This CVPR 2020 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Towards Visually Explaining Variational Autoencoders

Wenqian Liu[1*], Runze Li[2*], Meng Zheng[3], Srikrishna Karanam[4], Ziyan Wu[4],
Bir Bhanu[2], Richard J. Radke[3], and Octavia Camps[1]
[1]Northeastern University, Boston MA [2]University of California Riverside, Riverside CA
[3]Rensselaer Polytechnic Institute, Troy NY [4]United Imaging Intelligence, Cambridge MA
liu.wenqi@husky.neu.edu,rli047@ucr.edu,zhengm3@rpi.edu,{first.last}@united-imaging.com
bhanu@cris.ucr.edu,rjradke@ecse.rpi.edu,camps@ece.neu.edu

Abstract

Recent advances in convolutional neural network (CNN) model interpretability have led to impressive progress in visualizing and understanding model predictions. In particular, gradient-based visual attention methods have driven much recent effort in using visual attention maps as a means for visual explanations. A key problem, however, is these methods are designed for classification and categorization tasks, and their extension to explaining generative models, e.g.*, variational autoencoders (VAE) is not trivial. In this work, we take a step towards bridging this crucial gap, proposing the first technique to visually explain VAEs by means of gradient-based attention. We present methods to generate visual attention from the learned latent space, and also demonstrate such attention explanations serve more than just explaining VAE predictions. We show how these attention maps can be used to localize anomalies in images, demonstrating state-of-the-art performance on the MVTec-AD dataset. We also show how they can be infused into model training, helping bootstrap the VAE into learning improved latent space disentanglement, demonstrated on the Dsprites dataset.*

1. Introduction

Dramatic progress in computer vision, driven by deep learning [22, 13, 15], has led to widespread adoption of the associated algorithms in real-world tasks, including healthcare, robotics, and autonomous driving [17, 50, 23] among others. Applications in many such safety-critical and consumer-focusing areas demand a clear understanding of the reasoning behind an algorithm's predictions, in addition certainly to robustness and performance guarantees. Consequently, there has been substantial recent interest in devising ways to understand and explain the underlying *why* driving the output *what*.

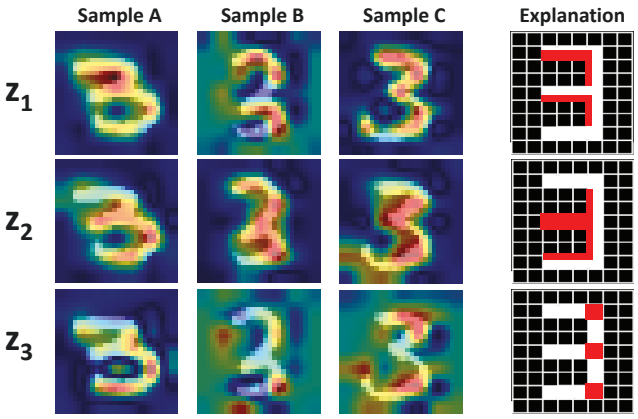

Figure 1. We propose to visually explain variational autoencoders. Each element in the latent vector (here $z_1 - z_3$) can be explained separately with our attention maps, visualizing consistent explanations across different samples.

Following the work of Zeiler and Fergus [40], much recent effort has been expended in developing ways to visualize feature activations in convolutional neural networks (CNNs). One line of work that has seen increasing adoption involves network attention [47, 33], typically visualized by means of attention maps that highlight feature regions considered (by the trained model) to be important for satisfying the training criterion. Given a trained CNN model, these techniques are able to generate attention maps that visualize where a certain object, *e.g.*, a *cat*, is in the image, helping explain why this image is classified as belonging to the *cat* category. Some extensions [24, 36] provide ways to use the generated attention maps as part of trainable constraints that are enforced during model training, showing improved model generalizability as well as visual explainability. While Zheng *et al.* [45] used a classification module to show how one can generate a pair of such attention maps to explain why two images of people are similar/dissimilar,

*Wenqian Liu and Runze Li contributed equally to this work.

all these techniques, by design, need to perform classification to guide model explainability, limiting their use to object categorization problems.

Starting from such classification model explainability, one would naturally like to explain a wider variety of neural network models and architectures. For instance, there has been an explosion in the use of generative models following the work of Kingma and Welling [21] and Goodfellow *et al.* [12], and subsequent successful applications in a variety of tasks [16, 26, 37, 39]. While progress in algorithmic generative modeling has been swift [38, 18, 30], explaining such generative algorithms is still a relatively unexplored field of study. There are certainly some ongoing efforts in *using* the concept of visual attention in generative models [35, 2, 41], but the focus of these methods is to use attention as an auxiliary information source for the particular task of interest, and not visually explain the generative model itself.

In this work, we take a step towards bridging this crucial gap, developing new techniques to visually explain variational autoencoders (VAE) [21]. Note that while we use VAEs as an instantiation of generative models in our work, some of the ideas we discuss are not limited to VAEs and can certainly be extended to GANs [12] as well. Our intuition is that the latent space of a trained VAE encapsulates key properties of the VAE and that generating explanations conditioned on the latent space will help explain the reasoning for any downstream model predictions. Given a trained VAE, we present new ways to generate visual attention maps from the latent space by means of gradient-based attention. Specifically, given the learned Gaussian distribution, we use the reparameterization trick [21] to sample a latent code. We then backpropagate the activations in each dimension of the latent code to a convolutional feature layer in the model and aggregate all the resulting gradients to generate the attention maps. While these visual attention maps serve as means to explain the VAE, we can do much more than just that. A classical application of a VAE is in anomaly localization, where the intuition is that any input data that is not from the standard Gaussian distribution used to train the VAE should be anomalous in the inferred latent space. Given this inference, we can now generate attention maps helping visually explain *why* this particular input is anomalous. We then also go a step further, presenting ways in which to use these explanations as cues to precisely localize where the anomaly is in the image. We conduct extensive experiments on the recently proposed MVTec anomaly detection dataset and present state-of-the-art anomaly localization results with just the standard VAE without any bells and whistles.

Latent space disentanglement is another important area of study with VAEs and has seen much recent progress [14, 19, 46]. With our visual attention explanations conditioned on the learned latent space, our intuition that using these attention maps as part of trainable constraints will lead to improved latent space disentanglement. To this end, we present a new learning objective we call attention disentanglement loss and show how one can train existing VAE models with this new loss. We demonstrate its impact in learning a disentangled embedding by means of experiments on the Dsprites dataset [29].

To summarize, our key contributions are:

- We take a step towards solving the relatively unexplored problem of visually explaining generative models, presenting new methods to generate visual attention maps conditioned on the latent space of a variational autoencoder. Furthermore, we show how our visual attention maps can be put to multipurpose use.
- We present new ways to localize anomalies in images by using our attention maps as cues, demonstrating state-of-the-art localization performance on the MVTec-AD dataset [3].
- We present a new learning objective called the attention disentanglement loss, showing how one incorporate it into standard VAE models, and demonstrate improved disentanglement performance on the Dsprites dataset [29].

2. Related Work

CNN Visual Explanations. Much recent effort has been expended in explaining CNNs as they have come to dominate performance on most vision tasks. Some widely adopted methods that attempt to visualize intermediate CNN feature layers included the work of Zeiler and Fergus [40] and Mahendran and Vedaldi [27], where methods to understand the activity within the layers of convolutional nets were presented. Some more recent extensions of this line of work include visual-attention-based approaches [47, 11, 34, 6], most of which can be categorized into either gradient-based methods or response-based methods. Gradient-based method such as GradCAM [34] compute and visualize gradients backpropagated from the decision unit to a feature convolutional layer. On the other hand, response-based approaches [42, 47, 11] typically add additional trainable units to the original CNN architecture to compute the attention maps. In both cases, the goal is to localize attentive and informative image regions that contribute the most to the model prediction. However, these methods and their extensions [11, 24, 36], while able to explain classification/categorization models, cannot be trivially extended to explaining deep generative models such as VAEs. In this work, we present methods, using the philosophy of gradient-based network attention, to compute and visualize attention maps directly from the learned latent embedding of the VAE. Furthermore, we make the resulting

attention maps end-to-end trainable and show how such a change can result in improved latent space disentanglement.

