	2.04	2.03
	Naive Solution	0.78	0.79	0.50	0.50	1.57	1.57
SimCLR	NEM-U	*0.90*	0.82	*0.21*	**0.06**	**4.33**	**14.09**
	NEM-U-INVERSE	**0.93**	**0.96**	0.79	0.94	1.18	1.02
	U-RELAX	0.84	*0.84*	**0.20**	*0.21*	*4.20*	*4.06*
	UMT-RELAX	0.85	0.85	0.42	0.42	2.12	2.10
	Naive Solution	0.68	0.69	0.50	0.50	1.36	1.38

Table 4. Faithfullness measurement by explaining a downstream supervised classifcation task on 1000 images extracted from the validation set of ImageNet.

Method	Monotonicity Correlation ↑
NEM-U	**0.568**
RELAX	0.463
U-RELAX	0.441
Integrated Gradients	*0.512*
Grad shap	0.505
Saliency	0.400

B. Additional Experimental Results

B.1. DINO Attribution Maps

To explore whether the NEM framework provides any benefits to models which are inherently interpretable, we compared our main results with using the attribution maps of the DINO model as an explanation. The results can be seen in Table 5. In general, the results indicate that the explanations provided by NEM-U have better locality and lower complexity.

Table 5. Comparing the DINO attention maps with using NEM-U to explain the model

	Relevancy Mass ↑		Relevancy Rank ↑		Complexity ↓		Sparseness ↑	
	VOC	COCO	VOC	COCO	VOC	COCO	VOC	COCO
Attention Map	0.687	0.646	0.687	**0.669**	10.161	10.169	**0.588**	0.586
NEM-U	**0.749**	**0.688**	**0.738**	0.638	**10.124**	**9.847**	0.573	**0.656**

B.2. Choice of Masking Scheme

Previous papers using masking based approaches (Kolek et al., 2022; 2023; Fong et al., 2019) have leveraged more sophisticated masking schemes during training compared to pixel scaling as is done in this paper. A common method is *noise injection*, where the input data is replaced with Gaussian noise in proportion to the masking value. Recent work has indicated that this approach is not helpful in the unsupervised setting (Wickstrøm et al., 2023a). Since proper noise injection require estimating the mean and variance of the training data, it would be interesting to determine, whether using a more advanced masking scheme yields benefits. To study this, we compared our main results with variants where we used noise injection. The results are summarized in Table 6 and indicate that there is no obvious benefit to choosing either intervention scheme.

Table 6. Comparing different intervention distributions.

Model	Intervention	Relevancy Rank ↑		Relevancy Mass ↑		Complexity ↓		Sparseness ↑	
		VOC	COCO	VOC	COCO	VOC	COCO	VOC	COCO
Supervised	Pixel scaling	**0.691**	**0.644**	**0.823**	**0.697**	**9.390**	**9.633**	**0.740**	**0.712**
	Noise injection	0.568	0.557	0.503	0.651	10.763	10.662	0.160	0.276
Dino	Pixel scaling	**0.738**	0.638	**0.749**	**0.688**	10.124	**9.847**	**0.573**	**0.656**
	Noise injection	0.719	**0.650**	0.709	0.495	**10.181**	10.059	0.508	0.571
SwAV	Pixel scaling	**0.686**	**0.582**	0.736	**0.641**	9.808	9.578	0.687	0.779
	Noise injection	0.549	0.524	**0.802**	0.571	**8.367**	**8.598**	**0.926**	**0.918**
SimCLR	Pixel scaling	0.692	0.562	0.710	**0.656**	9.993	**9.340**	0.649	**0.829**
	Noise injection	**0.698**	**0.579**	**0.751**	0.648	**9.872**	10.179	**0.652**	0.583

B.3. Hyperparameter Sweep

To understand how stable the NEM-U method is to choices of hyperparameter λ_1 and λ_2 , we conducted a hyperparameter sweep in the neighbourhood of our selected hyperparameters of all models on both datasets. The results can be seen in

Table 7, Table 8, Table 9, Table 10, Table 11, Table 12, Table 13, and Table 14. The results indicate that the NEM-U framework is robust to changes in its loss hyperparameters.

Table 7. Hyperparameter sweep results showing the Relevancy Mass for supervised classifier on the VOC dataset.

	λ_1= 1	λ_1=1.5	λ_1=2
λ_2 = 0.5	0.486	0.479	0.461
λ_2 = 1	0.853	0.823	0.740
λ_2 = 1.5	0.818	0.782	0.706

Table 8. Hyperparameter sweep results showing the Relevancy Mass for DINO on the VOC dataset.

	λ_1= 1	λ_1=1.5	λ_1=2
λ_2 = 0.5	0.738	0.662	0.553
λ_2 = 1	0.753	0.749	0.717
λ_2 = 1.5	0.798	0.805	0.781

Table 9. Hyperparameter sweep results showing the Relevancy Mass for SimCLR on the VOC dataset.

	λ_1= 1	λ_1=1.5	λ_1=2
λ_2 = 0.5	0.721	0.534	0.534
λ_2 = 1	0.759	0.710	0.600
λ_2 = 1.5	0.762	0.702	0.586

C. Visulization of Results

C.1. Comparing Explainability Methods Across Various Models and Images

In this appendix, we have added some visual examples of how the various methods handle images from different data sets and models, which can be seen in Figure 4, Figure 5, Figure 6, Figure 7, Figure 8, and Figure 9.

