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).

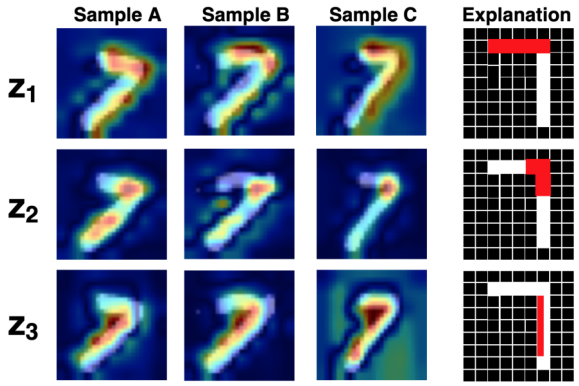

Figure 1. Element-wise attention results. Each element in the latent vector (here $z_1 - z_3$) can be explained separately with our attention maps, visualizing consistent explanations across different samples.

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.

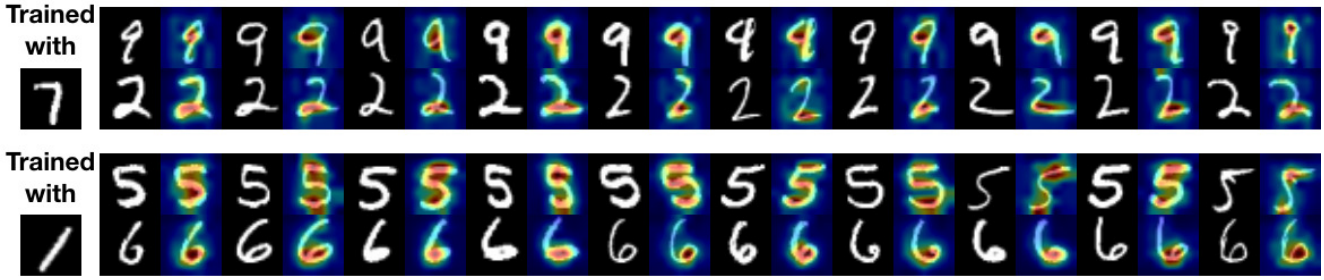

Figure 2. Additional qualitative results from MNIST dataset.

