Measuring the Biases and Effectiveness of Content-Style Disentanglement (Supplementary Material)

Xiao Liu [*,1]
Xiao.Liu@ed.ac.uk
Spyridon Thermos [*,1]
SThermos@ed.ac.uk
Gabriele Valvano [*,1,2]
gabriele.valvano@imtlucca.it
Agisilaos Chartsias[1]
Agis.Chartsias@ed.ac.uk
Alison O'Neil [1,3]
Alison.ONeil@mre.medical.canon
Sotirios A. Tsaftaris [1,4]
S.Tsaftaris@ed.ac.uk

[1] School of Engineering
University of Edinburgh
Edinburgh, UK

[2] IMT School for Advanced Studies Lucca
Lucca, Italy

[3] Canon Medical Research Europe Ltd.
Edinburgh, UK

[4] The Alan Turing Institute
London, UK

[*] Equal contribution

1 Maximizing Likelihood by Minimizing Mean Square Error

Let y denote a generic pixel in an image I, and $\tilde{y}$ the respective pixel in the reconstructed image $\tilde{I}$, obtained trough a learned decoding function.

If we assume the reconstruction error, denoted as ε, to be normally distributed (*i.e.* $\varepsilon \sim \mathcal{N}\left(0,\sigma^2\right)$), then, the predicted value $\tilde{y}$ is normally distributed around the true value y, thus $\tilde{y} \sim \mathcal{N}\left(y,\sigma^2\right)$. Based on this assumption, the probability density function can be defined as:

$$f\left(\tilde{y}|y,\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\tilde{y}-y)^2}{2\sigma^2}}. \tag{1}$$

Given a set of observations, *e.g.* the pixels of the image, we maximize the likelihood $\mathcal{L}$ as the product of the probability densities of the observations:

$$\mathcal{L} = \prod_{i=1}^{n} f\left(y_i|\tilde{y}_i,\sigma^2\right) = \left(2\pi\sigma^2\right)^{-n/2} e^{-\frac{\sum_{i=1}^{n}(y_i-\tilde{y}_i)^2}{2\sigma^2}}. \tag{2}$$

Assuming the variance of the error to be independent from the input variables, optimizing the latter formula is equivalent to optimize:

$$\log\left(\frac{\mathcal{L}}{\left(2\pi\sigma^2\right)^{-n/2}}\right) = -\frac{\sum_{i=1}^{n}\left(y_i-\tilde{y}_i\right)^2}{2\sigma^2}. \tag{3}$$

Table 1: *IOB* decoders design for the teapot dataset.

Decoder	Input Shape→Output Shape	Layer Information
$G_\theta(C)$	(1,64,64)→(8,64,64)	CONV-(O:8,K:7x7,S:1,P:3), IN, Leaky ReLU
	(8,64,64)→(16,32,32)	CONV-(O:16,K:4x4,S:2,P:1), IN, Leaky ReLU
	(16,32,32)→(32,16,16)	CONV-(O:32,K:4x4,S:2,P:1), IN, Leaky ReLU
	(32,16,16)→(64,8,8)	CONV-(O:64,K:4x4,S:2,P:1), IN, Leaky ReLU
	(64,8,8)→(32,16,16)	DECONV-(O:32,K:4x4,S:2,P:1), IN, Leaky ReLU
	(32,16,16)→(16,32,32)	DECONV-(O:16,K:4x4,S:2,P:1), IN, Leaky ReLU
	(16,32,32)→(8,64,64)	DECONV-(O:8,K:4x4,S:2,P:1), IN, Leaky ReLU
	(8,64,64)→(3,64,64)	CONV-(O:3,K:7x7,S:1,P:3), Tanh
$G_\theta(\underline{s})$	(3)→(256)	FC-(O:256)
	(256)→(4096)	FC-(O:4096), Flatten
	(64,8,8)→(32,16,16)	DECONV-(O:32,K:4x4,S:2,P:1), IN, Leaky ReLU
	(32,16,16)→(16,32,32)	DECONV-(O:16,K:4x4,S:2,P:1), IN, Leaky ReLU
	(16,32,32)→(8,64,64)	DECONV-(O:8,K:4x4,S:2,P:1), IN, Leaky ReLU
	(8,64,64)→(3,64,64)	CONV-(O:3,K:7x7,S:1,P:3), Tanh

Thus, maximizing the original likelihood function is equivalent to minimizing $\sum_{i=1}^{n}(y_i-\tilde{y}_i)^2$ that is the scaled Mean Squared Error (MSE). Thus, by training the decoder to minimize MSE, we train it to maximize the Mutual Information (MI) between z and I.

After training the decoder G_θ (see Sec. 3 of the main manuscript), computing MSE equivalent to directly measuring the MI. There is a relationship between likelihood and MSE (shown below), but the likelihood acts as a lower bound to MI.