C.2. Worst Results of NEM-U

To qualitatively explore the failure modes of NEM-U, we have included the nine images where the cosine similarity between the original image and the masked image produced by NEM-U were lowest for each pair of model and dataset, see Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, Figure 15, Figure 16, and Figure 17.

Table 10. Hyperparameter sweep results showing the Relevancy Mass for SwaV on the VOC dataset.

	λ_1= 1	λ_1=1.5	λ_1=2
λ_2 = 0.5	0.749	0.584	0.543
λ_2 = 1	0.665	0.736	0.512
λ_2 = 1.5	0.739	0.739	0.652

Table 11. Hyperparameter sweep results showing the Relevancy Mass for supervised trained classifier on the COCO dataset.

	λ_1= 1	λ_1=1.5	λ_1=2
λ_2 = 0.5	0.713	0.441	0.478
λ_2 = 1	0.784	0.697	0.679
λ_2 = 1.5	0.441	0.656	0.585

Table 12. Hyperparameter sweep results showing the Relevancy Mass for DINO on the COCO dataset.

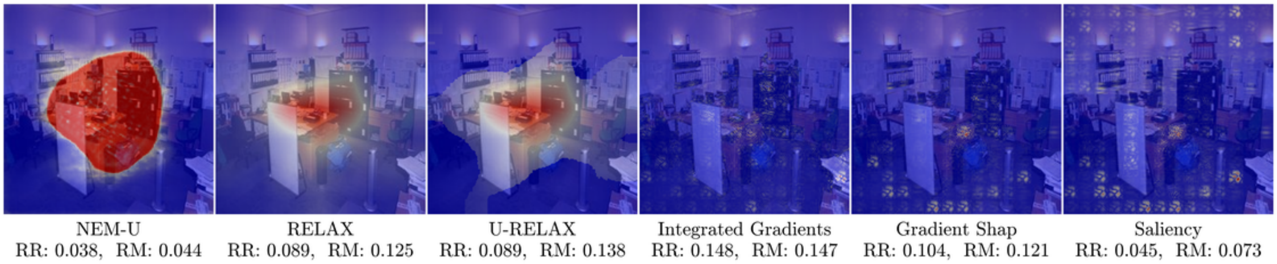
	λ_1= 1	λ_1=1.5	λ_1=2
λ_2 = 0.5	0.439	0.439	0.443
λ_2 = 1	0.833	0.688	0.646
λ_2 = 1.5	0.776	0.774	0.747

Table 13. Hyperparameter sweep results showing the Relevancy Mass for SimCLR on the COCO dataset.

	λ_1= 1	λ_1=1.5	λ_1=2
λ_2 = 0.5	0.684	0.523	0.507
λ_2 = 1	0.691	0.655	0.572
λ_2 = 1.5	0.703	0.650	0.583

Table 14. Hyperparameter sweep results showing the Relevancy Mass for SwaV on the COCO dataset.

	λ_1= 1	λ_1=1.5	λ_1=2
λ_2 = 0.5	0.630	0.504	0.478
λ_2 = 1	0.645	0.641	0.547
λ_2 = 1.5	0.718	0.644	0.546



NEM-U
RR: 0.038, RM: 0.044

RELAX
RR: 0.089, RM: 0.125

U-RELAX
RR: 0.089, RM: 0.138

Integrated Gradients
RR: 0.148, RM: 0.147

Gradient Shap
RR: 0.104, RM: 0.121

Saliency
RR: 0.045, RM: 0.073

Figure 4. Comparing XAI methods run on a vision transformer model trained using DINO on an image from the VOC test set. RR denotes the relevancy rank score for the method and RM represent the relevancy mass.

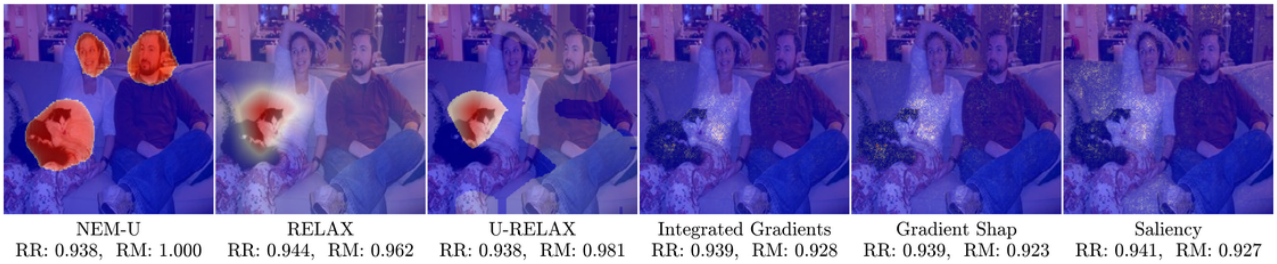


Figure 5. Comparing methods run on a ResNet50 model trained using a supervised classification task on an image from the VOC test set. RR is the relevancy rank score for the method and RM represents the relevancy mass.