Anomaly Detection. Unsupervised learning for anomaly detection [1] still remains challenging. Most recent work in anomaly detection is based on either classification-based [31, 5] or reconstruction-based approaches. Classification-based approaches aim to progressively learn representative one-class decision boundaries like hyperplanes [5] or hyperspheres [31] around the normal-class input distribution to tell outliers/anomalies apart. However, it was also shown [4] that these methods have difficulty dealing with high-dimensional data. Reconstruction-based models, on the other hand, assume input data that are anomalous cannot be reconstructed well by a model that is trained only with normal input data. This principle has been used by several methods based on the traditional PCA [20], sparse representation [44], and more recently deep autoencoders [49, 48]. In this work, we take a different approach to tackling this problem. We use the attention maps generated by our proposed VAE visual explanation generation method as cues to localize anomalies. Our intuition is that representations of anomalous data should be reflected in latent embedding as being anomalous, and that generating input visual explanations from such an embedding gives us the information we need to localize the particular anomaly.

VAE Disentanglement. Much effort has been expended in understanding latent space disentanglement for generative models. Early work of Schmidhuber *et al.* [32] proposed a principle to disentangle latent representations by minimizing the predictability of one latent dimension given other dimensions. Desjardins et. al [10] generalized an approach based on restricted Boltzmann machines to factor the latent variables. Chen et. al extended GAN [12] framework to design the InfoGAN [8] to maximise the mutual information between a subset of latent variables and the observation. Some of the more recent unsupervised methods for disentanglement include β-VAE [14] which attempted to explore independent latent factors of variation in observed data. While still a popular unsupervised framework, β-VAE sacrificed reconstruction quality for obtaining better disentanglement. Chen et. al [7] extended β-VAE to β-TCVAE by introducing a total correlation-based objective, whereas Mathieu *et al.* [28] explored decomposition of the latent representation into two factors for disentanglement, and Kim *et al.* [19] proposed FactorVAE that encouraged the distribution of representations to be factorial and independent across the dimensions. While these methods focus on factorizing the latent representations provided by each individual latent neuron, we take a different approach. We enforce learning a disentangled space by formulating disentanglement constraints based on our proposed visual explanations, *i.e.*, visual attention maps. To this end, we propose a new attention disentanglement learning objective that we quantitatively show provides superior performance when compared to existing work.

3. Approach

In this section, we present our method to generate explanations for a VAE by means of gradient-based attention. We first begin with a brief review of VAEs in Sections 3.1 followed by our proposed method to generate VAE attention. We discuss our framework for localizing anomalies in images with these attention maps and conduct extensive experiments on the MVTec-AD anomaly detection dataset [3], establishing state-of-the-art anomaly localization performance. Next, we show how our generated attention visualizations can assist in learning a disentangled latent space by optimizing our new attention disentanglement loss. Here, we conduct experiments on the Dsprites [29] dataset and quantitatively demonstrate improved disentanglement performance when compared to existing approaches.

3.1. One-Class Variational Autoencoder

A vanilla VAE is essentially an autoencoder that is trained with the standard autoencoder reconstruction objective between the input and decoded/reconstructed data, as well as a variational objective term attempts to learn a standard normal latent space distribution. The variational objective is typically implemented with Kullback-Leibler distribution metric computed between the latent space distribution and the standard Gaussian. Given input data $\mathbf{x}$, the conditional distribution $q(\mathbf{z}|\mathbf{x})$ of the encoder, the standard Gaussian distribution $p(\mathbf{z})$, and the reconstructed data $\hat{\mathbf{x}}$, the vanilla VAE optimizes:

$$L = L_r(\mathbf{x}, \hat{\mathbf{x}}) + L_{\text{KL}}(q(\mathbf{z}|\mathbf{x}), p(\mathbf{z})) \tag{1}$$

where L_{KL} is the Kullback-Leibler divergence term and L_r is the reconstruction term, which is typically a mean-squared error between $\mathbf{x}$ and $\hat{\mathbf{x}}$.

3.2. Generating VAE Attention

We propose a new technique to generate VAE visual attention by means of gradient-based attention computation. Our proposed approach is substantially different from existing work [34, 47, 45] that computes attention maps by back-propagating the score from a classification model. On the other hand, we are not restricted by such requirements and develop an attention mechanism directly using the learned latent space, thereby not needing an additional classification module. As illustrated in Figure 2 and discussed below, we compute a score from the latent space, which is then used to calculate gradients and obtain the attention map.

Specifically, given the posterior distribution $q(\mathbf{z}|\mathbf{x})$ inferred by the trained VAE for a data sample $\mathbf{x}$, we use the

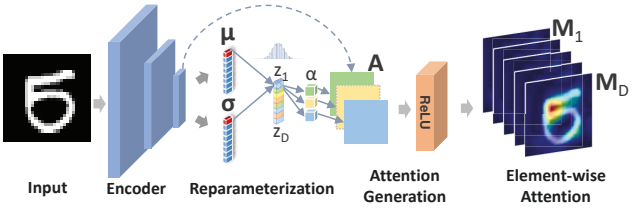

Figure 2. Element-wise attention generation with a VAE.

reparameterization trick to obtain a latent vector $\mathbf{z}$. For each element z_i, we backpropagate gradients to the last convolutional feature maps $\mathbf{A} \in \mathbb{R}^{n \times h \times w}$, giving the attention map $\mathbf{M}^i$ corresponding to z_i. Specifically, $\mathbf{M}^i$ is computed as the linear combination:

$$\mathbf{M}^i = \text{ReLU}(\sum_{k=1}^{n} \alpha_k \mathbf{A}_k) \tag{2}$$

where the scalar $\alpha_k = \text{GAP}(\frac{\partial z_i}{\partial \mathbf{A}_k})$ and $\mathbf{A}_k$ is the k^{th} feature channel ($k = 1, \ldots, n$) of the feature maps $\mathbf{A}$. Note $\frac{\partial z_i}{\partial \mathbf{A}_k}$ is a matrix and so we use the global average pooling (GAP) operation to get the scalar α_k. Specifically, this is:

$$\alpha_k = \frac{1}{T} \sum_{p=1}^{h} \sum_{q=1}^{w} (\frac{\partial z_i}{\partial A_k^{pq}}) \tag{3}$$

where $T = h \times w$ and A_k^{pq} is the pixel value at location (p, q) of the $h \times w$ matrix $\mathbf{A}_k$. We now repeat this for all elements $z_1, z_2, \ldots, z_D$ of the $D-$dimensional latent space, giving $\mathbf{M}^1, \ldots, \mathbf{M}^D$ (see Figure 2). An example of what each $\mathbf{M}^i$ represents is shown in Figure 1, where we see consistent high-response regions for each latent dimension across multiple data samples. While the above procedure gives one attention map per latent dimension, one can obtain a single overall attention map using any matrix aggregation scheme, *e.g.*, the mean, in which case the overall attention map is $\mathbf{M} = \frac{1}{D} \sum_i^D \mathbf{M}^i$.

3.3. Generating Anomaly Attention Explanations

We now discuss how our gradient-based attention generation mechanism can be used to localize anomaly regions given a trained one-class VAE. Inference with such a one-class VAE with data it was trained for, *i.e.*, normal data (digit "1" for instance), should ideally result in the learned latent space representing the standard normal distribution. Consequently, given a testing sample from a different class (abnormal data, digit "5" for instance), the latent representation inferred by the learned encoder should have a large difference when compared to the learned normal distribution. This intuition can be captured in many ways. A straightforward approach (which we use to show results next) is to take the inferred mean vector and generate the resulting attention map. Specifically, we compute the sum of all elements in the mean vector, giving a score s, which we backpropagate to compute the anomaly attention $\mathbf{M}$ (as in Equation 2). An alternative approach can be using the normal difference distribution. Given all normal images used to train the VAE, we can infer the overall μ^x and σ^x representing the distribution of embeddings of all the normal images $\mathbf{x} \in \mathbf{X}$. Now, given the μ_i^y and σ_i^y for each latent variable z_i inferred for an abnormal sample $\mathbf{y}$, we can define the normal difference distribution as:

$$P_{q(z_i|x)-q(z_i|y)}(u) = \frac{e^{-[u-(\mu_i^x-\mu_i^y)]^2/[2((\sigma_i^x)^2+(\sigma_i^y)^2)]}}{\sqrt{2\pi((\sigma_i^x)^2+(\sigma_i^y)^2)}} \tag{4}$$

for each latent variable z_i. Given a latent code $\mathbf{z}$ sampled from $P_{q(z_i|X)-q(z_i|Y)}$, one can follow the procedure described above to compute the anomaly attention map $\mathbf{M}$. This is visually summarized in Figure 3.

3.3.1 Results

In this section, we evaluate our proposed method to generate visual explanations as well as perform anomaly localization with VAEs.

