A Contrastive Objective for Learning Disentangled Representations

Jonathan Kahana and Yedid Hoshen

Hebrew University Of Jerusalem, Jerusalem, Israel

1 Augmentations Ablation Study

1.1 Investigating the Inductive Biases of Generative Models

We designed a principled (although not necessarily optimal) approach for selecting augmentations. The main idea is to select augmentations that have similar invariances as autoencoders - as autoencoders are often used in disentanglement. We conduct an experiment on the CelebA [4], Cars3D [2] and Edges2Shoes (shoes only) [7] datasets. The experiment consists of several stages: i) train an autoencoder AE on an image dataset without any augmentations s.t. $\min_{AE} \sum_{x \in \mathcal{X}} \|x - AE(x)\|^2$, where $\mathcal{X}$ is the training set. ii) transform image from the test set of the same dataset with a range of image augmentations T iii) evaluate the invariance of the outputs of the autoencoder. We use the following two invariance metrics f_{unnorm}, f_{norm} for evaluating how much the distance between the original and transformed images change when evaluated on autoencoder outputs. Meaning, for each image augmentation $t \in T$ we measure the followings:

$$f_{unnorm} = dist(AE(x), AE(t(x))) \tag{1}$$

$$f_{norm} = \frac{dist(AE(x), AE(t(x)))}{dist(x, t(x))} \tag{2}$$

We use a VGG [6] based perceptual loss as the distance function. If an autoencoder is invariant to a particular transformation, **both** metrics should be small. The normalized metric is sensitive to smaller transformations, and the unnormalized metric is sensitive to larger transformations. We average the results over different datasets in Tab. 1 while the full results are shown in Tab. 2.

We observe that autoencoders are highly invariant to gaussian blurring, high saturation and high contrast. As these are the inductive biases of generative methods, it suggests that providing these biases to discriminative methods can potentially transfer some of the attractive qualities of generative methods.

1.2 Additional Experiments

In addition to the experiments presented in Sec. 1.1, we perform another analysis to look for other augmentations that might be useful for our method. We

Table 1: An evaluation of the invariance of autoencoders to different image transformations

	Average	
	f_{norm}	f_{unnorm}
Horizontal Flip	0.868	0.2489
Vertical Flip	0.791	0.3125
Low Contrast	1.841	0.0903
Low Brightness	0.759	0.1666
Color Rotation	0.876	0.1020
Random Erase [8]	0.796	0.1967
Affine Transformation	**0.650**	0.382
GrayScale	0.787	**0.0570**
Crop	0.842	0.2053
High Brightness	0.806	**0.0759**
High Contrast	**0.433**	**0.0572**
High Saturation	**0.579**	**0.0261**
Gaussian Blurring	**0.315**	**0.0378**

propose the following experiment: training DCoDR-norec with only a single augmentation. We measure the invariance and informativeness resulted by this augmentation, mostly considering the *invariance*, in order to not interfere with the disentanglement process. We show our results over the SmallNorb [3] dataset in Fig. 1. Our results show that the crop augmentation performed well on invariance (as we defined it in Sec. 3.2 of the paper), and extremely well for the informativeness. For those reasons we decided to insert it to our augmentations set as well, reaching a total of 4: i) Gaussian Blurring ii) High Contrast iii) High Saturation iv) Cropping. Note, that in specific cases (e.g. Edges2Shoes) adding or removing augmentations might result in even better metrics, and we encourage future research to find a better selection method of a set of augmentations, which might be specific for each dataset. Saying that, our choice of transformations is reasonable, and not particularly optimized to a specific dataset.