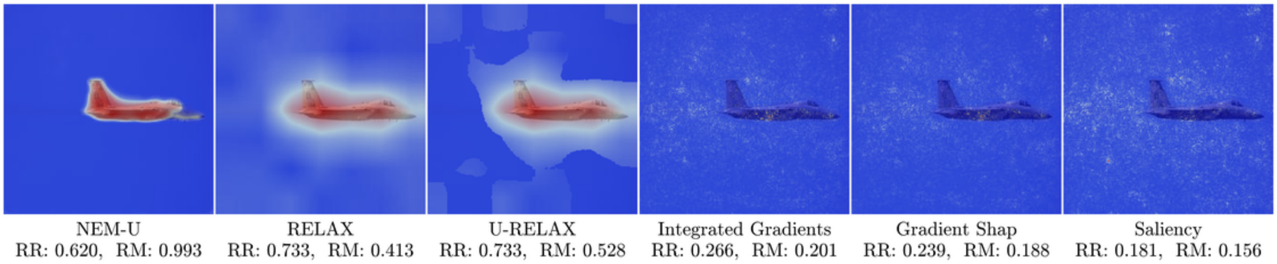


Figure 6. Comparing methods run on a ResNet50 model trained using SwAV on an image from the VOC test set. RR is the relevancy rank score for the method and RM represents the relevancy mass.

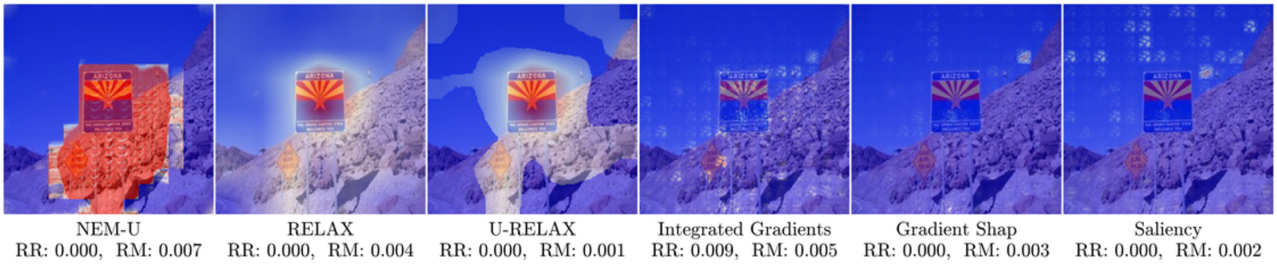


Figure 7. Comparing methods run on a vision transformer model trained using DINO on an image from the COCO validation set. RR is the relevancy rank score for the method and RM represents the relevancy mass.

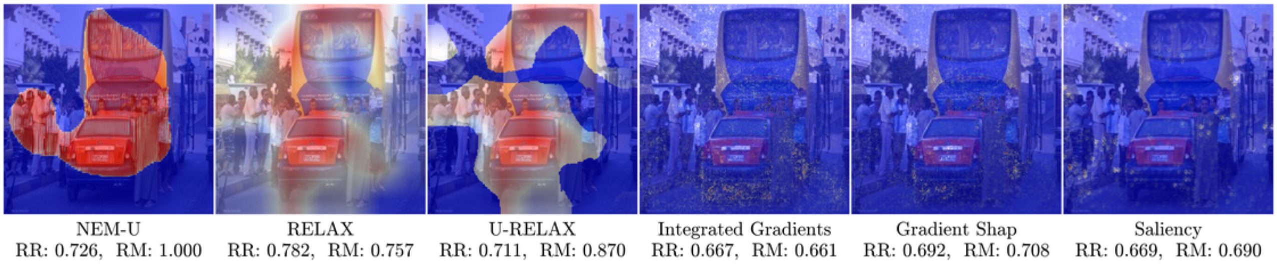


Figure 8. Comparing methods run on a ResNet50 model trained using a supervised classification task on an image from the COCO validation set. RR is the relevancy rank score for the method and RM represents the relevancy mass.

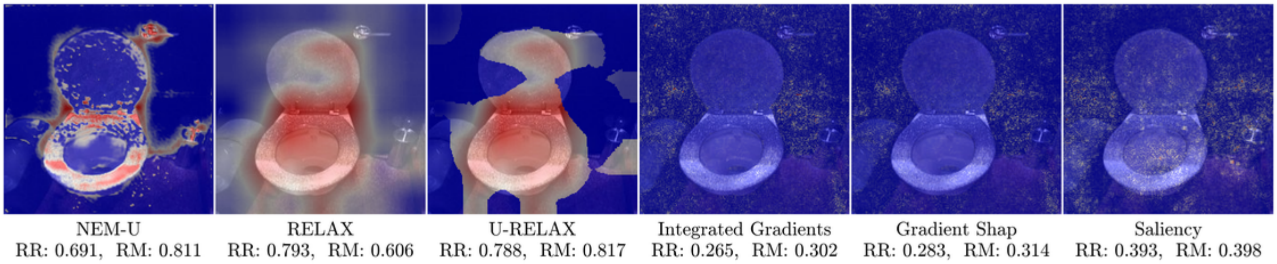


Figure 9. Comparing methods run on a ResNet50 model trained using SwAV model on an image from the COCO validation set. RR is the relevancy rank score for the method and RM represents the relevancy mass.