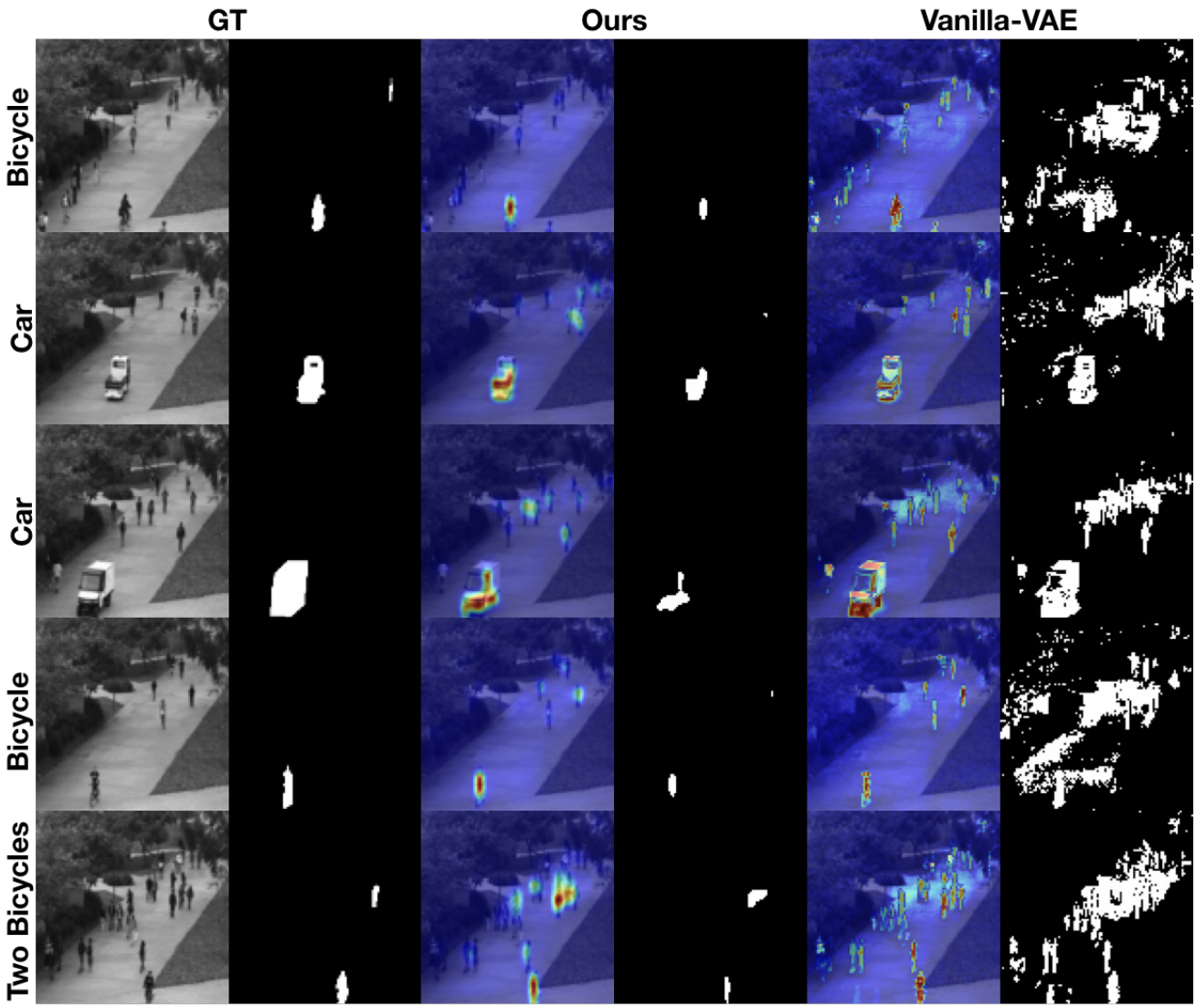

Figure 3. Additional qualitative results from UCSD Ped1 dataset. L-R: ground truth image and mask, our attention maps and masks, and Vanilla-VAE's attention maps and masks. Each row represents a different anomaly situation. Compared to vanilla-VAE, our attention maps and masks localize anomalies much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

