1. Supplementary material

1.1. Orthogonal parametrisation

In this section we present the details on the orthonormal matrix parametrisation we use. Let us define as Skew_d the set of all real $d \times d$ skew-symmetric matrices $\{S \mid S^T = -S\}$. We also write $\mathrm{Mat}_{d\times d}$ for the set of all real $d \times d$ matrices. Clearly, Skew_d can be associated with a $\frac{d\cdot(d-1)}{2}$-dimensional space as it is uniquely defined by the elements under the diagonal. One can consider the exponential map $\exp : \mathrm{Mat}_{d\times d} \to \mathrm{Mat}_{d\times d}$ that can be defined explicitly as $\exp(A) = \sum\limits_{n=0}^{\infty} \frac{A^n}{n!}$. It is known that this sum always converges and defines a bijective smooth map from Skew_d to the space of all orthogonal matrices with a positive determinant $SO(d)$. Once we are looking for non-oriented interpretable directions, without loss of generality we may assume that the desired latent basis has positive orientaion. Otherwise we may flip the first direction. Following these considerations, we may use $\mathbb{R}^{\frac{d\cdot(d-1)}{2}}$ as the latent directions parametrisation once we set them to be orthonormal. See e.g. Fulton & Harris (2013) for further details.

1.2. Direction Variation Naturalness (DVN)

We also experimented with an alternative measure for individual interpretability that does not require human supervision. We refer to this measure as Direction Variation Naturalness (DVN). DVN measures how "natural" is a variation of images obtained by moving in a particular direction in the latent space. Intuitively, a natural factor of variation should appear in both real and generated images. Furthermore, if one splits the images based on the large/small values of this factor, the splitting should operate similarly for real and generated data. We formalize this intuition as follows. Let us have a direction $h_{\mathcal{F}}$ in the latent space that corresponds to a semantic factor $\mathcal{F}$. We always scale the shift $h_{\mathcal{F}}$ length to be equal to 6 which is the maximal amplitude while training. Then we can construct a pseudo-labeled dataset for binary classification $\mathcal{D}_{\mathcal{F}} = \{(G(z \pm h_{\mathcal{F}}), \pm 1)\}$ with $z \sim \mathcal{N}(0, I)$. Given this dataset, we train a binary classification model $M_{\mathcal{F}} : G(z) \longrightarrow \{-1, 1\}$ to fit $\mathcal{D}_{\mathcal{F}}$. After that, $M_{\mathcal{F}}$ can induce pseudolabels for the dataset of real images $\mathcal{D}$, which results in pseudo-labeled dataset $\mathcal{D}_{\mathcal{F}}^{real}$=$\{(I, M_{\mathcal{F}}(I)), I \in \mathcal{D}\}$ (see Figure 1). We expect that if the factor of variation $\mathcal{F}$ is responsible for a single and easy-to-interpret attribute, the split of real images $\mathcal{D}_{\mathcal{F}}^{real}$ will be consistent with $\mathcal{D}_{\mathcal{F}}$. Therefore, the pseudo-labels-pretrained classifier is expected to demonstrate high performance on $\mathcal{D}_{\mathcal{F}}$. On the contrary, if the factor of variation is mixed and uninterpretable, we expect that the classifier, trained on $\mathcal{D}_{\mathcal{F}}^{\mathrm{real}}$, will perform poorly on $\mathcal{D}_{\mathcal{F}}$ (see Figure 2). Thus, we re-train the model $M_{\mathcal{F}}$ from scratch on $\mathcal{D}_{\mathcal{F}}^{real}$ and compute its accuracy on $\mathcal{D}_{\mathcal{F}}$. The obtained accuracy value is referred to as DVN.

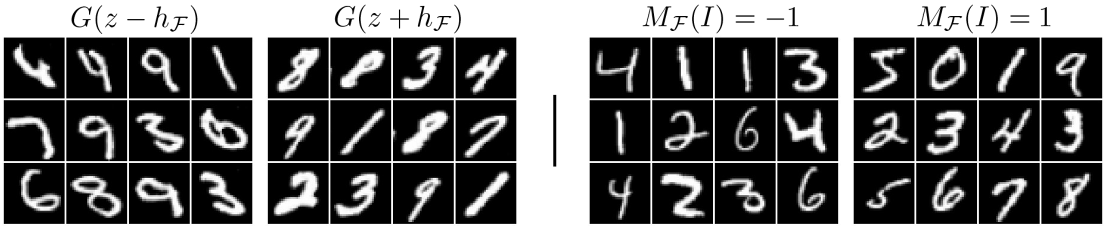

Figure 1. Left: generator samples from $\mathcal{D}_{\mathcal{F}}$ grouped by latent shift direction. *Right*: split of the real MNIST images according to a model, trained on the generated samples split. They form the pseudo-labeled dataset $\mathcal{D}_{\mathcal{F}}^{\mathrm{real}}$

.