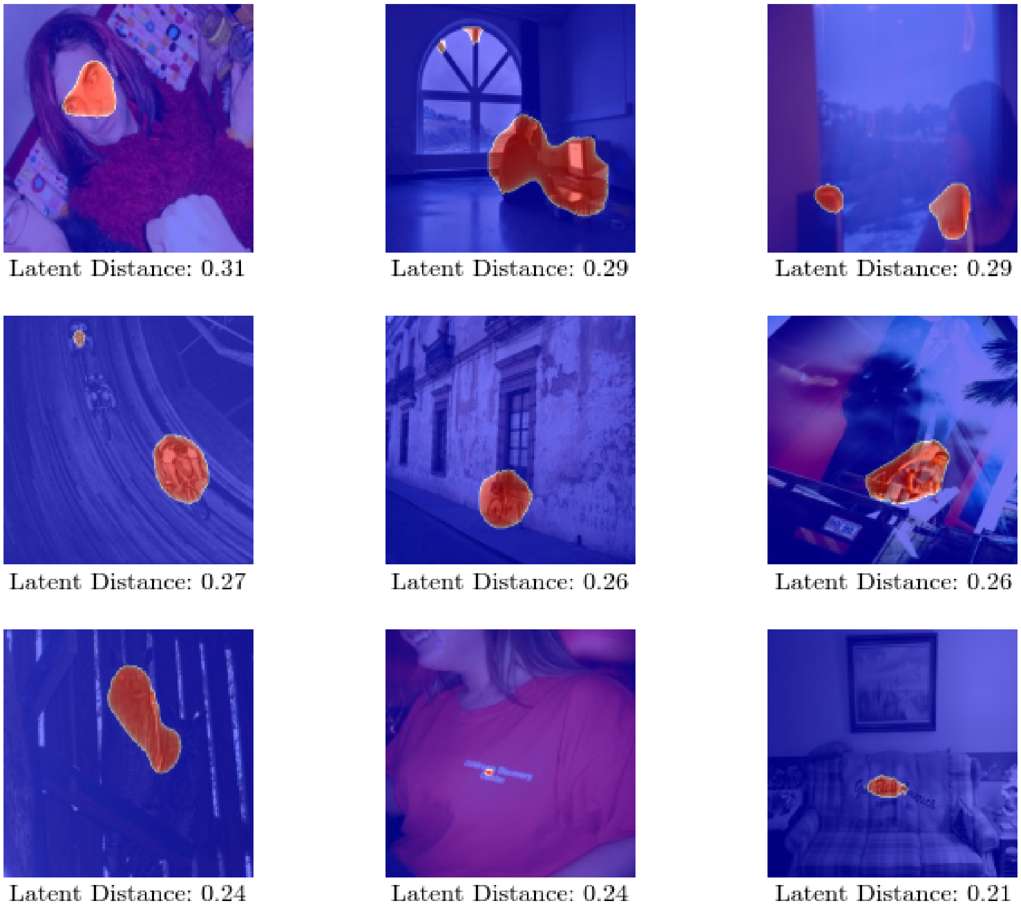


Figure 10. The nine images from the VOC test set for which the masked image had the largest distance to the original image in latent space. The distance was measured using cosine similarity. The masks were produced by NEM-U trained to explain a ResNet50 classifier trained using a supervised classification task.