1.3 Augmentations Detailed Experimental Results

We show the results over all augmentations in each dataset. We measure the autoencoder's invariance to several augmentations (not to be confused with our invariance metric from other sections) by f_{unnorm} (1) and f_{norm} (2). Results are shown in Tab. 2. We present the details of each augmentation in PyTorch TorchVision library [5] -style notations (but note this is psuedo-code):

- Horizontal Flipping: RandomHorizontalFlip(p=1.)
- Vertical Flipping: RandomVerticalFlip(p=1.)
- Low Contrast: ColorJitter(contrast=(0.3, 0.8))
- Low Brightness: ColorJitter(brightness=(0.3, 0.8))

- Color Rotation: ColorJitter(hue=(0.2, 0.5) or hue=(-0.5, -0.2))
- Random Erase: RandomErasing(p=1.)
- Affine Transformation: RandomAffine(degrees=(-40, 40), fill=255)
- GrayScale: RandomGrayscale(p=1.)
- Crop: RandomResizedCrop(scale=(0.5, 1.))
- High Brightness: ColorJitter(brightness=(1.4, 1.8))
- High Contrast: ColorJitter(contrast=(1.8, 3.0))
- High Saturation: ColorJitter(saturation=(1.8, 3.0))
- Gaussian Blurring: GaussianBlur(kernel_size=5, sigma=1.)

Table 2: A detailed evaluation of the invariance of autoencoders to different image transformations. f_{unnorm} (1) and f_{norm} (2) are described in Sec. 1.1

	Cars3D		**Edges2Shoes**		**CelebA**	
	f_{norm}	f_{unnorm}	f_{norm}	f_{unnorm}	f_{norm}	f_{unnorm}
Horizontal Flip	0.861	0.1212	0.868	0.3387	0.876	0.2869
Vertical Flip	0.853	0.1991	0.770	0.3458	0.751	0.3925
Low Contrast	1.231	0.0710	3.537	0.1442	0.756	0.0557
Low Brightness	0.749	0.1978	1.098	0.2265	0.429	0.0755
Color Rotation	1.118	0.0725	1.015	0.0829	0.496	0.1507
Random Erase [8]	0.728	0.1994	0.884	0.1994	0.776	0.1968
Affine Transformation	0.479	0.2658	0.734	0.4116	0.736	0.4675
GrayScale	0.957	0.0479	0.939	0.0502	0.464	0.0729
Crop	0.808	0.1683	0.830	0.2096	0.889	0.2380
High Brightness	0.355	0.0372	1.188	0.0956	0.875	0.0950
High Contrast	0.312	0.0356	0.454	0.0499	0.534	0.0862
High Saturation	0.337	0.0095	0.768	0.0243	0.632	0.0445
Gaussian Blurring	0.080	0.0111	0.180	0.0196	0.055	0.0071

1.4 Other Datasets

We leave Shapes3D out of this ablation study in order to validate that our selected augmentations are able to transfer to unseen datasets. We hypothesize our conservative selection process has a higher chance to better transfer to other datasets. That being said, and as noted in Sec. 1.2, this augmentation selection can be significantly improved, which we leave for future work.

2 Implementation Details

In this section, we elaborate the implementation details of our algorithm to ensure reproducibility. Note that code is also included.