Metrics: We adopt the commonly used the area under the receiver operating characteristic curve (ROC AUC) for all quantitative performance evaluation. We define true positive rate (TPR) as the percentage of pixels that are correctly classified as anomalous across the whole testing class, whereas the false positive rate (FPR) the percentage of pixels that are wrongly classified as anomalous. In addition, we also compute the best intersection-over-union (IOU) score by searching for the best threshold based on our ROC curve. Note that we first begin with qualitative (visual) evaluation on the MNIST and UCSD datasets, and then proceed to a more thorough quantitative evaluation on the MVTec-AD dataset.

MNIST. We start by qualitatively evaluating our visual attention maps on the MNIST dataset [9]. Using training images from one single digit class, we train our one-class VAE model, which will be used to test on all the digit numbers' testing images. We reshape all the training and testing images to resolution of 28×28 pixels.

In Figure 4 (top), we show results with a model trained on the digit "1" (normal class) and test on all other digits (each of which becomes an abnormal class). For each test image, we infer the latent vector using our trained encoder and generate the attention map. As can be observed in the results, the attention maps computed with the proposed method is intuitively satisfying. For instance, let us consider the attention maps generated with digit "7" as the test image. Our intuition tells us that a key difference between the "1" and the "7" is the top-horizontal bar in "7", and our generated attention map indeed highlights this region. Similarly, the differences between an image of the

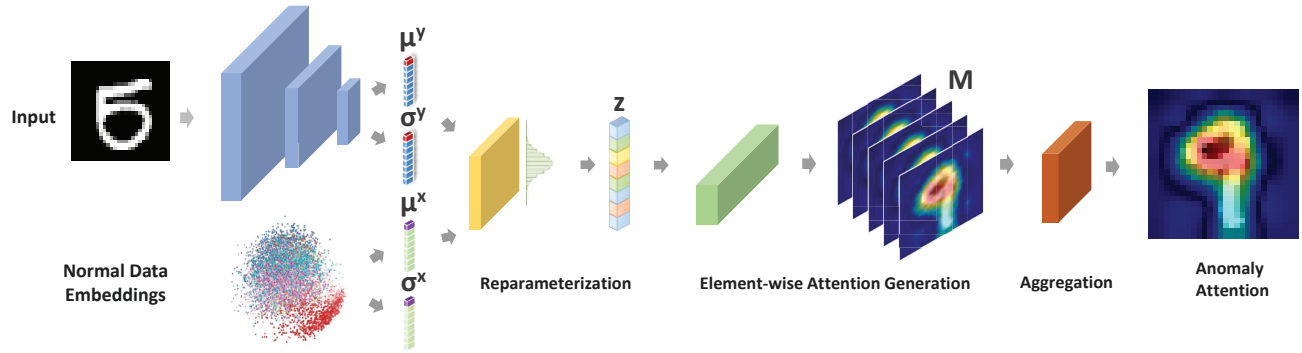

Figure 3. Attention generation with a one-class VAE.

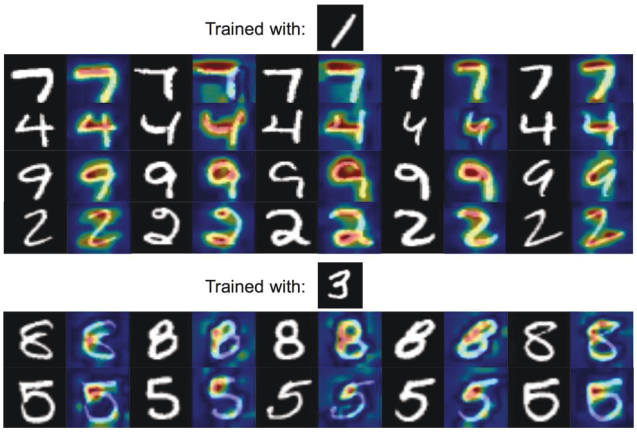

Figure 4. Anomaly localization results from the MNIST dataset.

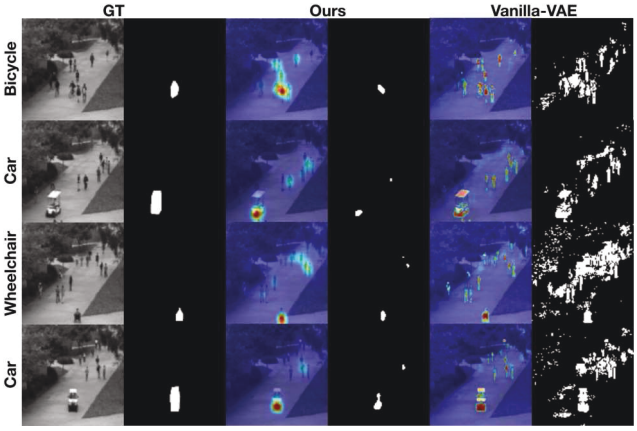

Figure 5. Qualitative results from UCSD Ped1 dataset. L-R: Original test image, ground-truth masks, our anomaly attention localization maps, and difference between input and the VAE's reconstruction . The anomalies in these samples are moving cars, bicycle, and wheelchair.

digit "2" and the "1" are the horizontal base and the top-round regions in the "2". From the generated attention maps for "2", we notice that we are indeed able to capture these differences, highlighting the top and bottom regions in the images for "2". We also show testing results with other digits (*e.g.*, "4","9") as well as with a model trained on digit "3" and tested on the other digits in the same figure. We note similar observations can be made from these results as well, suggesting that our proposed attention generation mechanism is indeed able to highlight anomalous regions, thereby capturing the features in the underlying latent space that cause a certain data sample to be abnormal.

UCSD Ped1 Dataset: We next test our proposed method on the UCSD Ped 1[25] pedestrian video dataset, where the videos were captured with a stationary camera to monitor a pedestrian walkway. This dataset includes 34 training sequences and 36 testing sequences, with about 5500 "normal" frames and 3400 "abnormal" frames. We resize the data to 100×100 pixels for training and testing.

We first qualitatively evaluate the performance of our proposed attention generation method in localizing anomalies. As we can see from Figure 5 (where the corresponding anomaly of interest is annotated on the left, *e.g.*, *bicycle*, *Car etc.*), our anomaly localization technique with attention maps performs substantially better than simply computing the difference between the input and its reconstruction (this result is annotated as *Vanilla-VAE* in the figure). We note more precise localization of the high-response regions in our generated attention maps, and these high-response regions indeed correspond to anomalies in these images.

We next conduct a simple ablation study using the pixel-level segmentation AUROC score against the baseline method of difference between input data and the reconstruction. We test our proposed attention generation mechanism with varying levels of spatial resolution by backpropagating to each of the encoder's convolutional layers: 50×50, 25×25, and 12×12. The results are shown in Table 1 where we see our proposed mechanism gives better performance than the baseline technique.

MVTec-AD Dataset: We consider the recently released comprehensive anomaly detection dataset: MVTec Anomaly Detection (MVTec AD) [3] that provides multi-object, multi-defect natural images and pixel-level ground

	Vanilla-VAE	Ours(Conv1)	Ours(Conv2)	Ours(Conv3)
AUROC	0.86	0.89	**0.92**	0.91

Table 1. Results on UCSD Ped1 using pixel-level segmentation AUROC score. We compare results obtained using our anomaly attention generated with different target network layers to reconstruction-based anomaly localization using Vanilla-VAE.

truth. This dataset contains 5354 high-resolution color images of different objects/textures, with both normal and defect (abnormal) images provided in the testing set. We resize all images to 256×256 pixels for training and testing. We conduct extensive qualitative and quantitative experiments and summarize results below.

We train a VAE with ResNet18 [13] as our feature encoder and a 32-dimensional latent space. We further use random mirroring and random rotation, as done in the original work [3], to generate an augmented training set. Given a test image, we infer its latent representation $\mathbf{z}$ to generate the anomaly attention map. Given our anomaly attention maps, we generate binary anomaly localization maps using a variety of thresholds on the pixel response values, which is encapsulated in the ROC curve. We then compute and report the area under the ROC curve (ROC AUC) and generate the best IOU number for our method based on FPR and TPR from the ROC curve.

The results are shown in Table 2, where we compare our performance with the techniques evaluated in the benchmark paper of Bergmann *et al.* [3] (note that the baselines here are the same methods as in [3]). From the results, we note that with our anomaly localization approach using the proposed VAE attention, we obtain better results on most of the object categories than the competing methods. It is worth noting here that some of these methods are specifically designed for the anomaly localization task, whereas we train a standard VAE and generate our VAE attention maps for localization. Despite this simplicity, our method achieves competitive performance, demonstrating the potential of such an attention generation technique to be useful for tasks other than just model explanation.