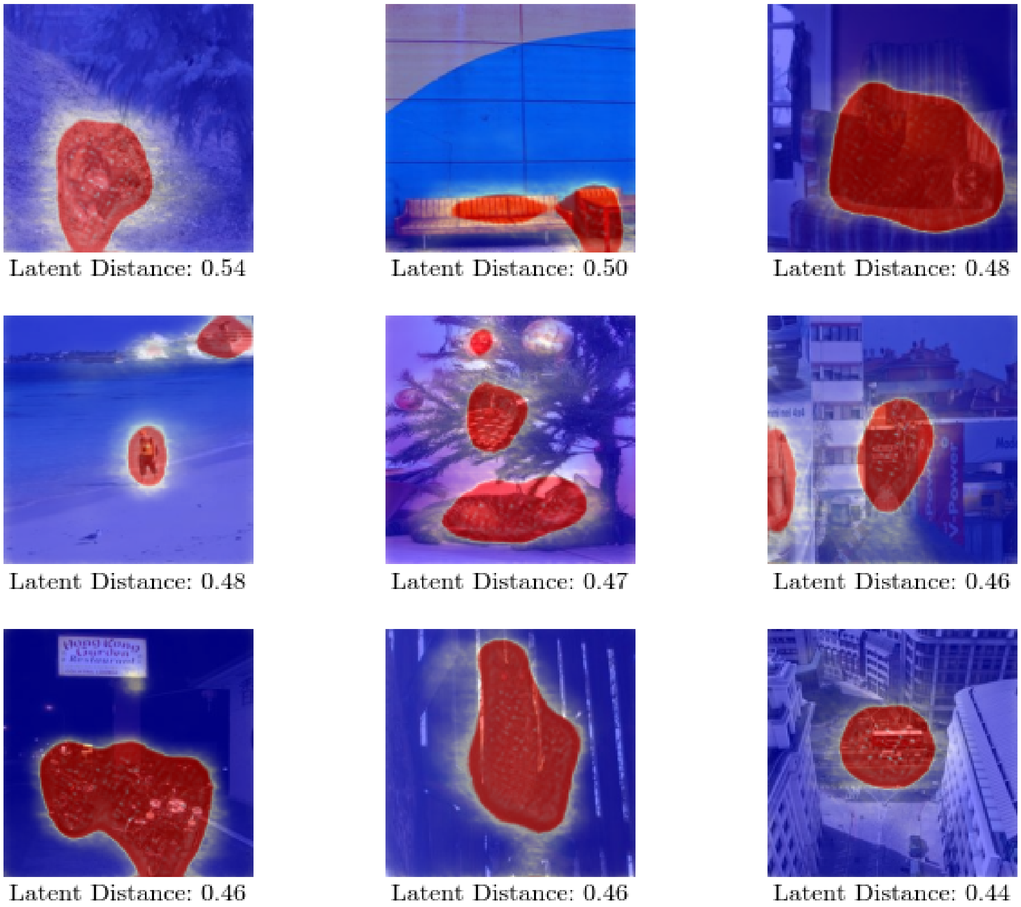
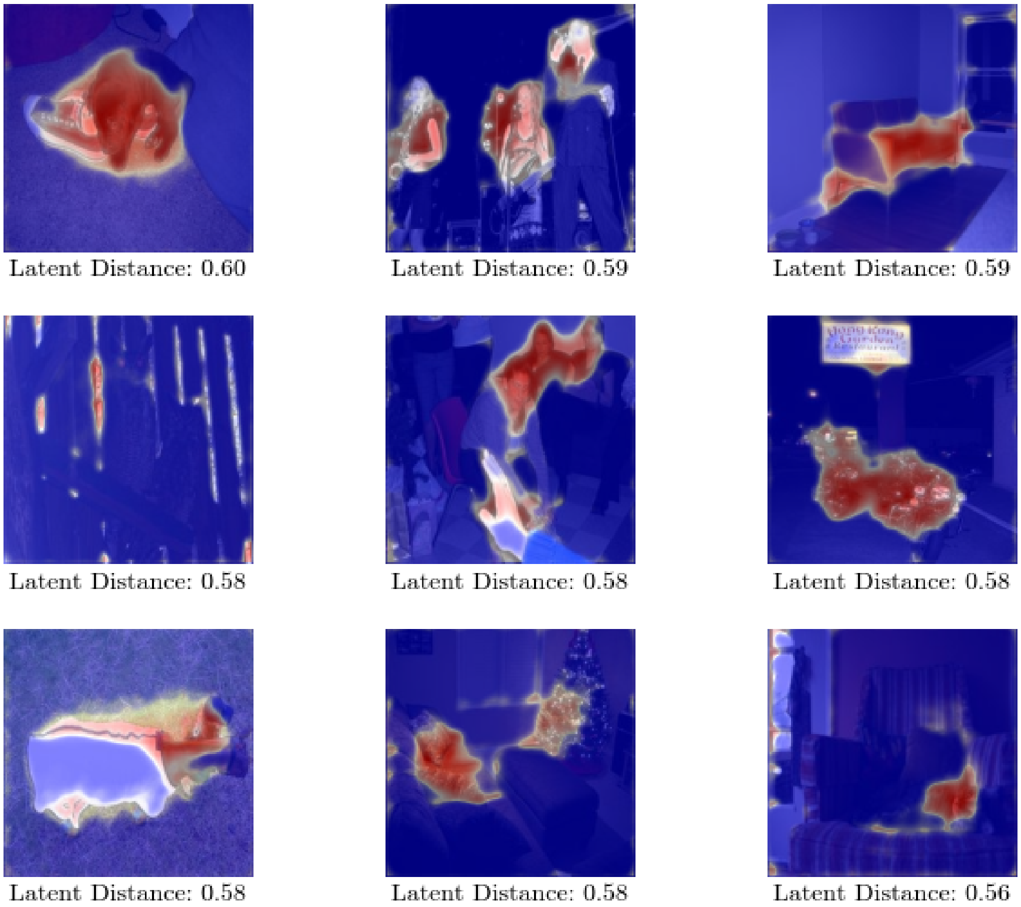


Figure 11. The nine images from the VOC test set for which the masked image had the largest distance to the original image in latent space. The distance was measured using cosine similarity. The masks were produced by NEM-U trained to explain a vision transformer trained using DINO.