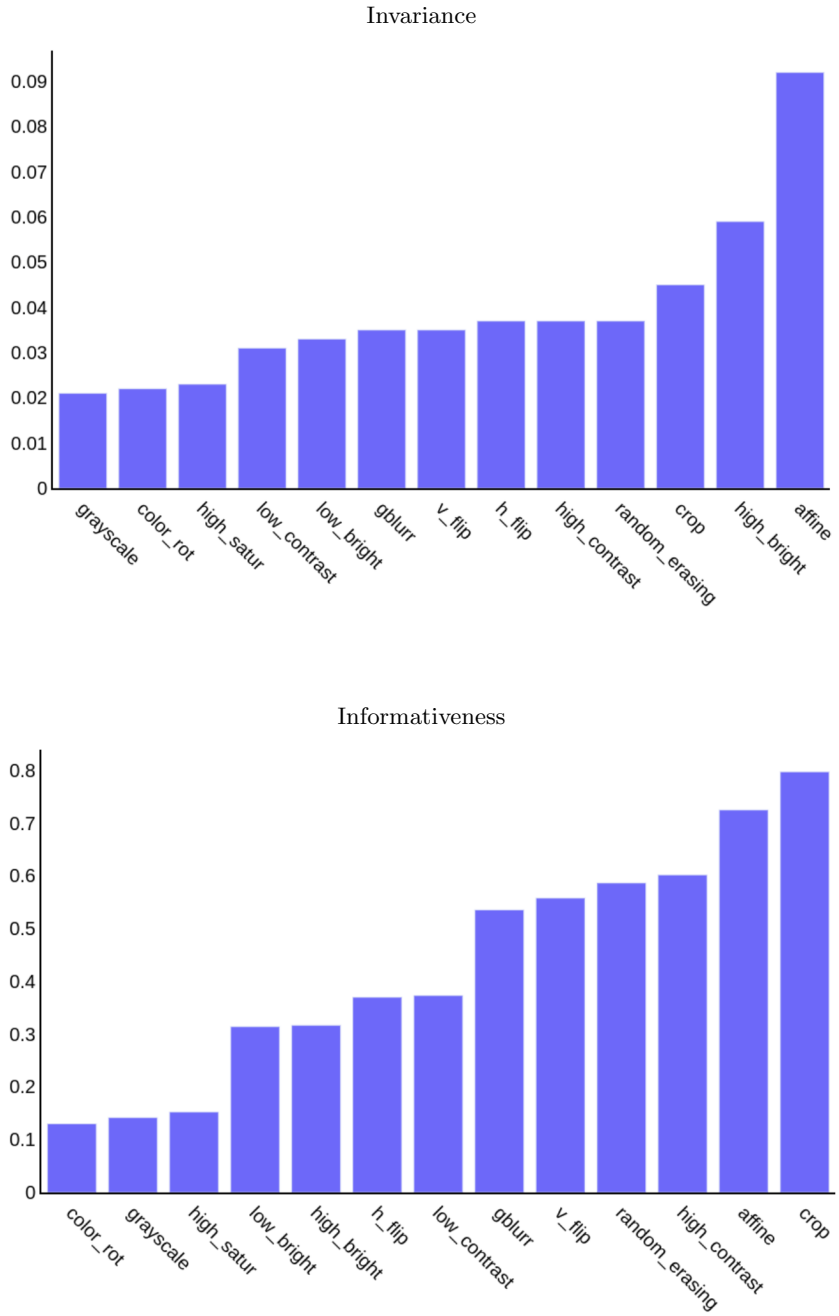

Fig. 1: Invariance and Informativeness of different augmentations.

Datasets We use 5 datasets for our evaluation: Cars3D [2], SmallNorb [3], Shapes3D [1], CelebA [4] and Edges2Shoes [7]. All datasets are used in 64x64 resolution. Shapes3D is subsampled as described in Sec. 5 in the paper. Since our method uses contrastive learning over each domain separately, it requires several examples from each class to achieve uniformity. For this reason, and for the CelebA dataset alone, we limit the minimum number of samples in each class to be 20, ignoring all classes which have 19 or less samples in the training set alone. Note that i) we do that for our method alone, while the other methods use the entire training dataset ii) the test set is unchanged, except for classes with a single example which are removed, as they cannot be evaluated using a classifier. This filtering causes our method to train on 62% of the training classes which are 81% of the training samples in CelebA.

Architecture. We use a ResNet50 encoder. For SmallNORB [3] and Shapes3D [1] we add 3 fully-connected layers at the end of the encoder. In line with other methods such as LORD, for our generator we use a VGG based perceptual loss pre-trained on ImageNet.

Optimization hyperparameters. We use a learning rate of 0.0001 for the encoder and 0.0003 for the generator, except for CelebA where we use 0.001 for both. We train our method for 200 epochs, using a batch size of 128, composed from 32 images drawn from 4 different classes.

Temperature. We use 0.2 for Cars3D, SmallNorb and CelebA. For Shapes3D and Edges2Shoes we use 0.1.

