Robust Deep Interpretable Features for Binary Image Classification

Robert Hu[1] and Dino Sejdinovic[1]

[1]University of Oxford

Abstract

The problem of interpretability for binary image classification is considered through the lens of kernel two-sample tests and generative modeling. A feature extraction framework coined Deep Interpretable Features is developed, which is used in combination with IntroVAE, a generative model capable of high-resolution image synthesis. Experimental results on a variety of datasets, including COVID-19 chest x-rays demonstrate the benefits of combining deep generative models with the ideas from kernel-based hypothesis testing in moving towards more robust interpretable deep generative models.

1 Introduction

While Machine Learning (ML) has enjoyed tremendous growth in both academia and industry, it has also caused concern in regards to its transparency and interpretability [5, 23] in high-stakes decision making in areas such as medicine and engineering. In this paper, we consider interpretability for binary image classification and focus on extracting interpretable features that affect the classification boundary. In this context, convolutional neural networks (CNN) [2] are often used as they exhibit superior performance. However, due to their complexity, they are often considered black-box models [12, 8]. In contrast, simple models such as logistic classification applied directly to the pixel representation would give an interpretation in terms of pixel-wise effects, but these are not meaningful for the overall classification task. A compromise between CNNs and linear models (see Figure 1) would be to consider lower-dimensional latent representations for each image, where each

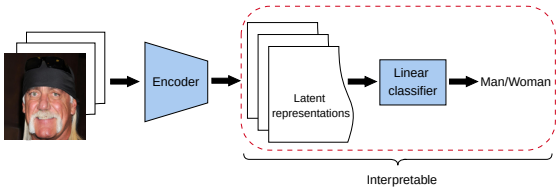

Figure 1: Logistic classifier applied to latent representations

element in the latent representations corresponds to an extracted feature that is interpretable. Such latent representations can be recovered by using deep generative models, in particular when the latent representations are linearly separated on class. To impose such a structure on the latent space, one can add a regularization objective to the loss function of the generative model. In that scenario, we can extract these features using a linear model and visualize them using the generative component of our model. For the remainder of this paper, we will consider features to be computable quantities given any input and our goal is to discover such quantities that are class-discerning. Our motivation for building such an interpretable framework is to extend the existing post-hoc methods [22, 18] in the deep learning setting by providing a more thorough analysis of latent features and exploring deep generative models as fundamentally interpretable models[3, 9]. We consider a supervised approach when finding latent features, as unsupervised disentanglement methods are demonstrated to be incapable of recovering these features[16]. A supervised method has the added benefit of imposing structure on the latent space, potentially improving the robustness of the generative model to adversarial attacks [6].

https://doi.org/10.7557/18.5708

We pose the following questions to guide us:

1. How can we find linearly separated latent representations?
2. What implications on performance do such latent representations have?
3. How can we interpret a model using linearly separated latent representations?

The rest of the paper is organized as follows. We start by providing desiderata for a generative model that can be used for our purposes, leading to the choice of Variational AutoEncoders (VAE) and its variants. We then present our main contribution: the framework termed Deep Interpretable Features (DIF) and its key components. The following sections describe our experimental setup and evaluation criteria, present empirical results and ablation studies, followed by concluding remarks.

2 Background: Generative Modeling

We present desiderata based on our application needs and provide examples and motivation for feasible and unfeasible generative model choices.

Desiderata. *Consider generative models* $G : \mathbf{z} \to \mathbf{x}$ *and the latent representations* $\mathbf{z}$ *of images* $\mathbf{x}$ *in the context of interpretability:*

1. $\mathbf{z}$ *is a low dimensional representation of* $\mathbf{x}$.
2. *Semantically dissimilar images are encoded into semantically dissimilar latent representations* $\mathbf{z}$.
3. *Semantically dissimilar latent representations are decoded back into semantically dissimilar images through* G.
4. *All latent representations have a visually meaningful map* $G(\mathbf{z})$.

If desideratum 1 does not hold, $G(\mathbf{z})$ fails to generalize pixels into features. If desideratum 2 does not hold, distinguishing images on a feature-wise basis will fail. If desideratum 3 or 4 fail, then features found in latent space would not bear any visual meaning when decoded back. We briefly review the generative model literature in the context of our desiderata below.

VAE [13] based models are generally the most feasible, as they have an encoder component that can be trained to directly map images to latent representations. In particular they encourage desideratum 3 by training on the reconstruction error objective $\mathcal{L}_{\text{recon}}(\mathbf{x}, G(\mathbf{z})) = \|G(\mathbf{z}) - \mathbf{x}\|^2$. The only shortcoming of VAEs is the quality of the synthesized images, which many different variations address [10, 19].

Generative Adversarial Network (GAN) [7] based models work well due to the adversarial training procedure which encourages desideratum 4. However, they become computationally infeasible; as GANs do not have an encoder, one needs to reverse optimize [15] a GAN generator to obtain the latent representation of each image.

Normalizing flows [21] are not feasible since they do not yield low dimensional latent representations due to their bijection property.

We conclude that VAE-based models are the most suitable choice wherein the problem lies in finding a suitable variation capable of achieving our aims.

3 Main Result: Deep Interpretable Features (DIF)

We propose our framework Deep Interpretable Features (DIF), which at its core gives any generative model the three following properties:

1. Encourages the generative model desiderata through a training objective
2. Finds interpretable features

As the majority of the technical contribution lies in property 1, we begin by reviewing the necessary literature in kernel two-sample testing.