Latent Distance: 0.60

Latent Distance: 0.59

Latent Distance: 0.59

Latent Distance: 0.58

Latent Distance: 0.58

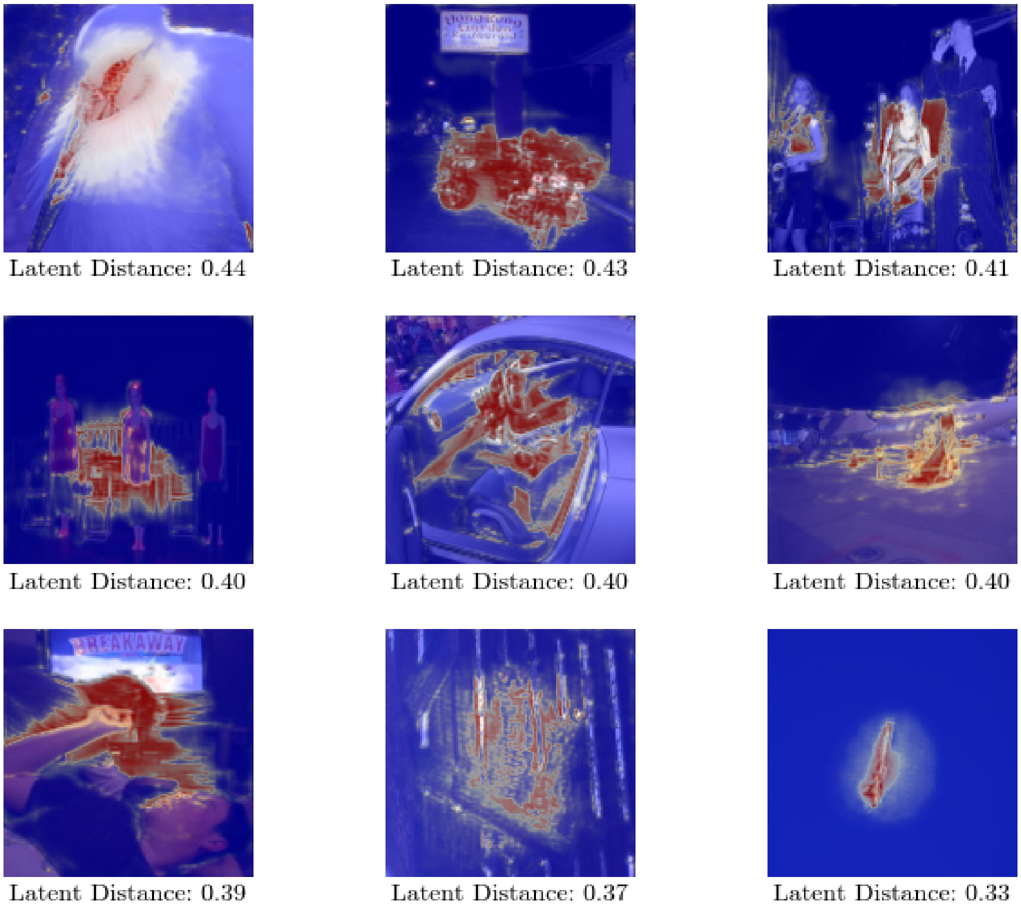
Latent Distance: 0.58

Latent Distance: 0.58

Latent Distance: 0.58

Latent Distance: 0.56

Figure 12. The nine images from the VOC test set for which the masked image had the largest distance to the original image in latent space. The distance was measured using cosine similarity. The masks were produced by NEM-U trained to explain a ResNet50 classifier trained using SwaV.



Latent Distance: 0.44

Latent Distance: 0.43

Latent Distance: 0.41

Latent Distance: 0.40

Latent Distance: 0.40

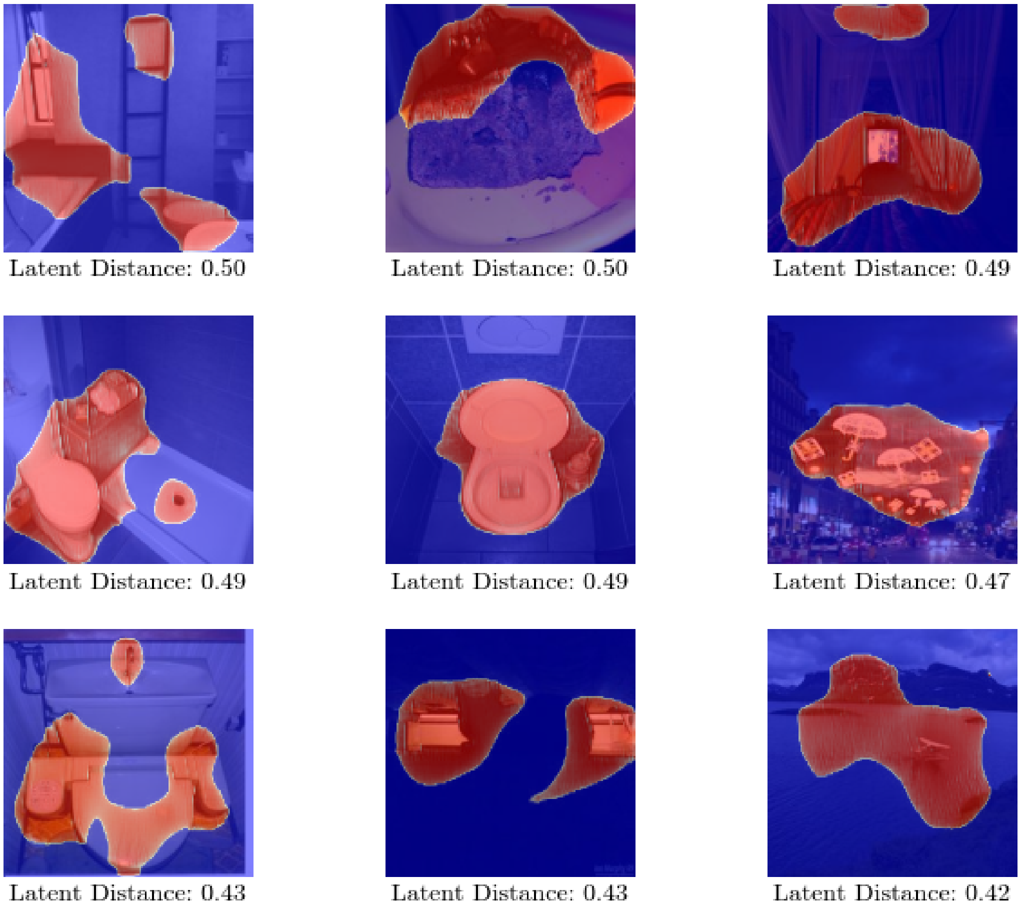
Latent Distance: 0.40

Latent Distance: 0.39

Latent Distance: 0.37

Latent Distance: 0.33

Figure 13. The nine images from the VOC test set for which the masked image had the largest distance to the original image in latent space. The distance was measured using cosine similarity. The masks were produced by NEM-U trained to explain a ResNet50 classifier trained using SimCLR.