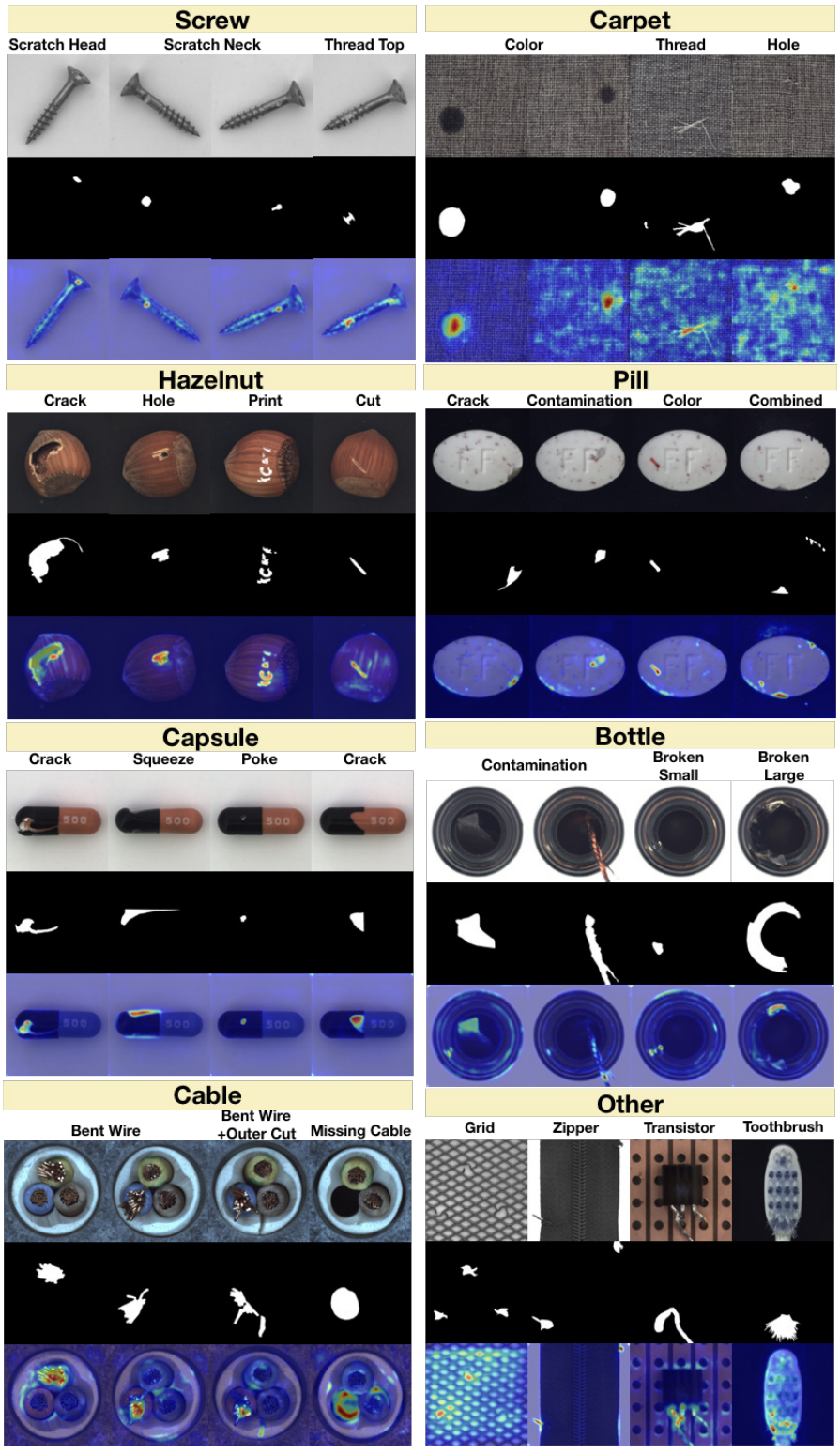

Figure 4. Additional qualitative results on the MVTec-AD dataset.

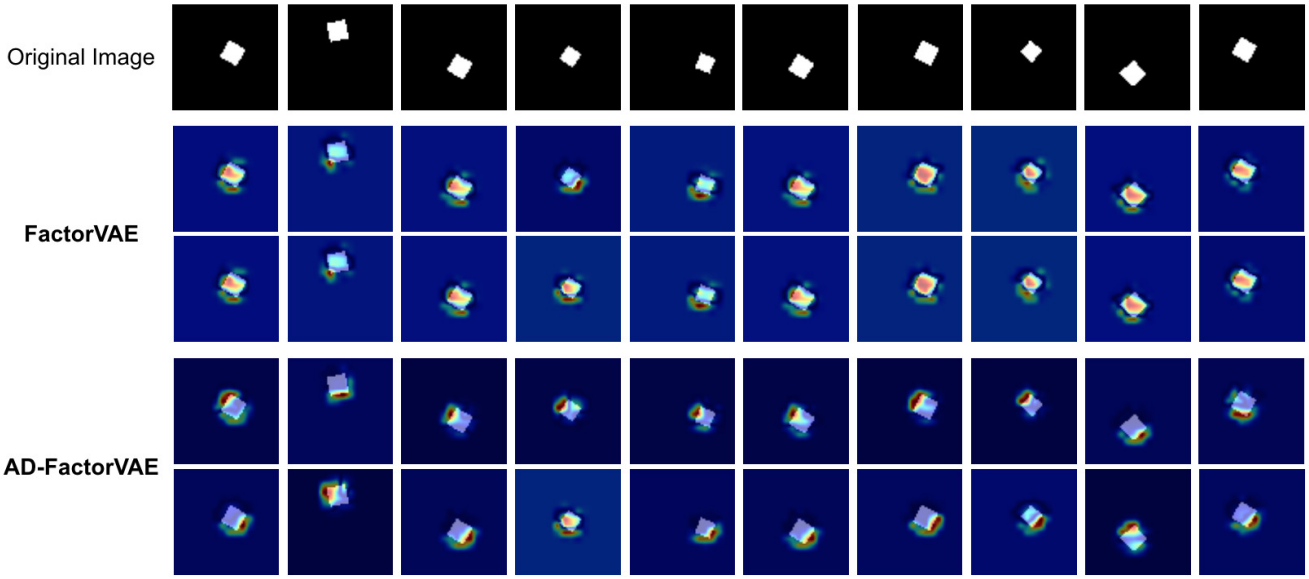

Figure 5. Attention separations of the “Square” shape of Dsprites.