Relationship MSE - likelihood: Note that if we divide both parts of the equation by n and then we multiply by $-2\sigma^2$, we obtain:

$$\sum_{i=1}^{n}\frac{(y_i-\tilde{y}_i)^2}{n} = -\frac{2\sigma^2}{n}\cdot\log\frac{\mathcal{L}}{(2\pi\sigma^2)^{-n/2}}, \tag{4}$$

that is:

$$MSE = -\frac{2\sigma^2}{n}\log(\mathcal{L}) - \sigma^2\log(2\pi\sigma^2). \tag{5}$$

Since we assume homoscedastic distributions, *i.e.* fixed σ^2, Equation 5 can be expressed as:

$$MSE = -\frac{a}{n}\log(\mathcal{L}) - b, \tag{6}$$

where a and b are positive constants.

2 Empirical Study with the Teapot Dataset

Visual examples and qualitative results of the empirical study on the proposed metrics with the teapot dataset are included in Fig. 1. It is notable that the artifacts in the reconstructed images introduced by the decoder bias are observed in the results of both decoders and bias decoders.

3 Model Design and Training Scheme for *IOB*

For the structure of *IOB* decoders, we vary the number of layers for different applications due to the different dimensions of the representations. The design of the decoders for the

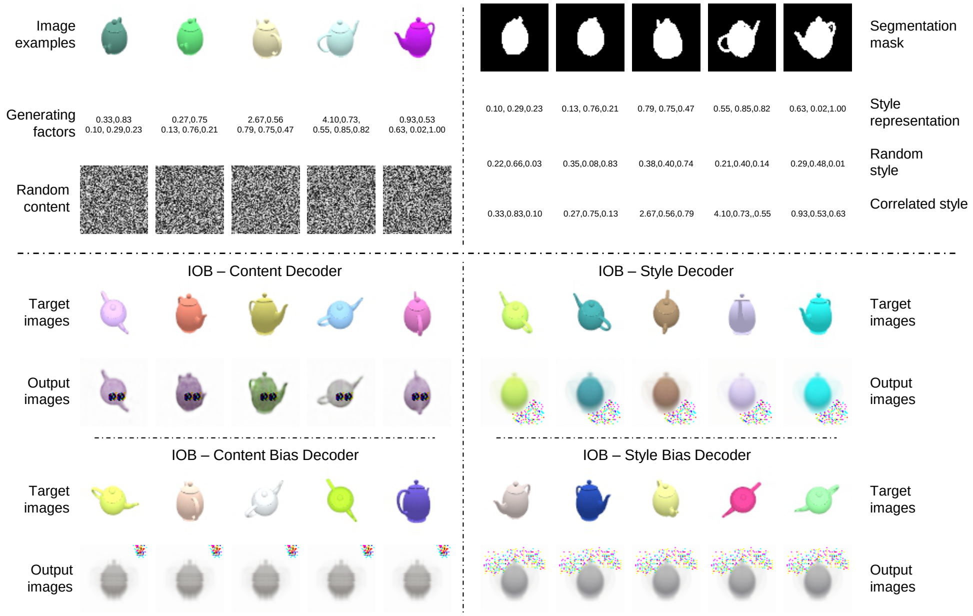


Figure 1: Visuals for the empirical study with the teapot dataset. Top: examples of original images, ground truth generating factors and segmentation masks. We also show the randomly sampled content and style representations. Bottom: examples of target images and output images for the *IOB* decoders.

teapot dataset, MUNIT, SDNet and PANet can be found in Table 1, 2, 3 and 4. The notations in the tables are: O: the number of output channels; K: the kernel size; S: the stride size; P: the padding size; FC: fully-connected layer; IN: instance normalization; Overall, $G_\theta(\underline{s})$ consists of several linear layers, followed by transpose (upsampling steps) and one plain CONV layer that generates the final image. $G_\theta(C)$ follows an autoencoder structure with several encoder and decoder CONV layers. For the teapot dataset, the content representation has size $1 \times 64 \times 64$ and the style representation has size 3. For MUNIT, the content representation has size $128 \times 64 \times 64$ and the style representation has size 8. For SDNet, the content representation has size $8 \times 224 \times 224$ and the style representation has size 8. For PANet, the content representation has size $3 \times 64 \times 64$ and the style representation has size 1024. Note that it is not necessary to have exactly same design as in the tables, where the key suggestion is to design the decoders to generate as high-quality as possible reconstructed images.

All the decoders are trained using the Adam optimiser [8] ($\beta_1 = 0.5, \beta_2 = 0.999$) with a learning rate of $1e^{-4}$ for 40 epochs using batch size 10.