Latent Distance: 0.50

Latent Distance: 0.50

Latent Distance: 0.49

Latent Distance: 0.49

Latent Distance: 0.49

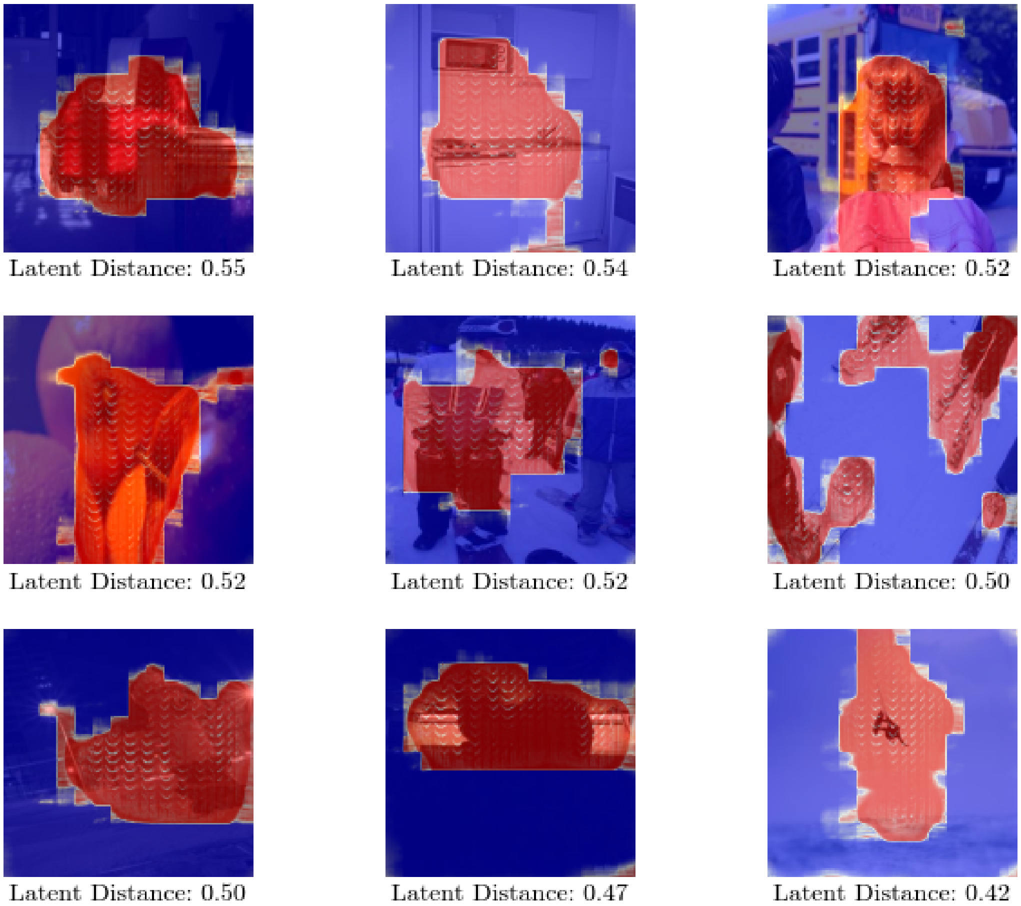
Latent Distance: 0.47

Latent Distance: 0.43

Latent Distance: 0.43

Latent Distance: 0.42

Figure 14. The nine images from the COCO validation set for which the masked image had the largest distance to the original image in latent space. The distance was measured using cosine similarity. The masks were produced by NEM-U trained to explain a ResNet50 classifier trained using a supervised classification task.