We also show some qualitative results in Figure 6. We show results from six categories - three textures and three objects. For each category, we also show four types of defects provided by the dataset. We show, from the top row to the bottom, the original images, ground truth segmentation masks, and our anomaly attention maps. One can note that our attention maps are able to accurately localize anomalous regions across these various defect categories.

3.4. Attention Disentanglement

In the previous section, we discussed how one can generate visual explanations, by means of gradient-based attention, as well as anomaly attention maps for VAEs. We also discussed and experimentally evaluated using these

	Category	AE (SSIM)	AE (L2)	Ano GAN	CNN Feature Dictionary	ours
Texture	Carpet	**0.87**	0.59	0.54	0.72	0.78
		0.69	0.38	0.34	0.20	0.1
	Grid	**0.94**	0.90	0.58	0.59	0.73
		0.88	0.83	0.04	0.02	0.02
	Leather	0.78	0.75	0.64	0.87	**0.95**
		0.71	0.67	0.34	**0.74**	0.24
	Tile	0.59	0.51	0.50	**0.93**	0.80
		0.04	**0.23**	0.08	0.14	**0.23**
	Wood	0.73	0.73	0.62	**0.91**	0.77
		0.36	0.29	0.14	**0.47**	0.14
Objects	Bottle	**0.93**	0.86	0.86	0.78	0.87
		0.15	0.22	0.05	0.07	**0.27**
	Cable	0.82	0.86	0.78	0.79	**0.90**
		0.01	0.05	0.01	0.13	**0.18**
	Capsule	**0.94**	0.88	0.84	0.84	0.74
		0.09	**0.11**	0.04	0.00	**0.11**
	Hazelnut	0.97	0.95	0.87	0.72	**0.98**
		0.00	0.41	0.02	0.00	**0.44**
	Metal Nut	0.89	0.86	0.76	0.82	**0.94**
		0.01	0.26	0.00	0.13	**0.49**
	Pill	**0.91**	0.85	0.87	0.68	0.83
		0.07	**0.25**	0.17	0.00	0.18
	Screw	0.96	0.96	0.80	0.87	**0.97**
		0.03	**0.34**	0.01	0.00	0.17
	Toothbrush	0.92	0.93	0.90	0.77	**0.94**
		0.08	**0.51**	0.07	0.00	0.14
	Transistor	0.90	0.86	0.80	0.66	**0.93**
		0.01	0.22	0.08	0.03	**0.30**
	Zipper	**0.88**	0.77	0.78	0.76	0.78
		0.10	**0.13**	0.01	0.00	0.06

Table 2. Quantitative results for pixel level segmentation on 15 categories from MVTec-AD dataset. For each category, we report the area under ROC AUC curve on the top row, and best IOU on the bottom row. We adopt comparison scores from [3].

anomaly attention maps for anomaly localization on a variety of datasets. We next discuss another application of our proposed VAE attention: VAE latent space disentanglement. Existing approaches for learning disentangled representations of deep generative models focus on formulating factorised, independent latent distributions so as to learn interpretable data representations. Some examples include β-VAE [14], InfoVAE [43], and FactorVAE [19], among others, all of which attempt to model the latent prior with factorial probability distribution. In this work, we present an alternative technique, based on our proposed VAE attention, called the attention disentanglement loss. We show how it can be integrated with existing baselines, *e.g.*, FactorVAE, and demonstrate the resulting impact by means of qualitative attention maps and quantitatively performance characterization with standard disentanglement metrics.

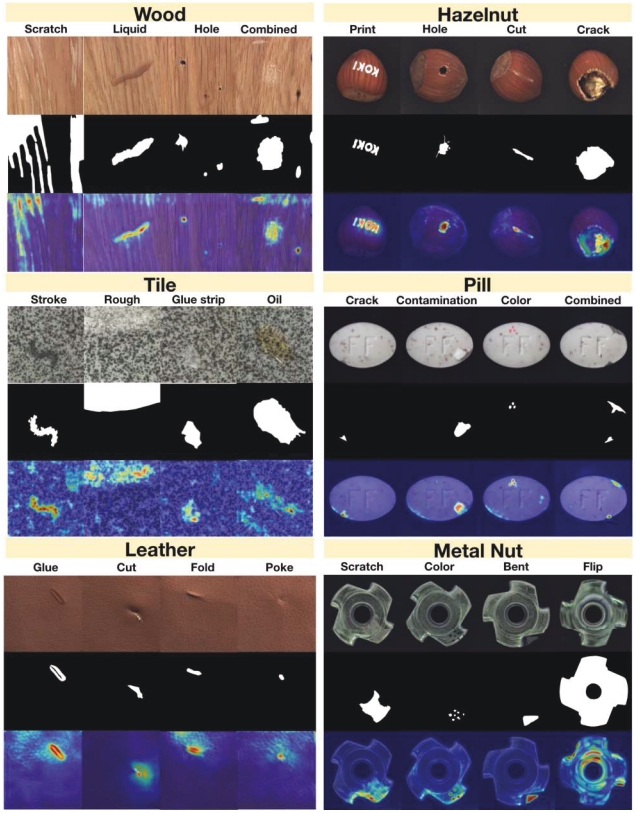

Figure 6. Qualitative results from MVTec-AD. Here, we provide results from: Wood, Tile, Leather, Hazelnut, Pill, and Metal Nut. For each category, we show four different type of defects. As can be seen from the figure, our anomaly attention maps are able to accurately localize anomalies.

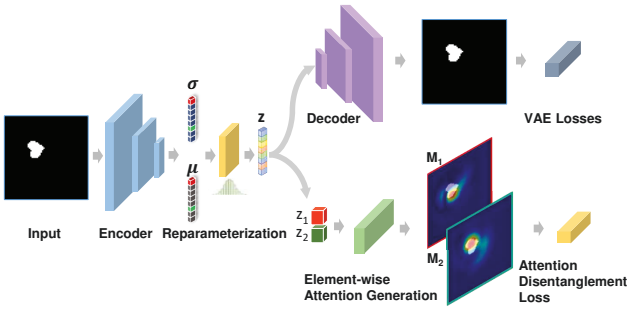

Figure 7. Training a variational autoencoder with the proposed attention disentanglement loss.

3.4.1 Training with Attention Disentanglement

As we showed earlier, our proposed VAE attention, by means of gradient-based attention, generates attention maps that can explain the underlying latent space represented by the trained VAE. We showed how attention maps intuitively represent different regions of normal and abnormal images, directly corresponding to differences in the latent space (since we generate attention from the latent code). Consequently, our intuition is that using these attention maps to further bootstrap the training process of the VAE model should help boost latent space disentanglement. To this end, our big-picture idea is to use these attention maps as trainable constraints to explicitly force the attention computed from the various dimensions in latent space to be as disentangled, or separable as possible. Our hypothesis is that if we are able to achieve this, we will be able to learn an improved disentangled latent space. To realize this objective, we propose a new loss called the *attention disentanglement* loss (L_{AD}) that can be easily integrated with existing VAE-type models (see Figure 7). Note that while we use the FactorVAE [19] for demonstration in this work, the proposed attention disentanglement loss is in no way limited to this model and can be used in conjunction with other models as well (*e.g.*, β-VAE [14]). The proposed L_{AD} takes two attention maps $\mathbf{A}^1$ and $\mathbf{A}^2$ (each computed from a certain dimension in the latent space following Equation 2) as input, and attempts to separate the high-response pixel regions in them as much as possible. This can be mathematically expressed as:

$$L_{AD} = 2 \cdot \frac{\sum_{ij} min(A^1_{ij}, A^2_{ij})}{\sum_{ij} A^1_{ij} + A^2_{ij}} \tag{5}$$

where $\cdot$ is the scalar product operation, and A^1_{ij} and A^2_{ij} are the $(i,j)^{th}$ pixel in the attention maps $\mathbf{A}^1$ and $\mathbf{A}^2$ respectively. The proposed L_{AD} can be directly integrated with the standard FactorVAE training objective L_{FV}, giving us an overall learning objective that can be expressed as:

$$L = L_{\text{FV}} + \lambda L_{\text{AD}} \tag{6}$$

We now train the FactorVAE with our proposed overall learning objective of Equation 6, and evaluate the impact of L_{AD} by comparisons with the baseline FactorVAE trained only with L_{FV}. For this purpose, we use the same evaluation metric discussed in FactorVAE [19].

3.4.2 Results

Data: We use the Dsprites dataset [29] which provides 737,280 binary 64×64 2D shape images.