4 Detailed Application Description

Table 5 summarizes the *design* and *learning biases* of the methods presented in Sec. 5 of the main manuscript. Note that the biases are reported as modules, without indicating the way they are used in our experiments (*e.g.* AdaIN is reported without specifying that it is removed from the original MUNIT, but is added to PANet as a variant).

Table 2: *IOB* decoders design for MUNIT.

Decoder	Input Shape→Output Shape	Layer Information
$G_\theta(C)$	(128,64,64)→(128,64,64)	CONV-(O:128,K:7x7,S:1,P:3), IN, Leaky ReLU
	(128,64,64)→(128,32,32)	CONV-(O:128,K:4x4,S:2,P:1), IN, Leaky ReLU
	(128,32,32)→(128,16,16)	CONV-(O:128,K:4x4,S:2,P:1), IN, Leaky ReLU
	(128,16,16)→(64,32,32)	DECONV-(O:64,K:4x4,S:2,P:1), IN, Leaky ReLU
	(64,32,32)→(32,64,64)	DECONV-(O:32,K:4x4,S:2,P:1), IN, Leaky ReLU
	(32,64,64)→(16,128,128)	DECONV-(O:16,K:4x4,S:2,P:1), IN, Leaky ReLU
	(16,128,128)→(3,128,128)	CONV-(O:3,K:7x7,S:1,P:3), Tanh
$G_\theta(\underline{s})$	(8)→(256)	FC-(O:256)
	(256)→(4096)	FC-(O:4096)
	(4096)→(8192)	FC-(O:8192), Flatten
	(128,8,8)→(64,16,16)	DECONV-(O:64,K:4x4,S:2,P:1), IN, Leaky ReLU
	(64,16,16)→(32,32,32)	DECONV-(O:32,K:4x4,S:2,P:1), IN, Leaky ReLU
	(32,32,32)→(16,64,64)	DECONV-(O:16,K:4x4,S:2,P:1), IN, Leaky ReLU
	(16,64,64)→(8,128,128)	DECONV-(O:8,K:4x4,S:2,P:1), IN, Leaky ReLU
	(8,128,128)→(3,128,128)	CONV-(O:3,K:7x7,S:1,P:3), Tanh

Table 3: *IOB* decoders design for SDNet.

Decoder	Input Shape→Output Shape	Layer Information
$G_\theta(C)$	(8,224,224)→(8,224,224)	CONV-(O:8,K:7x7,S:1,P:3), IN, Leaky ReLU
	(8,224,224)→(16,112,112)	CONV-(O:16,K:4x4,S:2,P:1), IN, Leaky ReLU
	(16,112,112)→(32,56,56)	CONV-(O:32,K:4x4,S:2,P:1), IN, Leaky ReLU
	(32,56,56)→(64,28,28)	CONV-(O:64,K:4x4,S:2,P:1), IN, Leaky ReLU
	(64,28,28)→(128,14,14)	CONV-(O:128,K:4x4,S:2,P:1), IN, Leaky ReLU
	(128,14,14)→(64,28,28)	DECONV-(O:64,K:4x4,S:2,P:1), IN, Leaky ReLU
	(64,28,28)→(32,56,56)	DECONV-(O:32,K:4x4,S:2,P:1), IN, Leaky ReLU
	(32,56,56)→(16,112,112)	DECONV-(O:16,K:4x4,S:2,P:1), IN, Leaky ReLU
	(16,112,112)→(8,224,224)	DECONV-(O:8,K:4x4,S:2,P:1), IN, Leaky ReLU
	(8,224,224)→(1,224,224)	CONV-(O:1,K:7x7,S:1,P:3), Tanh
$G_\theta(\underline{s})$	(3)→(256)	FC-(O:256)
	(256)→(4096)	FC-(O:4096)
	(4096)→(25088)	FC-(O:25088), Flatten
	(128,14,14)→(64,28,28)	DECONV-(O:64,K:4x4,S:2,P:1), IN, Leaky ReLU
	(64,28,28)→(32,56,56)	DECONV-(O:32,K:4x4,S:2,P:1), IN, Leaky ReLU
	(32,56,56)→(16,112,112)	DECONV-(O:16,K:4x4,S:2,P:1), IN, Leaky ReLU
	(16,112,112)→(8,224,224)	DECONV-(O:8,K:4x4,S:2,P:1), IN, Leaky ReLU
	(8,224,224)→(1,224,224)	CONV-(O:1,K:7x7,S:1,P:3), Tanh

4.1 MUNIT for Image-to-Image Translation

Multimodal Unsupervised Image-to-image Translation (MUNIT) [7] does not impose strict constraints on the learned representations, and achieves disentanglement with both design and learning biases.