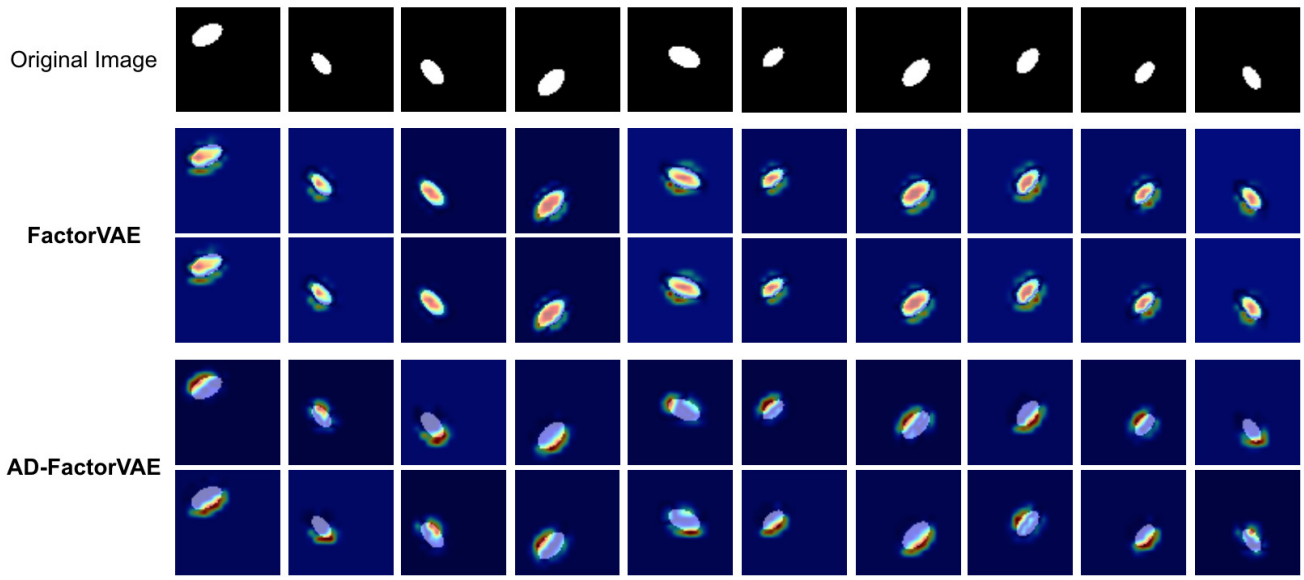

Figure 6. Attention separations of the “Ellipse” on Dsprites.

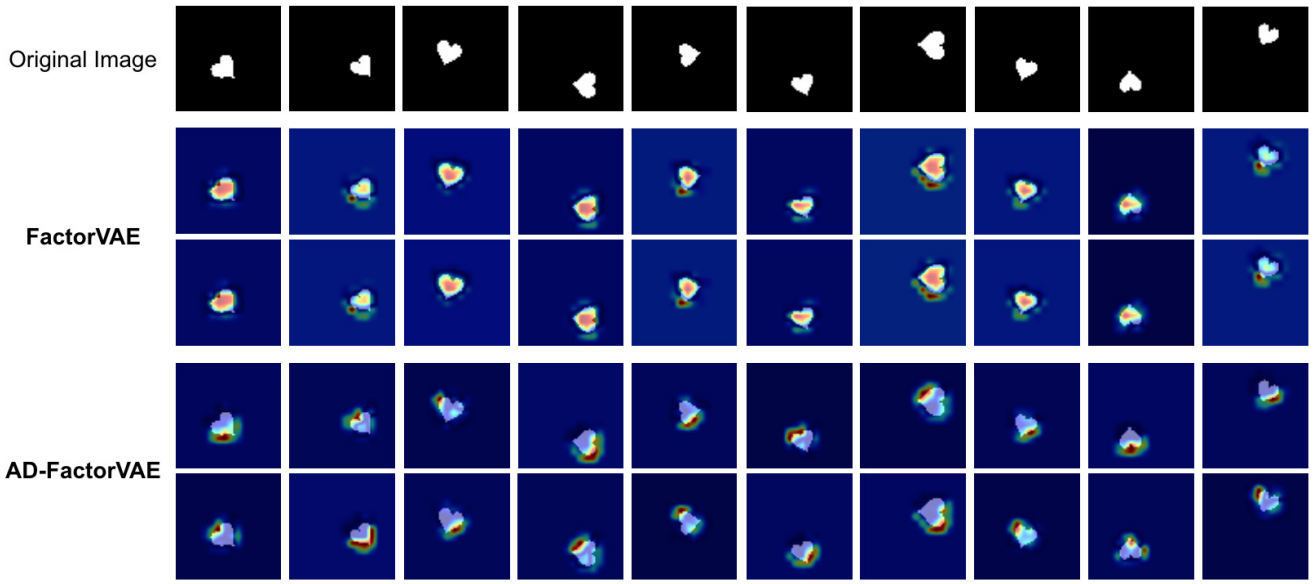

Figure 7. Attention separations of the “Heart” shape on Dsprites Dataset.

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.