In the experiments below, we report the DVN averaged over all directions. Since the directions with higher DVN values are typically more interpretable, we additionally report the average DVN over the top 50 directions ($\mathrm{DVN}_{\mathrm{top}}$). Following the experiment in Section 4, in Table 1 the directions discovered by our method are compared in terms of DVN with random and coordinate directions.

In all the experiments, we use a LeNet-like classification model (see Table 2) with the cross-entropy objective. We train it for both $\mathcal{D}_{\mathcal{F}}$ and $\mathcal{D}_{\mathcal{F}}^{\mathrm{real}}$ for 100 steps of Adam optimizer, as in all our experiments it converges rapidly. We use batch 32 and learning rate 0.001. The sizes of $\mathcal{D}_{\mathcal{F}}$ and $\mathcal{D}_{\mathcal{F}}^{\mathrm{real}}$ always equal 3200 and we did not observe any difference from the usage of more samples.

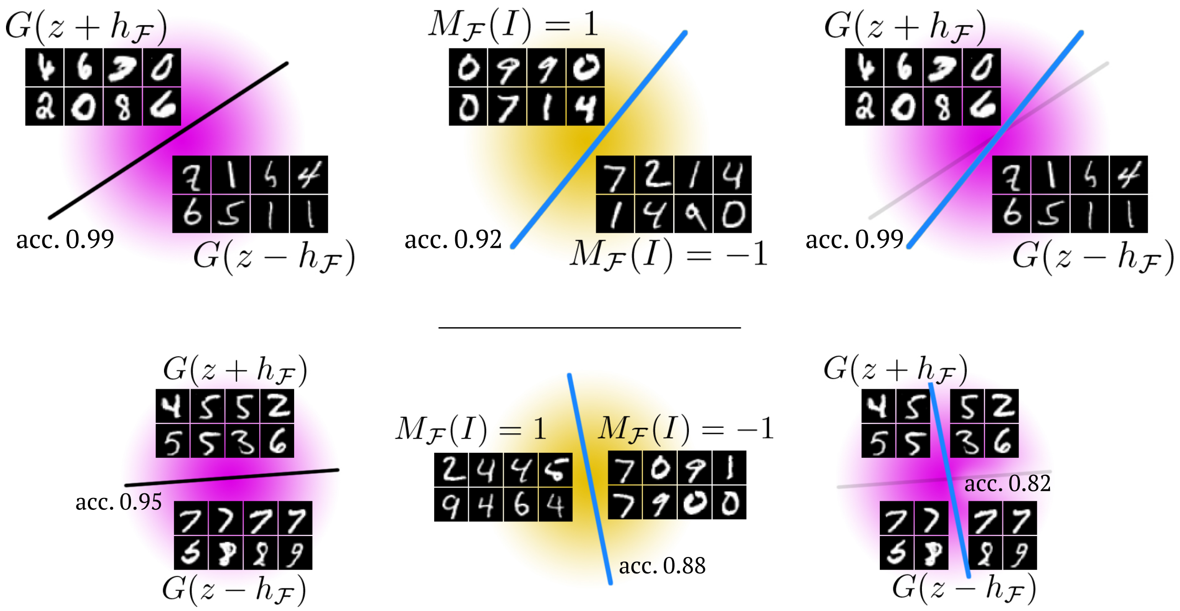

Figure 2. DVN computation process. *Purple*: generated images domain, *yellow*: real images domain. *Top row*: the split of generated images naturally transfers to the real images domain and finally induces almost the same split of the generated images. *Bottom row*: the split of the generated images is difficult to interpret and does not correspond to any natural factor of variation. Therefore, it is hard for a simple classification model to "generalize" to real data, which results in lower DVN values.

Table 1. Quantitave comparison of our method with random and coordinate axes directions in terms of DVN.