The basic assumption is that multi-domain images (a necessary *data bias*), share common content information, but differ in style. A content encoder maps images to multi-channel feature maps, by removing style with IN layers [6] (*design bias*). A second encoder extracts global style information with fully connected layers and global pooling. Finally, style and content are combined in a decoder with AdaIN modules [6] (*design bias*).

Disentanglement is additionally promoted with a bidirectional reconstruction loss [14] that enables style transfer. In order to learn a smooth representation manifold, two LR losses (*learning bias*) are applied on content and style extracted from input images: content LR penalizes the distance to the content extracted from reconstructed images, whereas style LR encourages encoded style distributions to match their Gaussian priors. Finally, adversarial

Table 4: *IOB* decoders design for PANet.

Decoder	Input Shape→Output Shape	Layer Information
$G_\theta(C)$	(3,64,64)→(16,64,64)	CONV-(O:16,K:7x7,S:1,P:3), IN, Leaky ReLU
	(16,64,64)→(32,32,32)	CONV-(O:32,K:4x4,S:2,P:1), IN, Leaky ReLU
	(32,32,32)→(16,64,64)	DECONV-(O:16,K:4x4,S:2,P:1), IN, Leaky ReLU
	(16,64,64)→(8,128,128)	DECONV-(O:8,K:4x4,S:2,P:1), IN, Leaky ReLU
	(8,128,128)→(3,128,128)	CONV-(O:3,K:7x7,S:1,P:3), Tanh
$G_\theta(\underline{s})$	(1024)→(1,32,32)	Flatten
	(1,32,32)→(16,64,64)	DECONV-(O:16,K:4x4,S:2,P:1), IN, Leaky ReLU
	(16,64,64)→(8,128,128)	DECONV-(O:8,K:4x4,S:2,P:1), IN, Leaky ReLU
	(8,128,128)→(3,128,128)	CONV-(O:3,K:7x7,S:1,P:3), Tanh

Table 5: Overview of the *design* and *learning biases* that are investigated in the context of the three investigated vision tasks: a) image-to-image translation (MUNIT), b) medical segmentation (SDNet), and c) pose estimation (PANet).

		MUNIT	**SDNet**	**PANet**
Design Bias	AdaIN	√		√
	Instance Normalization	√		
	SPADE		√	
	Binarization		√	
	MLP			√
Learning Bias	Latent Regression	√	√	
	KL Divergence		√	
	Equivariance			√

learning encourages realistic synthetic images.

4.2 SDNet for Medical Image Segmentation

SDNet [4] is a semi-supervised framework that disentangles medical images in anatomical features (content) and imaging-specific characteristics (style). Similarly to other models, SDNet uses separate content and style encoders, but here a segmentation network is applied on the content features trained with supervised objectives and annotated images (*data bias*).

However, in contrast to MUNIT, SDNet does not impose a design bias on the encoder, but rather on the content which is represented as multi-channel binary maps of the same resolution as the input (*design bias*).

This is obtained with a softmax and a thresholding function with the straight-through operator [2], such that any style is removed from the content. To encourage style features to encode residual information (and not content), a loss enforces the style representation to approximate a standard Gaussian, following the VAE formulation [9] (*learning bias*). In this setup, any information encoded in style comes at a cost, and thus encoding redundant information is prevented [1]. Furthermore, a LR loss of the style is employed to prevent posterior collapse of the decoder (*learning bias*).

Finally, style and content are combined to reconstruct the input image by applying a series of convolutional layers with feature-wise linear modulation (FiLM) conditioning. Similarly to AdaIN, FiLM modules are restrictive, allowing the style only to normalize the conditioned feature maps, and thus further discouraging the style from encoding content information (*design bias*).

4.3 PANet for Pose Estimation

For the pose estimation task, we consider a dual-stream autoencoder denoted as PANet [11]. PANet consists of two branches that decouple pose (content) and appearance (style) but employs heavily entangled encoders-decoders.

The content is represented as a multi-channel feature map, where each channel corresponds to a specific body part (since the number of parts are fixed, this imposes a strong *data bias*). A Gaussian distribution is applied to each feature map to remove any style information, whilst also preserving the spatial correspondence (*design bias*).

The corresponding style information is extracted from the encoder features using average pooling (*design bias*). More critically, style vectors do not correspond to global image style, since they are applied to specific content parts during decoding (*design bias*).

Finally, disentanglement is encouraged with a transformation equivariance loss (*learning bias*). This ensures that the spatial transformations, such as translations and rotations, affect only the content, while the intensity ones, such as the color and texture information, affect only the style.

5 SYNTHIA-Cityscapes Description and MUNIT Training Setup

Data. We use SYNTHIA [13], which consists of over $20,000$ rendered images and corresponding pixel-level semantic annotations, where 13 classes of objects are labeled for aiding segmentation and scene understanding problems. We also use Cityscapes [5], which contains a set of diverse street scene stereo video sequences and over 5k frames of high-quality semantic annotations, where 30 classes of instances are labeled in the segmentation masks.