Quantitative Results: In Figure 8, we compare the best disentanglement performance (plotted against the reconstruction error) of our proposed method (called AD-FactorVAE) with other competing approaches: baseline FactorVAE [19] (training with only L_{FV}) and β-VAE[14]. We note that training with our proposed L_{AD} results in higher disentanglement scores under the same experimental setting, giving a best disentanglement score of around 0.90, whereas baseline FactorVAE ($\gamma = 40$) gives around 0.82, both with a reconstruction error around 40. We also

Figure 8. Reconstruction error plotted against disentanglement metric [19]. The numbers at each point show β and γ values. We want a low reconstruction error and a high disentanglement metric.

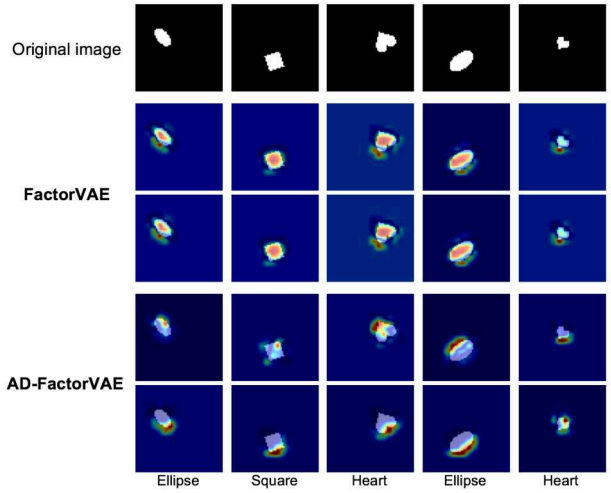

Figure 9. Attention separation on the Dsprites dataset. Top row: the original shape images. Middle two rows: attention maps from FactorVAE. Bottom two rows: attention maps from AD-FactorVAE.

note our proposed method obtains a higher disentanglement score compared to β-VAE (0.73 with $\beta = 4$ as the best result). These results demonstrate the potential of both our proposed VAE attention and L_{AD} in improving the performance of existing methods in the disentanglement literature. These improvements are also reflected in the qualitative results we discuss next.

Qualitative Results: Figure 9 shows some attention maps generated using the baseline FactorVAE and our proposed AD-FactorVAE. The first row shows 5 input images, and the next 4 rows show results with the baseline FactorVAE and our proposed method. Row 2 shows attention maps generated with FactorVAE by backpropagating from the latent dimension with the highest response, whereas row 3 shows attention maps generated by backpropagating from the latent dimension with the next highest response. Rows 4 and 5 show the corresponding attention maps with the proposed AD-FactorVAE. Our intuition and expectation with AD-FactorVAE is that each dimension's attention map will have high responses in different spatial regions of the input. From Figure 9, this is indeed the case, with high-response regions in different areas in the image (rows 4 and 5), whereas we see attention overlap in baseline FactorVAE (rows 2 and 3).

4. Summary and Future Work

We presented new techniques to visually explain variational autoencoders, taking a first step towards explaining deep generative models by means of gradient-based network attention. We showed how one can use the learned latent representation to compute gradients and generate VAE attention maps, without relying on classification-kind of models. We demonstrating applicability of the resulting VAE attention on two tasks: anomaly localization and latent space disentanglement. In anomaly localization, we used the fact that an abnormal input will result in latent variables that do not conform to the standard Gaussian in gradient backpropagation and attention generation. These anomaly attention maps were then used as cues to generate pixel-level binary anomaly masks. In latent space disentanglement, we showed how we can use our VAE attention from each latent dimension to enforce new attention disentanglement learning constraints, resulting in improved attention separability as well as disentanglement performance. Since a VAE can infer a full posterior distribution, with our method, one can obtain a distribution of attention matrices (maps) with repeated sampling. While one way of visualizing this distribution is with the resulting sample mean, generating more generic visual explanations for the full matrix distribution is an interesting topic for future research.

Acknowledgements

This material is based upon work supported in part by NSF grants 1911197, IIS–1814631, ECCS–1808381 and CMMI–1638234, and the U.S. Department of Homeland Security, Science and Technology Directorate, Office of University Programs, under Grant Award 2013-ST-061-ED0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

References

[1] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *ACCV*, 2018.

[2] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *NeurIPS*. 2018.

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*, 2019.

[4] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[5] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

[6] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, 2018.

[7] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *NeurIPS*. 2018.

[8] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*. 2016.

[9] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[10] Guillaume Desjardins, Aaron C. Courville, and Yoshua Bengio. Disentangling factors of variation via generative entangling. *ArXiv*, abs/1210.5474, 2012.

[11] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *CVPR*, 2019.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[17] Dakai Jin, Dazhou Guo, Tsung-Ying Ho, Adam P. Harrison, Jing Xiao, Chen-Kan Tseng, and Le Lu. Accurate esophageal gross tumor volume segmentation in pet/ct using two-stream chained 3d deep network fusion. In *MICCAI*, 2019.

[18] Takuhiro Kaneko, Yoshitaka Ushiku, and Tatsuya Harada. Label-noise robust generative adversarial networks. In *CVPR*, 2019.

[19] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *ICML*, 2018.

[20] Jaechul Kim and Kristen Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *CVPR*, 2009.

[21] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[23] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *CVPR*, 2019.

[24] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Guided attention inference network. *IEEE T-PAMI*, 2019.

[25] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE T-PAMI*, 36(1):18–32, 2013.

[26] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NIPS*. 2017.

[27] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015.

[28] Emile Mathieu, Tom Rainforth, Siddharth Narayanaswamy, and Yee Whye Teh. Disentangling disentanglement. *ArXiv*, abs/1812.02833, 2018.

[29] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[30] Nazanin Mehrasa, Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. A variational auto-encoder model for stochastic point processes. In *CVPR*, 2019.

[31] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *ICML*, 2018.

[32] Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural Comput.*, 4(6):863–879, Nov. 1992.

[33] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[34] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[35] Yichuan Tang, Nitish Srivastava, and Ruslan Salakhutdinov. Learning generative models with visual attention. In *NIPS*, 2013.

[36] Lezi Wang, Ziyan Wu, Srikrishna Karanam, Kuan-Chuan Peng, Rajat Vikram Singh, Bo Liu, and Dimitris N. Metaxas. Sharpen focus: Learning with attention separability and consistency. In *ICCV*, 2019.

[37] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018.

[38] Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *CVPR*, 2019.

[39] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. F-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, 2019.

[40] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[41] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019.

[42] Jianming Zhang, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *IJCV*, 126:1084–1102, 2016.

[43] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *ArXiv*, abs/1706.02262, 2017.

[44] Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. Spatio-temporal autoencoder for video anomaly detection. In *ACM MM*, 2017.

[45] Meng Zheng, Srikrishna Karanam, Ziyan Wu, and Richard J Radke. Re-identification with consistent attentive siamese networks. In *CVPR*, 2019.

[46] Zhilin Zheng and Li Sun. Disentangling latent space for vae by label relevant/irrelevant dimensions. In *CVPR*, 2019.

[47] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

[48] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *KDD*, 2017.

[49] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *ICLR*, 2018.

[50] Yiming Zuo, Weichao Qiu, Lingxi Xie, Fangwei Zhong, Yizhou Wang, and Alan L. Yuille. Craves: Controlling robotic arm with a vision-based economic system. In *CVPR*, 2019.

Supplemental Material
Towards Visually Explaining Variational Autoencoders

Wenqian Liu[1*], Runze Li[2*], Meng Zheng[3], Srikrishna Karanam[4], Ziyan Wu[4],
Bir Bhanu[2], Richard J. Radke[3], and Octavia Camps[1]
[1]Northeastern University, Boston MA [2]University of California Riverside, Riverside CA
[3]Rensselaer Polytechnic Institute, Troy NY [4]United Imaging Intelligence, Cambridge MA
liu.wenqi@husky.neu.edu,rli047@ucr.edu,zhengm3@rpi.edu,{first.last}@united-imaging.com
bhanu@cris.ucr.edu,rjradke@ecse.rpi.edu,camps@ece.neu.edu

1. Element-wise attention

In Figure 1, we show additional results with our element-wise attention generation mechanism (section 3.2 in the main paper).

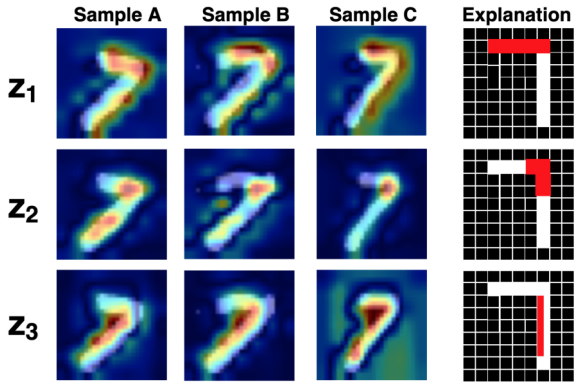

Figure 1. Element-wise attention results. Each element in the latent vector (here $z_1 - z_3$) can be explained separately with our attention maps, visualizing consistent explanations across different samples.