Directions	MNIST	Anime	CelebA	ILSVRC
DVN				
Random	0.87	0.81	0.56	0.71
Coordinate	0.87	0.82	0.59	0.65
Ours	**0.95**	**0.84**	**0.66**	**0.71**
DVN$_{\text{top}}$				
Random	0.89	0.92	0.64	0.82
Coordinate	0.89	0.93	0.74	0.78
Ours	**0.99**	**0.93**	**0.82**	**0.83**

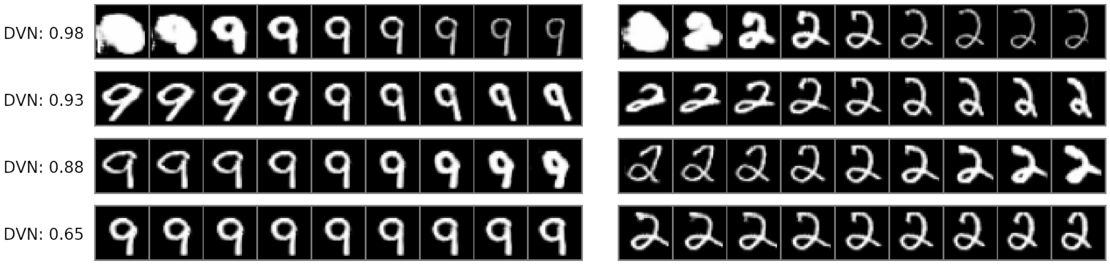

Figure 3. Image variations obtained by moving latent codes along four directions and the corresponding DVN values. The directions with high DVN are easier to interpret.

LeNet-based binary classifier
input: $C \times h \times w$
CONV, kernel: 5×5, channels: 6
BN, RELU
MAXPOOL, kernel: 2×2, stride: 2
CONV, kernel: 5×5, channels: 16
BN, RELU
MAXPOOL, kernel: 2×2, stride: 2
CONV, kernel: 5×5, channels: 120
BN, RELU, AVGPOOL
FC, channels: 84
BN, RELU
FC, channels: 2

Table 2. The binary classification model used for DVN computation.

1.3. Discovered directions uniformity.

As for further analysis, we report the global affect of a latent shift for some of the discovered directions. We calculate the Fréchet Inception Distance (Fulton & Harris, 2013) between the real images and the shifted distribution $\{G(z + h), z \sim \mathcal{N}(0, I)\}$ for different shift magnitudes. We compute FID with ILSVRC test data and 50.000 randomly sampled latent z. Table 3 presents the affect of different shift scales for some of directions from BigGAN latent space. The base model performs with FID = 10.2. Notably, the generated data transformation appears to be more homogeneous for different directions compared to i.e. (Jahanian et al., 2020). Apparently, this happens because we learn these directions simultaneously instead of independently.

shift scale	-12	-6	-3	0	3	6	12
Backgndound removal	16.5	11.8	10.4	10.2	11.3	14.9	27.3
Lighting	13.0	11.0	10.3	10.2	10.6	11.4	15.3
Vertical shift	40.3	15.1	12.7	10.2	11.8	14.3	44.2
Zooming	21.6	12.6	10.9	10.2	10.4	11.8	17.9

Table 3. FID for different shift magnitudes for some of explored latent directions.

1.4. Alternative disentanglement metrics.

In our work, we have introduced three quantitative measures to compare different sets of directions from the same latent space. We do not use the common disentanglement metric from the β-VAE paper (Higgins et al., 2017), since it heavily relies on an additional encoder that can be difficult to obtain for existing GAN models. Moreover, metric from (Higgins et al., 2017) is typically applied for a relatively small number of factors K (up to five) and it is unclear if it is reliable for large K.

1.5. Other details

Interestingly, the discovered directions sometimes behave in an unexpected manner. For instance, we have observed that the BigGAN direction responsible for the background removal simply blanks the images that do not have explicit foreground objects (see Figure 4).

As a final comment, we describe the exact procedure we have followed to find the desired interpretable directions. Once the method proposes K directions, we sort them with respect to the individual DVN values. Then for each direction h_k, we draw the images $G(z + \varepsilon A(h_k))$ varying ε from -9 to 9. We review them manually and highlight the most interesting directions. For instance, for K=128 this procedure takes about ten minutes for a single person.

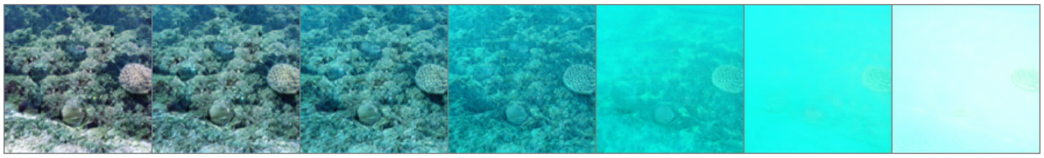

Figure 4. Variation along the background removal direction for the BigGAN generator with class "coral reef". As there seems to be no foreground, the model blanks the whole image

References

Fulton, W. and Harris, J. *Representation theory: a first course*, volume 129. Springer Science & Business Media, 2013.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.

Jahanian, A., Chai, L., and Isola, P. On the"steerability" of generative adversarial networks. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.