Training setup. MUNIT achieves unsupervised multi-modal image-to-image translation by minimizing the following loss function:

$$\mathcal{L}_{total} = \mathcal{L}_{GAN} + \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{c-rec} + \lambda_3 \mathcal{L}_{s-rec}, \tag{7}$$

where $\mathcal{L}_{rec}$ is the image reconstruction loss, $\mathcal{L}_{c-rec}$ and $\mathcal{L}_{s-rec}$ are the content and style reconstruction losses, and $\lambda_1 = 10$, $\lambda_2 = 1$ are the hyperparameters used by the authors in [7].

6 ACDC Description and SDNet Training Setup

Data. We use data from the Automatic Cardiac Diagnosis Challenge (ACDC) [3], which contains cardiac cine-MR images acquired from different MR scanners and resolution on 100 patients. Images were resampled to 1.37 mm^2/pixel resolution and cropped to 224×224 pixels. Manual segmentations are provided for the left ventricular cavity, the myocardium and right ventricle in the end-systolic and end-diastolic cardiac phases. In total there are 1920 images with manual segmentations and 23,530 images with no segmentations.

Training setup. SDNet is trained by minimizing the following loss function:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{KL} + \lambda_2 \mathcal{L}_{seg} + \lambda_3 \mathcal{L}_{rec} + \lambda_4 \mathcal{L}_{z_{rec}}, \tag{8}$$

where $\mathcal{L}_{KL}$ is the KL Divergence measured between the sampled and the predicted style vectors, $\mathcal{L}_{rec}$ is the image reconstruction loss, $\mathcal{L}_{seg}$ is the anatomy segmentation loss, and

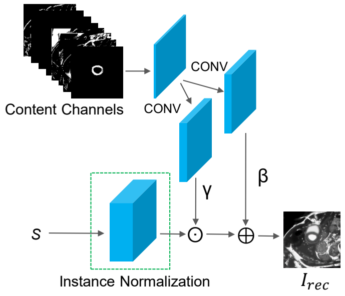


Figure 2: Detailed representation of the SPADE decoder [12] in the context of SDNET [4]. Style is denoted as S, while CONV represents the convolution operation. Note that γ and β parameters are applied to the normalized style activations through element-wise multiplication and addition, respectively.

$\mathcal{L}_{z_{rec}}$ is the LR loss between the sampled and the re-encoded style vector. $\lambda_1 = 0.01, \lambda_2 = 10, \lambda_3 = 1$, and $\lambda_4 = 1$ are the hyperparameters used by the authors in [4].

6.1 SPADE Decoder

As described in Sec. 5 of the main manuscript, SDNet relies on a FiLM-based decoder to combine the content and style information and reconstruct the input image. The key characteristic of FiLM is that it gradually adds style information to the content-based reconstruction process. Additionally, an alternative approach for combining the content and style features is investigated, by using a SPADE decoder [12] to further expose the design bias added by the decoder architecture.

A SPADE block receives the content channels and projects them onto an embedding space using two convolutional layers to produce the modulation parameters (tensors) γ and β. These parameters are then used to scale (γ) and shift (β) the normalized activations of the style representation. We utilize multiple SPADE blocks to fuse content and style information at different levels of granularity during decoding. A schematic of the utilized SPADE decoder in the context of SDNet is depicted in Fig. 2.

Table 6: Comparative evaluation of SDNet [4] variants on the ACDC [3] dataset with 100% annotation masks, using the proposed metrics. The *Dice Score* metric is used to measure the performance in terms of semantic segmentation.

		Learning Bias	**Design Bias**	
SDNet	Original Model	w/o KLD and Latent Regression	w/o Binarization	SPADE
$DC(C,S)$ (↓)	0.48	0.57	**0.43**	0.59
$DC(I,C)$ (↑)	0.97	0.95	**0.97**	0.94
$DC(I,S)$ (↑)	0.44	0.53	0.44	**0.57**
$IOB(I,C)$ (↑)	5.66	3.86	**6.21**	5.63
$IOB(I,S)$ (↑)	0.99	0.96	1.00	**1.02**
Dice Score (↑)	0.82	0.81	0.82	**0.83**

6.2 Medical Segmentation (100% Annotations)

In Sec. 5.2 of the main manuscript, we present the results of the SDNet model variants trained with minimal supervision, using only the 1.5% of the provided ACDC [3] annotations. Here, we provide the results for the same experiment but using the 100% of the provided annotations. From the results reported in Table 6, it can be seen that when using strong inductive biases, such as the supervised losses in this experiment, the degree of disentanglement does not significantly affect the segmentation performance (utility).