Network	Layer	Output Dimensions
Encoder	Conv 2D, 4×4, 64,2,1	$14 \times 14 \times 64$
	ReLU	$14 \times 14 \times 64$
	Conv 2D, 4×4, 128,2,1	$7 \times 7 \times 128$
	ReLU	$7 \times 7 \times 128$
	Flatten	6272
	Linear	1024
	ReLU	1024
	Linear	32
Decoder	Linear	1024
	ReLU	1024
	Linear	6272
	ReLU	6272
	Unflatten	$7 \times 7 \times 128$
	ReLU	$7 \times 7 \times 128$
	ConvTr 2D, 4×4, 64,2,1	$14 \times 14 \times 64$
	ReLU	$14 \times 14 \times 64$
	ConvTr 2D, 4×4, 1,2,1	$28 \times 28 \times 1$
	Sigmoid	$28 \times 28 \times 1$

Table 1. Architecture details for the one-class VAE on the MNIST dataset. The notation for "Layer" column is as follows: operation, kernel size $h \times w$, number of filter channels, stride, padding. ConvTr 2D denotes the transpose convolution layer.

2. MNIST Dataset

Implementation details and additional results: We resized images in the MNIST dataset to 28×28 pixels. We train our network only on one digit at a time, and then test the trained model on all other digit classes. We set the learning rate to 0.001 and batch size to 128, with a detailed network architecture shown in Table 1. Here, we present additional qualitative results in Figure 2, where we train with digit "1" and test with digit "5" and "6" respectively. These results correspond to Figure 4 in the main paper.

3. UCSD Ped1 Dataset

Implementation details and additional results: We resized each input frame from the UCSD Ped1 dataset to 100×100 pixels, considering each image independently without any temporal knowledge. We set the learning rate to 0.0001 with a batch size of 32 frames for training. A detailed network architecture is shown in Table 2. We show more qualitative results for anomaly localization on UCSD Ped1 in Figure 3. These figures correspond to Figure 5 in the main paper.

*Wenqian Liu and Runze Li contributed equally to this work.

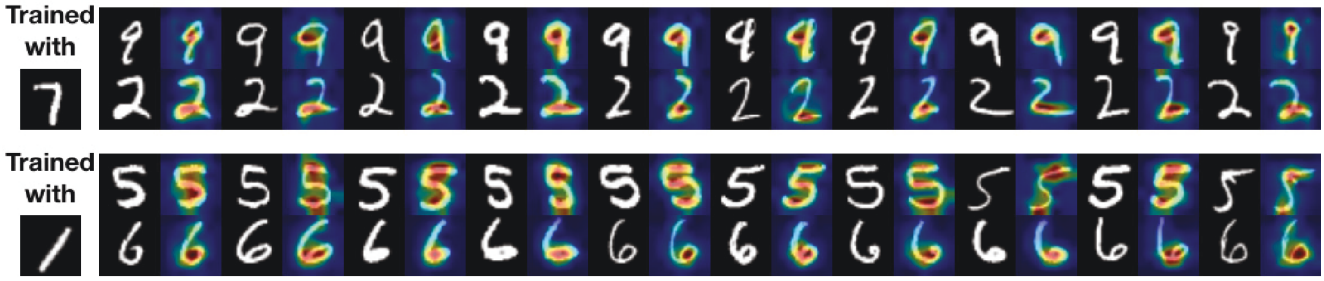

Figure 2. Additional qualitative results from MNIST dataset.

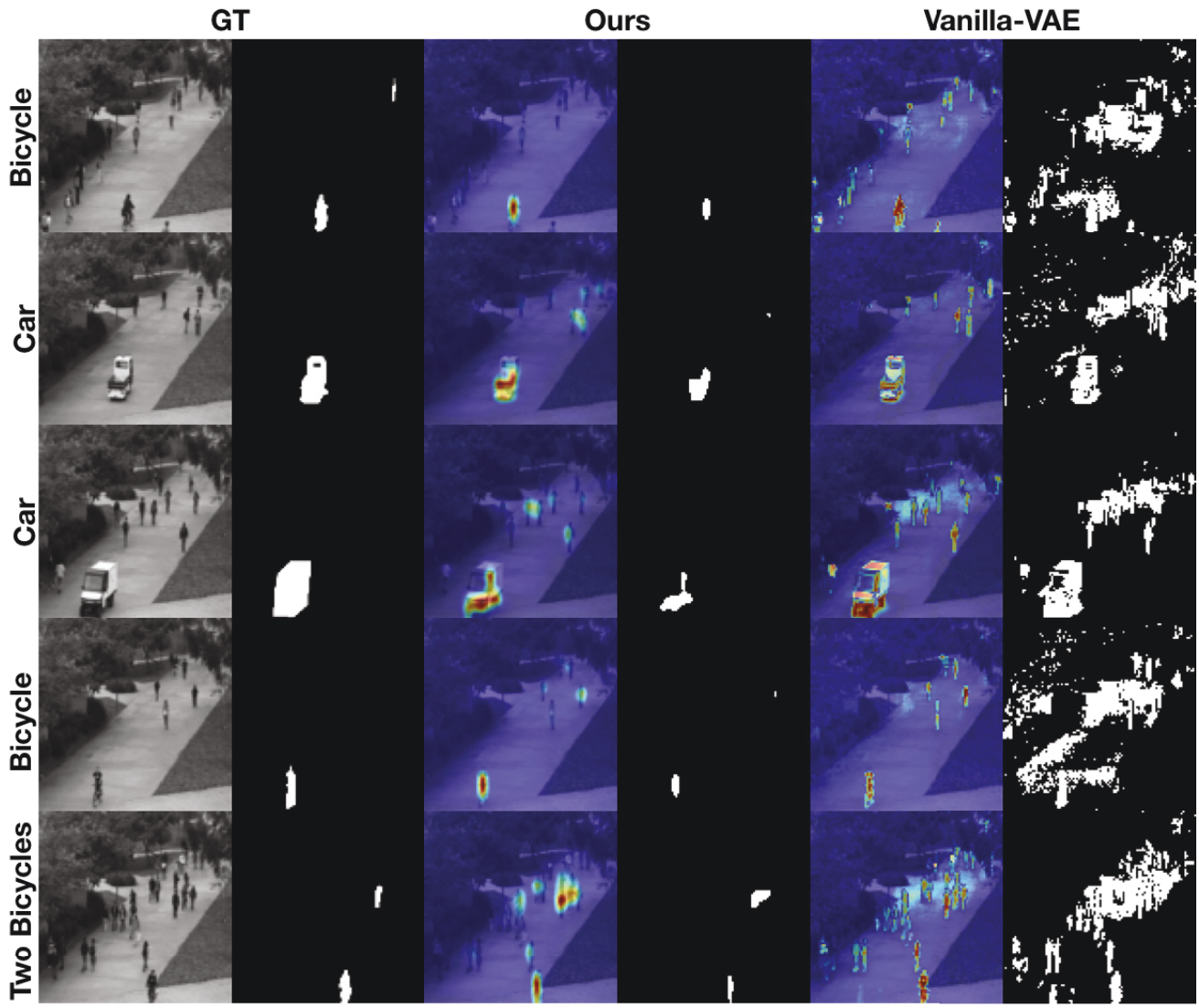

Figure 3. Additional qualitative results from UCSD Ped1 dataset. L-R: ground truth image and mask, our attention maps and masks, and Vanilla-VAE's attention maps and masks. Each row represents a different anomaly situation. Compared to vanilla-VAE, our attention maps and masks localize anomalies much more accurately.

4. MVTec-AD Dataset

Implementation details and additional results: All images are resized to 256×256 pixels. During training, we apply data augmentation with random rotations between $[-30°, +30°]$ and mirroring. We set the learning rate to

Network	Layer	Output Dimensions
Encoder	Conv 2D, 4×4, 64,2,1	$50 \times 50 \times 64$
	ReLU	$50 \times 50 \times 64$
	Conv 2D, 4×4, 128,2,1	$25 \times 25 \times 128$
	ReLU	$25 \times 25 \times 128$
	Conv 2D, 4×4, 256,2,1	$12 \times 12 \times 256$
	ReLU	$12 \times 12 \times 256$
	Flatten	36864
	Linear	1024
	ReLU	1024
	Linear	32
Decoder	Linear	1024
	ReLU	1024
	Linear	36864
	ReLU	36864
	Unflatten	$256 \times 12 \times 12$
	ReLU	$256 \times 12 \times 12$
	ConvTr 2D, 5×5, 128,2,1	$25 \times 25 \times 128$
	ReLU	$25 \times 25 \times 128$
	ConvTr 2D, 4×4, 64,2,1	$50 \times 50 \times 64$
	ReLU	$50 \times 50 \times 64$
	ConvTr 2D, 4×4, 1,2,1	$100 \times 100 \times 1$
	Sigmoid	$100 \times 100 \times 1$

Table 2. Architecture details of the model we use for training and testing on the UCSD Ped1 dataset. The notation for "Layer" column is as follows: operation, kernel size $h \times w$, number of filter channels, stride, padding. ConvTr 2D denotes the transpose convolution layer.