Reconstruction Loss Weight. We use a scalar constant to weight the importance of the reconstruction loss relative to the contrastive loss. We use 0.3 for all datasets.

3 Complete Experimental Results

3.1 Representation Evaluation

We present the complete results of all experiments on the SmallNorb [3] and Shapes3D [1] datasets in Tab. 3 and Tab. 4 accordingly. For Cars3D [2], CelebA [4] we only predict a single attribute (full pose - azimuth and elevation, landmarks regression) therefore the full results have already been presented in Sec. 5.2 in the paper.

Table 3: Representation evaluation for each factor of SmallNorb.

	Domain	**Azimuth**	**Elevation**	**Lighting**
SimCLR	0.956	0.838	0.664	0.771
LORD	0.393	0.731	0.384	0.895
DrNet	0.953	0.973	0.766	0.957
ML-VAE	0.968	0.982	0.868	0.982
DCoDR-norec	0.071	0.619	0.594	0.977
DCoDR	0.143	0.684	0.695	0.977
Optimal	0.021	1	1	1

Table 4: Representation evaluation for each factor of Shapes3D.

		Colors				
	Domain	**Floor**	**Wall**	**Object**	**Scale**	**Orientation**
SimCLR	1	0.983	0.985	0.988	1	1
LORD	0.703	0.998	0.999	0.990	0.991	0.999
DrNet	0.892	1	1	1	0.999	1
ML-VAE	0.999	1	1	1	1	1
DCoDR-norec	0.246	1	0.999	0.998	0.991	0.997
DCoDR	0.245	0.999	0.999	0.999	1	0.999
Optimal	0.25	1	1	1	1	1

3.2 Retrieval

Complete Retrieval Results. We display the complete results of the retrieval task, showing retrieval accuracies for both each factor separately and perfect match retrievals. For Cars3d we have only a single attribute (pose) meaning results are displayed in Tab. 4 in the paper. Additional results are presented in Tab. 5, 6 and 7.

Error Margin in the Retrieval Task. As the changes in the azimuth property of the SmallNorb dataset are relatively small, we decided to allow an error margin of 3 for this attribute. For all other attributes in all the other datasets we require a perfect match.

Table 5: SmallNorb retrieval accuracies.

	Azimuth	**Elevation**	**Lighting**	**All**
SimCLR	0.31	0.14	0.24	0.02
LORD	0.43	0.18	0.58	0.06
DrNet	0.47	0.25	0.62	0.09
ML-VAE	0.48	0.30	0.24	0.06
DCoDR-norec	0.56	0.39	0.94	0.22
DCoDR	0.56	0.46	0.95	0.26

Table 6: Shapes3D retrieval accuracies.

	Colors					
	Floor	**Wall**	**Object**	**Scale**	**Orientation**	**All**
SimCLR	0.186	0.132	0.107	0.142	0.171	<0.01
LORD	0.99	0.99	0.93	0.87	0.94	0.77
DrNet	1	1	1	0.86	1	0.86
ML-VAE	1	1	1	0.64	0.98	0.63
DCoDR-norec	1	1	1	0.99	1	0.99
DCoDR	1	1	1	1	1	1

Table 7: Edges2Shoes retrieval accuracies.

	Shoe Type	**Gender**	**All**
SimCLR	0.67	0.56	0.40
LORD	0.85	0.74	0.66
DrNet	0.90	0.72	0.66
ML-VAE	0.91	0.70	0.65
DCoDR-norec	0.69	0.55	0.41
DCoDR	0.97	0.92	0.90

References

1. Burgess, C., Kim, H.: 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/ (2018) 5
2. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13). Sydney, Australia (2013) 1, 5
3. LeCun, Y., Huang, F.J., Bottou, L.: Learning methods for generic object recognition with invariance to pose and lighting. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition **2**, II–104 Vol.2 (2004) 2, 5
4. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: Towards diverse and interactive facial image manipulation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 1, 5
5. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019) 2
6. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 1
7. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 192–199 (2014) 1, 5
8. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 13001–13008 (2020) 2, 3