7 DeepFashion Description and PANet Training Setup

Data. We use DeepFashion [10], a large-scale dataset with over 800,000 diverse images of people in different poses and clothing, that also has annotations of body joints. We only used full-body images, specifically 32k images for training and 8k images for testing.

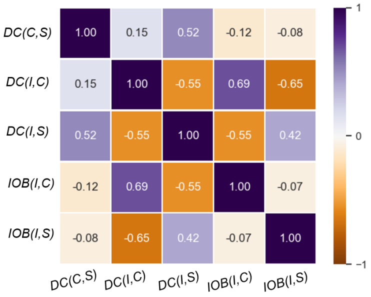
Training setup. PANet is trained in an unsupervised way with the following loss function:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{equiv}, \tag{9}$$

where $\mathcal{L}_{rec}$ is the mean absolute error between the reconstructed and the input image. $\mathcal{L}_{equiv}$ is an equivariance cost, that ensures that the mean and covariance of the parts coordinates don't change after some style transformation. Based on the implementation details presented in [11], we set $\lambda_1 = \lambda_2 = 1$.

8 Metrics Correlation and Disentanglement-performance Trade-off

As noted in Sec. 3 of the main manuscript, we report that the proposed metrics are uncorrelated with each other. Here, we present the Pearson correlation computed between disentanglement and performance metrics for each of the investigated models. Intuitively, contrary to the desired low (or no) correlation between disentanglement metrics across all models (see Fig. 3), we would expect that the performance metric(s) of each application would be correlated with at least one *DC* or *IOB* variant. In fact, this correlation can be exploited to find the "sweet spot" between disentanglement and performance. Fig. 4 confirms our intuition for



	DC(C,S)	DC(I,C)	DC(I,S)	IOB(I,C)	IOB(I,S)
DC(C,S)	1.00	0.15	0.52	-0.12	-0.08
DC(I,C)	0.15	1.00	-0.55	0.69	-0.65
DC(I,S)	0.52	-0.55	1.00	-0.55	0.42
IOB(I,C)	-0.12	0.69	-0.55	1.00	-0.07
IOB(I,S)	-0.08	-0.65	0.42	-0.07	1.00

Figure 3: Pearson correlation coefficients of the proposed metrics across all models visualized as a heatmap. Values close to 1 and -1 indicate a strong correlation.

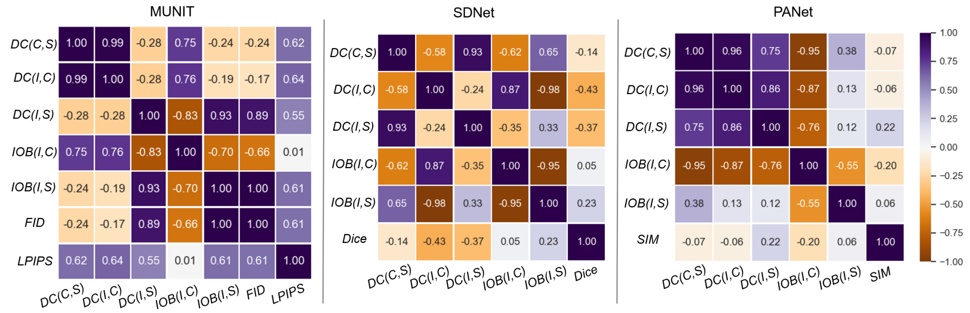


Figure 4: Pearson correlation of the proposed metrics across all applications/models visualized as heatmap. Values close to 1 and -1 indicate strong correlation.

all investigated models, highlighting the strong correlation of FID and LPIPS in the MUNIT scenario, which is the only model that utilizes both C and S directly in the main task, *i.e.* I2I translation, and not in any parallel one.

9 Qualitative Evaluation

We visualize the content and style representations in order to reason about their interpretability. We consider the content semantic if distinct objects appear in different channels, whereas the style is semantic when images reconstructed while traversing the style manifold between two points have smooth appearance changes, and are realistic.

As an extension of the samples presented and discussed in Sec. 5.4 of the main manuscript, here we provide visualizations for all model variants. In particular, Figs. 5 and 6 depict several channels of content, as well as style traversals for different MUNIT and SDNet model variants, respectively. However, Fig. 7 presents solely content representations, as PANet does not assume a prior distribution on the style latent vector, thus style traversals are not possible. When interpolating between two style vectors, the originally proposed MUNIT produces realistic images, and smooth appearance changes. Instead, removing the LR constraint affects the image quality. Similarly, the original SDNet presents high image quality and smooth transitions, while removing the content Binarization leads to low intensity (style) diversity.