Network	Layer	Output Dimensions
Encoder	Resnet18(w/o last 2 layers)	$8 \times 8 \times 512$
	Linear	1024
	Linear	32
Decoder	Linear	1024
	Linear	$1024 \times 4 \times 4$
	ConvTr 2D, 4×4, 512,2,1	$8 \times 8 \times 512$
	BatchNorm	$8 \times 8 \times 512$
	ReLU	$8 \times 8 \times 512$
	ConvTr 2D, 4×4, 256,2,1	$16 \times 16 \times 256$
	BatchNorm	$16 \times 16 \times 256$
	ReLU	$16 \times 16 \times 256$
	ConvTr 2D, 4×4, 128,2,1	$32 \times 32 \times 128$
	BatchNorm	$32 \times 32 \times 128$
	ReLU	$32 \times 32 \times 128$
	ConvTr 2D, 4×4, 64,2,1	$64 \times 64 \times 64$
	BatchNorm	$64 \times 64 \times 64$
	ReLU	$64 \times 64 \times 64$
	ConvTr 2D, 4×4, 32,2,1	$128 \times 128 \times 32$
	BatchNorm	$128 \times 128 \times 32$
	ReLU	$128 \times 128 \times 32$
	ConvTr 2D, 4×4, 3,2,1	$256 \times 256 \times 3$
	Sigmoid	$256 \times 256 \times 3$

Table 3. Architecture details of the model we use for training and testing on the MVTec-AD dataset. The notation for "Layer" column is as follows: operation, kernel size $h \times w$, number of filter channels, stride, padding. ConvTr 2D denotes transpose convolution layer. We take Resnet18's architecture except its last 2 layers in the encoder, and retrain the whole network on the MVTec-AD dataset.

0.0001 and batch size to 8 for training. A detailed network architecture is shown in Table 3. We show more qualitative results for anomaly localization on the MVTec-AD dataset in Figure 4. These results correspond to Figure 6 in the main paper.

5. Attention Disentanglement

Implementation details and additional results: We resize input images to 64×64 pixels. We replace the last convolutional layer in the standard FactorVAE network with two fully connected layers with an output size of 32. A detailed network architecture is shown is Table 4. We do not perform any hyperparameter search and instead use the same training parameters as FactorVAE, which is our baseline. Figures 5, 6, and 7 show additional attention maps generated with FactorVAE [1] trained with our proposed L_{AD} loss (called AD-FactorVAE in the figure) as well as the baseline FactorVAE. These results correspond to Figure 9 in the main paper. As in the main paper, in each figure, the first row shows the input images, and the next 4 rows show results with the baseline FactorVAE and our proposed method. Row 2 shows attention maps generated with FactorVAE by backpropagating from the latent dimension with the highest response, whereas row 3 shows attention maps generated by backpropagating from the latent dimension with the next highest response. Rows 4 and 5 show the corresponding attention maps with the proposed AD-FactorVAE. From these figures, we can note that for each shape (square, ellipse and heart), our proposed method results in better attention separation when compared to the baseline FactorVAE, with high-response regions in different areas in the image.

References

[1] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *ICML*, 2018.

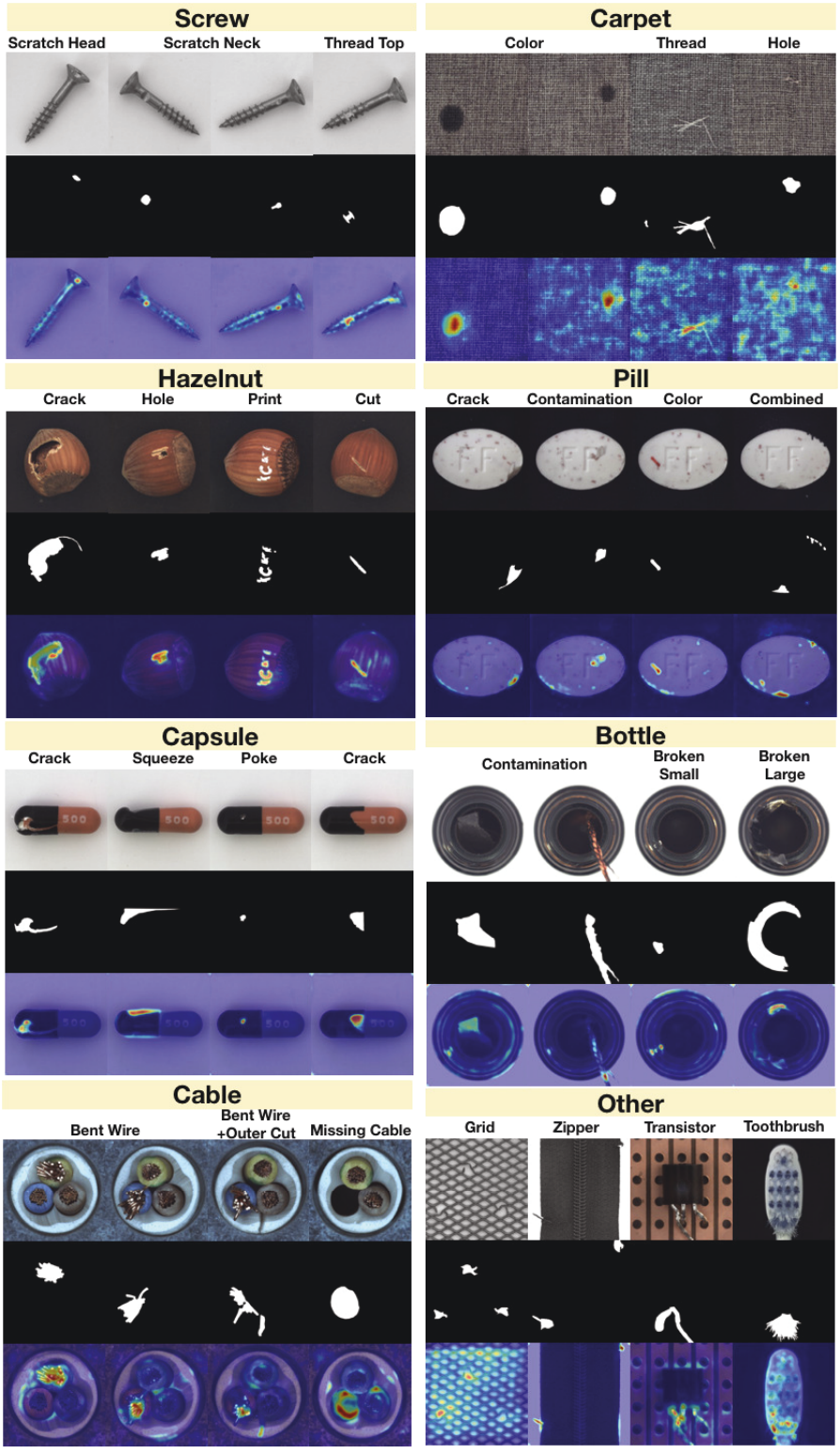

Figure 4. Additional qualitative results on the MVTec-AD dataset.