Latent Distance: 0.55

Latent Distance: 0.54

Latent Distance: 0.52

Latent Distance: 0.52

Latent Distance: 0.52

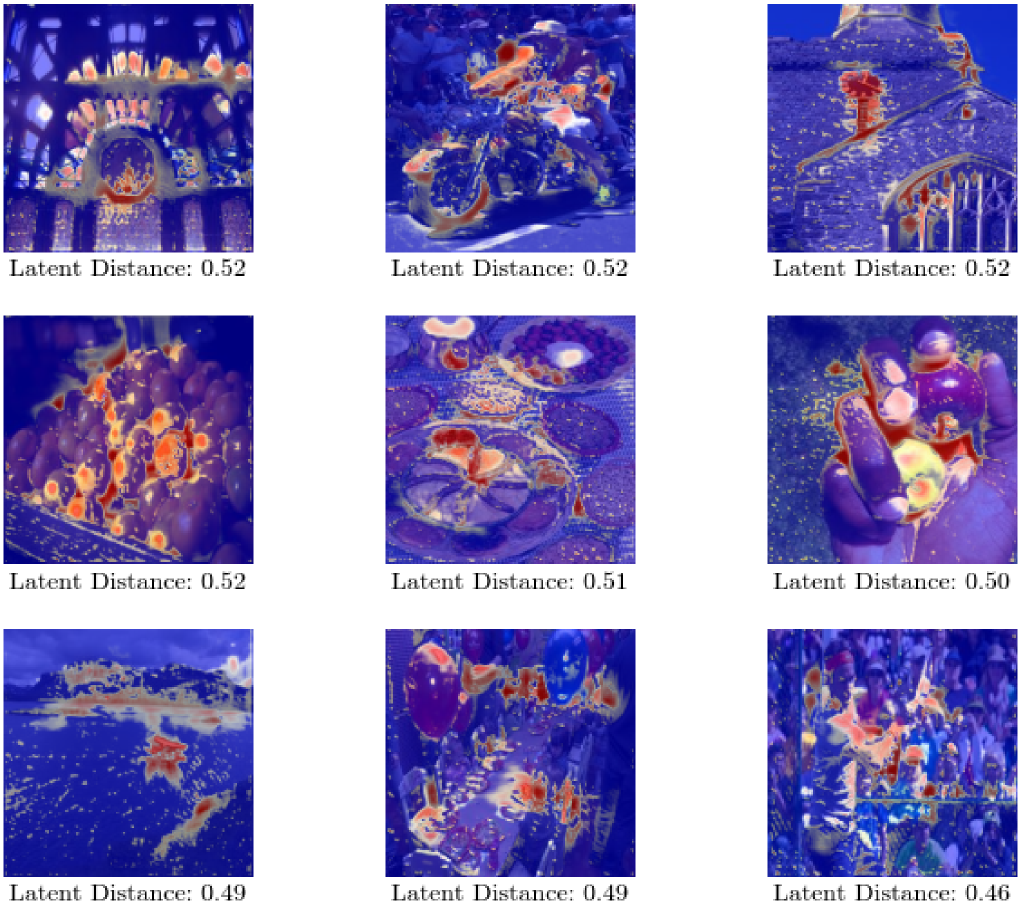
Latent Distance: 0.50

Latent Distance: 0.50

Latent Distance: 0.47

Latent Distance: 0.42

Figure 15. The nine images from the COCO validation set for which the masked image had the largest distance to the original image in latent space. The distance was measured using cosine similarity. The masks were produced by NEM-U trained to explain a vision transformer trained using DINO.



Latent Distance: 0.52

Latent Distance: 0.52

Latent Distance: 0.52

Latent Distance: 0.52

Latent Distance: 0.51

Latent Distance: 0.50

Latent Distance: 0.49

Latent Distance: 0.49

Latent Distance: 0.46

Figure 16. The nine images from the COCO validation set for which the masked image had the largest distance to the original image in latent space. The distance was measured using cosine similarity. The masks were produced by NEM-U trained to explain a ResNet50 classifier trained using SWaV.

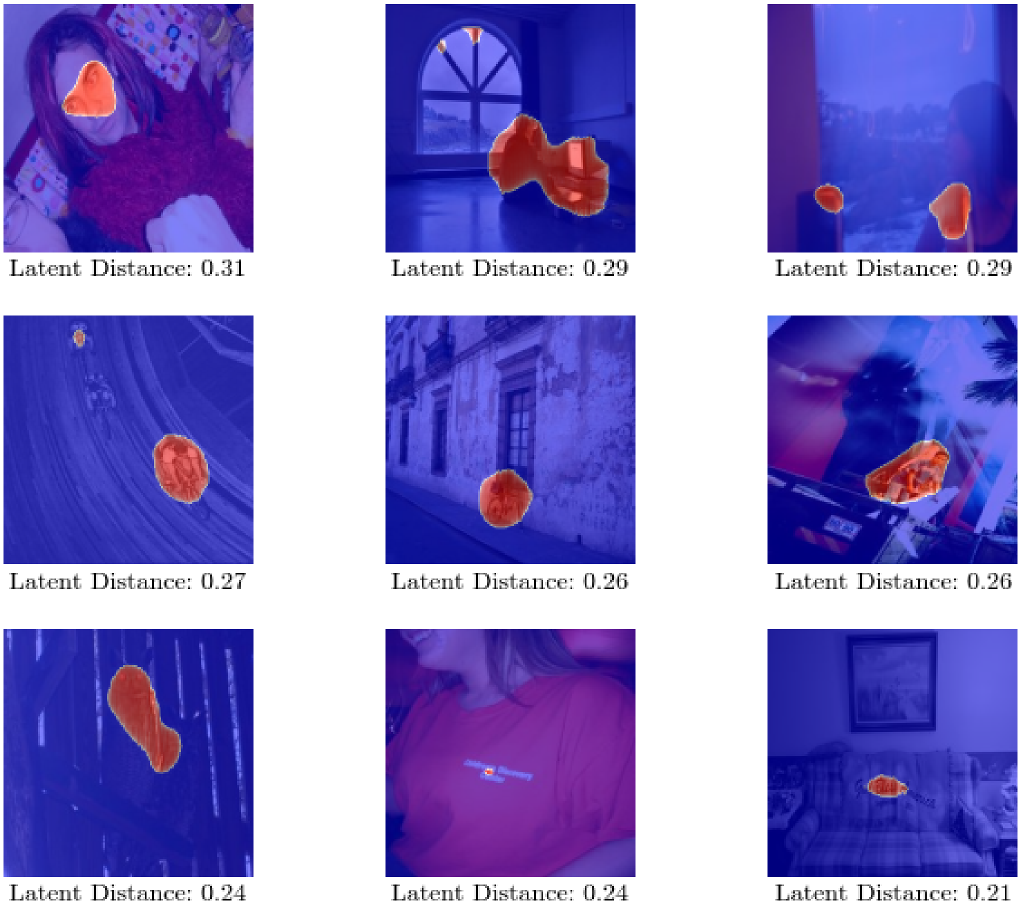


Figure 17. The nine images from the COCO validation set for which the masked image had the largest distance to the original image in latent space. The distance was measured using cosine similarity. The masks were produced by NEM-U trained to explain a ResNet50 classifier trained using SimCLR.