References

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[3] Olivier Bernard, Alain Lalande, Clement Zotti, and et al. Deep learning techniques

for automatic MRI cardiac multi-structures segmentation and diagnosis: is the problem solved? *IEEE Transactions on Medical Imaging (TMI)*, 37(11):2514–2525, 2018.

[4] Agisilaos Chartsias, Thomas Joyce, Giorgos Papanastasiou, Scott Semple, Michelle Williams, David E. Newby, Rohan Dharmakumar, and Sotirios A. Tsaftaris. Disentangled representation learning in cardiac image analysis. *Medical Image Analysis*, 58, 2019.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[6] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1501–1510, 2017.

[7] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proc. European Conference on Computer Vision (ECCV)*, pages 179–196, 2018.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR)*, 2014.

[10] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. DeepFashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1096–1104, 2016.

[11] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Bjorn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10955–10964, 2019.

[12] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2337–2346, 2019.

[13] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016.

[14] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 465–476, 2017.

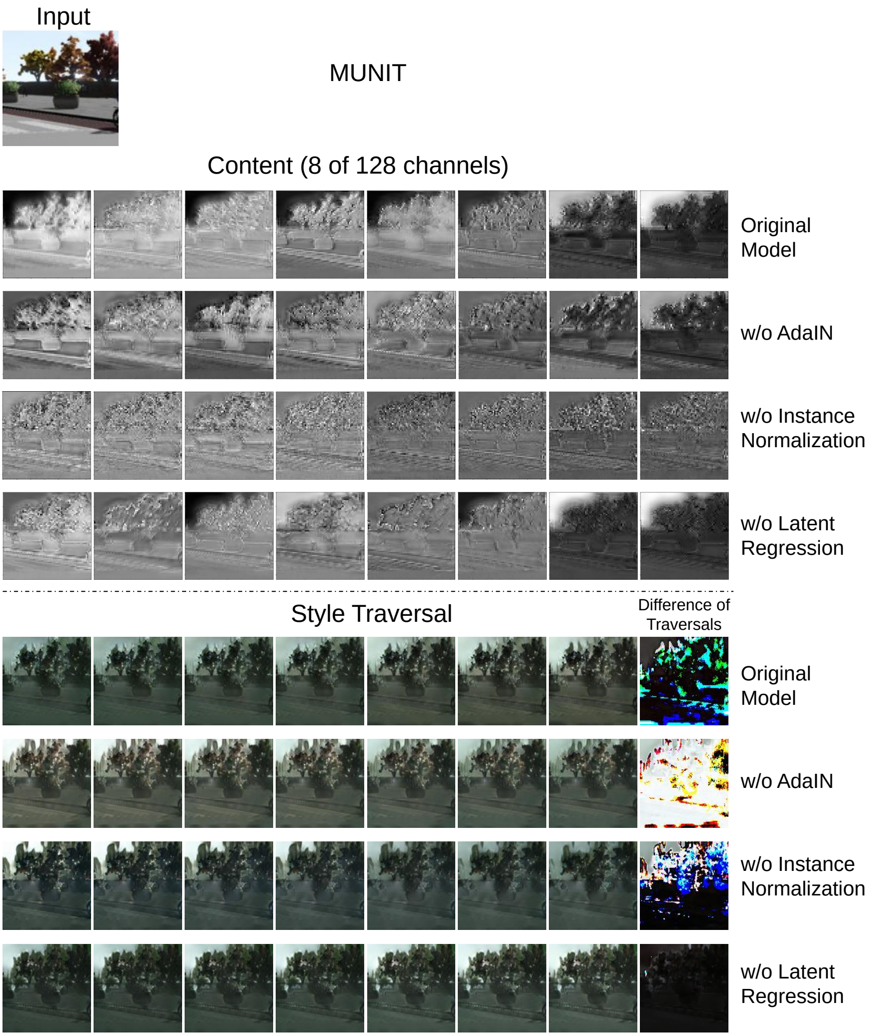


Figure 5: MUNIT: Qualitative examples to assess the interpretability of the content and style representations of the investigated model variants for different biases. For each variant, we show 8 channels of the content and 7 indicative style traversals and the difference between the first and last traversal images. The input image is depicted at the top left of the figure.