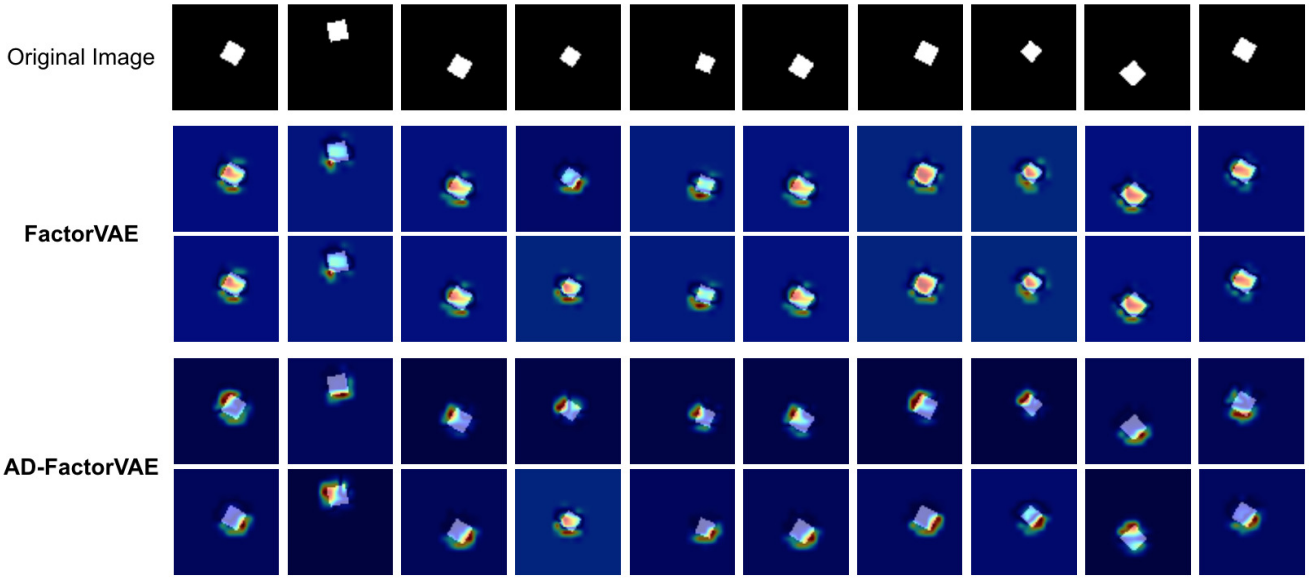

Figure 5. Attention separations of the “Square” shape of Dsprites.

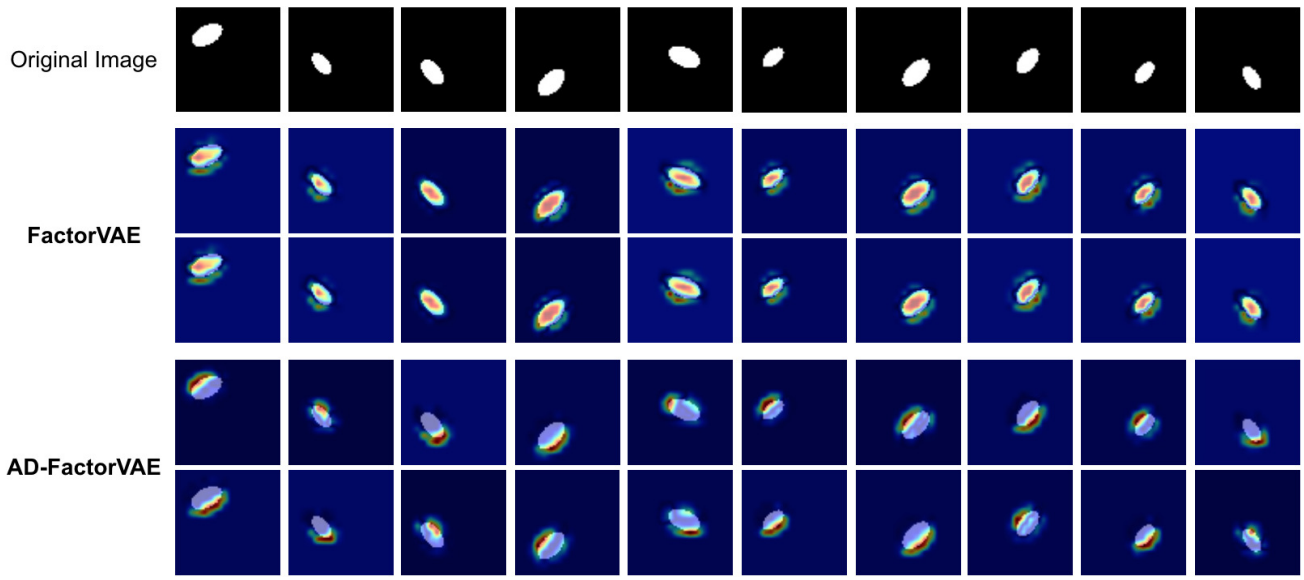

Figure 6. Attention separations of the “Ellipse” on Dsprites.

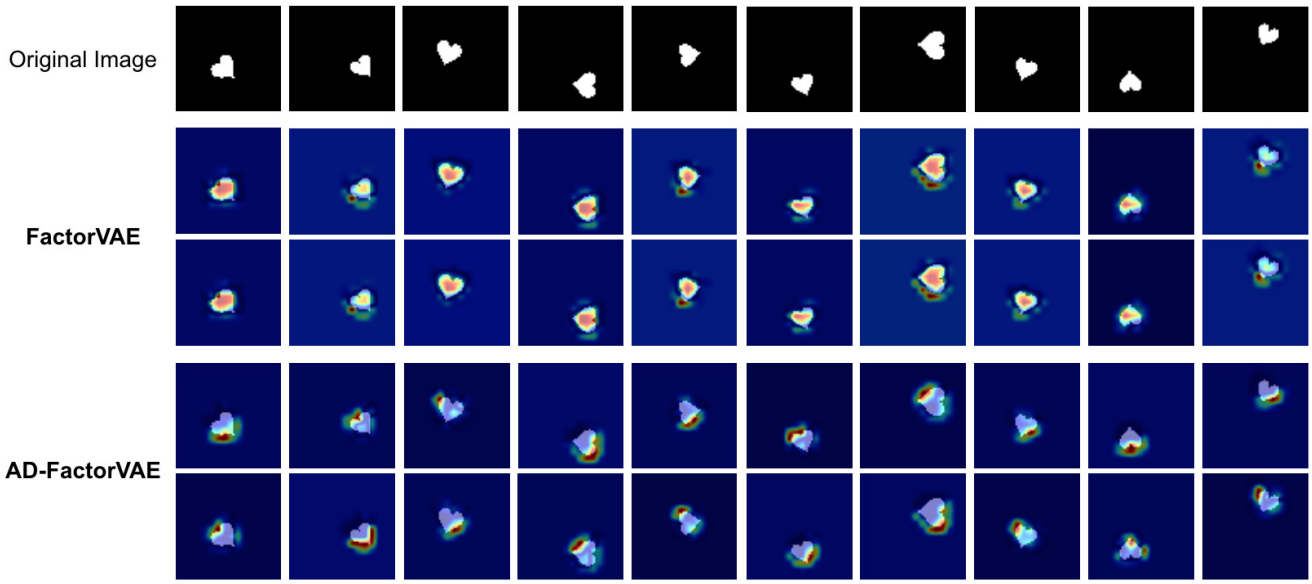

Figure 7. Attention separations of the “Heart” shape on Dsprites Dataset.

Network	Layer	Output Dimensions
	Input Image	64×64
Encoder	Conv 2D, 4×4, 32,2,1	$32 \times 32 \times 32$
	ReLU	$32 \times 32 \times 32$
	Conv 2D, 4×4, 32,2,1	$16 \times 16 \times 32$
	ReLU	$16 \times 16 \times 32$
	Conv 2D, 4×4, 64,2,1	$8 \times 8 \times 64$
	ReLU	$8 \times 8 \times 64$
	Conv 2D, 4×4, 64,2,1	$4 \times 4 \times 64$
	ReLU	$4 \times 4 \times 64$
	Conv 2D, 4×4, 128,1,1	$1 \times 1 \times 128$
	ReLU	$1 \times 1 \times 128$
	Conv 2D, 1×1, 32,1,0	32
	Conv 2D, 1×1, 32,1,0	32
	Input	$\mathbb{R}^{32}$
Decoder	Conv 2D, 1×1, 128,1,0	128
	ReLU	$1 \times 1 \times 128$
	ConvTr 2D, 4×4, 64,1,0	$4 \times 4 \times 64$
	ReLU	$4 \times 4 \times 64$
	ConvTr 2D, 4×4, 64,2,1	$8 \times 8 \times 64$
	ReLU	$8 \times 8 \times 64$
	ConvTr 2D, 4×4, 32,2,1	$16 \times 16 \times 32$
	ReLU	$16 \times 16 \times 32$
	ConvTr 2D, 4×4, 32,2,1	$32 \times 32 \times 32$
	ReLU	$32 \times 32 \times 32$
	ConvTr 2D, 4×4, 1,2,1	$64 \times 64 \times 1$

Table 4. Architecture details of the AD-FactorVAE we use for training and testing on Dsprites dataset. The notation for "Layer" column is as follows: operation, kernel size $h \times w$, number of filter channels, stride, padding. ConvTr 2D denotes 2D transpose convolution layer.