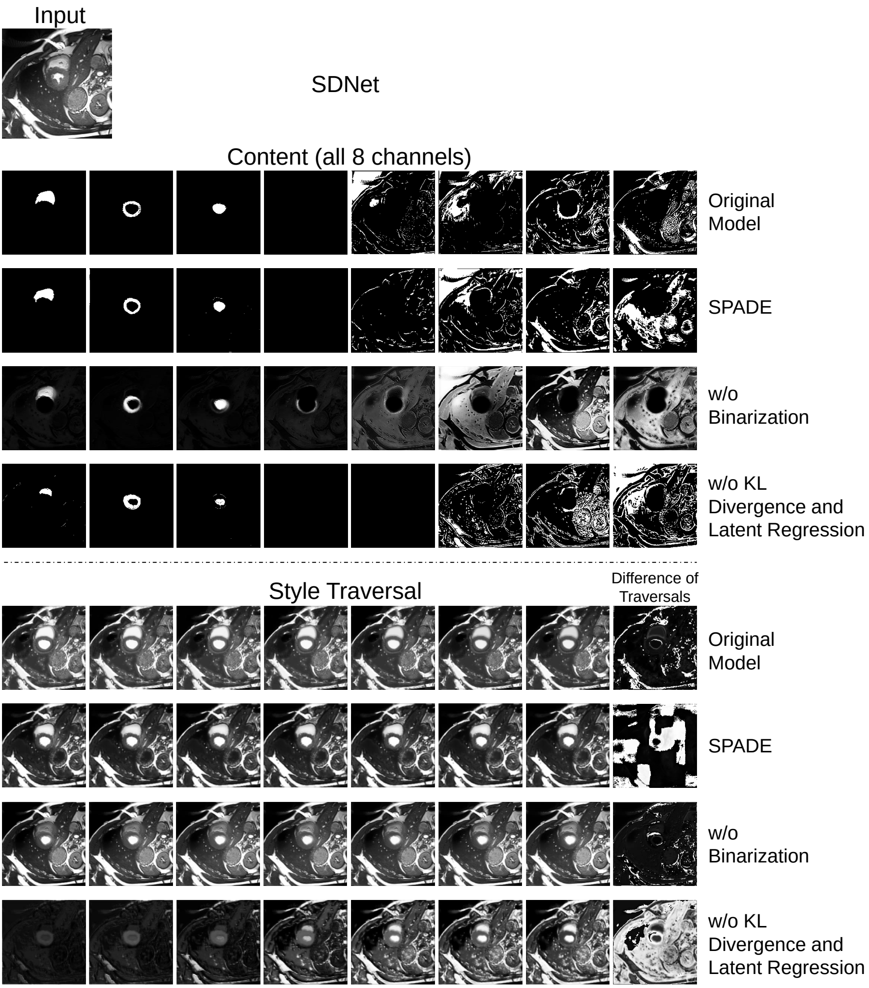


Figure 6: SDNet: Qualitative examples to assess the interpretability of the content and style representations of the investigated model variants for different biases. For each variant, we show 8 channels of the content and 7 indicative style traversals and the difference between the first and last traversal images. The input image is depicted at the top left of the figure.

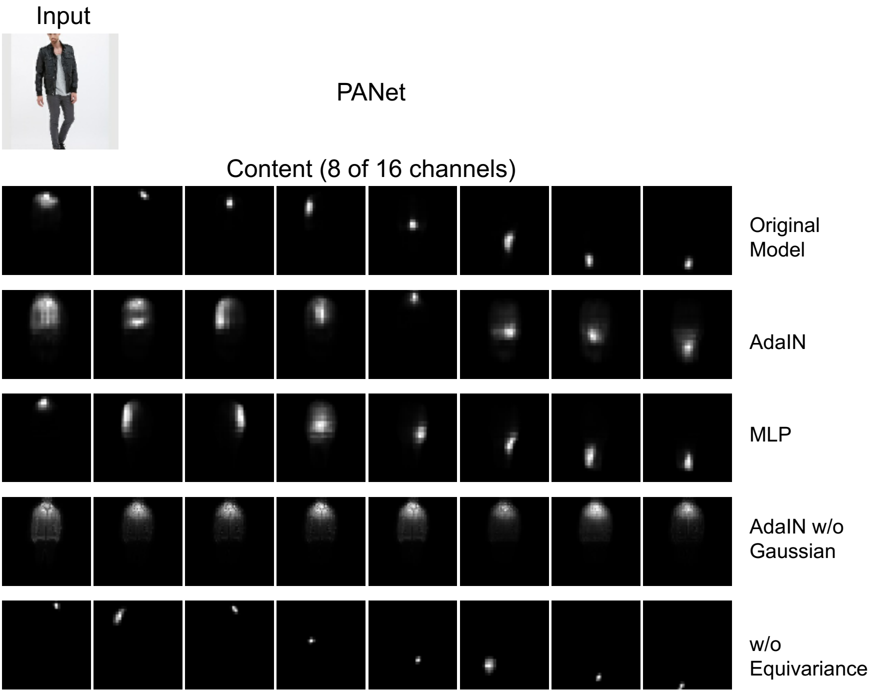


Figure 7: PANet: Qualitative examples to assess the interpretability of the content and style representations of the investigated model variants for different biases. For each variant, we show 8 channels of the content. Note that since PANet does not assume a prior distribution on the style, no style are shown. The input image is depicted at the top left of the figure.