3.1 The Mean Embedding test

Non-parametric hypothesis testing considers whether two independently and identically drawn samples $\mathbb{X} = \{\mathbf{x}_i\}_{i=1}^{n_1}, \mathbb{Y} = \{\mathbf{y}_i\}_{i=1}^{n_2} \in \mathbb{R}^d$ are drawn from different distributions P and Q based on observed samples $\mathbb{X} \sim P$ and $\mathbb{Y} \sim Q$. The null hypothesis $H_0 : P = Q$ is tested vs. the general alternative $H_1 : P \neq Q$. We focus on the Mean Embedding test (ME test) of [11],

which introduces a Hotelling's T-squared statistic computed using a kernel mean embedding expressed as $\hat{\lambda}_n = n\overline{\mathbf{h}}_n^\top \mathbf{S}_n^{-1}\overline{\mathbf{h}}_n$ where $\overline{\mathbf{h}}_n = \frac{1}{n}\sum_{i=1}^n \mathbf{h}_i$, $\mathbf{h}_i = (k(\mathbf{x}_i,\mathbf{v}_j) - k(\mathbf{y}_i,\mathbf{v}_j))_{j=1}^J \in \mathbb{R}^J$, $\mathbf{S}_n = \frac{1}{n-1}\sum_{i=1}^n \left(\mathbf{h}_i - \overline{\mathbf{h}}_n\right)\left(\mathbf{h}_i - \overline{\mathbf{h}}_n\right)^\top$, for n observations.

The statistic depends on a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (with $\mathcal{X} \subseteq \mathbb{R}^d$), and a set of J prototypes $\mathcal{V} = \{\mathbf{v}_j\}_{j=1}^J \subset \mathbb{R}^d$. Under H_0, $\hat{\lambda}_n$ asymptotically follows $\chi^2(J)$, a chi-squared distribution with J degrees of freedom. The ME-test is executed by calculating $\hat{\lambda}_n$ and rejects H_0 if $\hat{\lambda}_n > T_\alpha$, where the test threshold T_α is given by the $(1-\alpha)$-quantile of the asymptotic null distribution $\chi^2(J)$ with α denoting the significance of the test. When computing $\hat{\lambda}_n$ we modify the statistic with a regularization parameter $\gamma_n > 0$, giving $\hat{\lambda}_n = n\overline{\mathbf{h}}_n^\top \left(\mathbf{S}_n + \gamma_n \mathbf{I}\right)^{-1}\overline{\mathbf{h}}_n$, for numerical stability. We use a slightly modified version of the ME-test statistic proposed in [11] since our application considers $n_1 \neq n_2$. For different sample sizes we use the pooled covariance version of the ME-test statistic defined as

$$\begin{aligned}
\hat{\lambda}_{n_1,n_2} &= \frac{n_1 n_2 \overline{\mathbf{h}}_{n_1,n_2}^\top \mathbf{S}_{n_1,n_2}^{-1}\overline{\mathbf{h}}_{n_1,n_2}}{n_1+n_2}, \\
\mathbf{S}_{n_1,n_2} &= \frac{(n_1-1)\mathbf{S}_{n_1} + (n_2-1)\mathbf{S}_{n_2}}{n_1+n_2-2}, \\
\overline{\mathbf{h}}_{n_1,n_2} &= \\
&\left(\frac{1}{n_1}\sum_{i=1}^{n_1} k(\mathbf{x}_i,\mathbf{v}_j) - \frac{1}{n_2}\sum_{k=1}^{n_2} k(\mathbf{y}_k,\mathbf{v}_j)\right)_{j=1}^J \in \mathbb{R}^J.
\end{aligned} \tag{1}$$

We are now ready to present our first technique in linearly separating the latent space.

3.2 Generative modeling objective: Mean Embedding-objective

In principle, the VAE encourages all desiderata except desideratum 2, which establishes the need of a correction. To encourage desideratum 2 we introduce the ME-objective for generative models. We will denote a mapping parametrized by θ as $T_\theta(\cdot)$, labeled objects we wish to separate as $\zeta_i^{\mathbb{X}}, \zeta_j^{\mathbb{Y}}$ and batches of latent representations $\mathbf{b}_{\mathbb{X}} = \{\mathbf{z}_i^{\mathbb{X}}\}_{i=1}^{b_1}$, $\mathbf{b}_{\mathbb{Y}} = \{\mathbf{z}_j^{\mathbb{Y}}\}_{j=1}^{b_2}$ with $\mathbf{z}_i^{\mathbb{X}} = C\tanh\left(\frac{T_\theta(\zeta_i^{\mathbb{X}})}{C}\right)$, $\mathbf{z}_j^{\mathbb{Y}} = C\tanh\left(\frac{T_\theta(\zeta_j^{\mathbb{Y}})}{C}\right)$ and labeled prototypes $\mathbf{v}_{\mathbb{X}} \subset \mathbf{b}_{\mathbb{X}}$, $\mathbf{v}_{\mathbb{Y}} \subset \mathbf{b}_{\mathbb{Y}}$. The ME-objective $\Lambda(\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}, \mathcal{V})$ is calculated as

$$\begin{aligned}
&\Lambda(\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}, \mathcal{V}) = \\
&\frac{b_1 b_2 \overline{\mathbf{h}}(\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}, \mathcal{V})^\top (\mathbf{S}_{b_1,b_2} + \gamma_{b_1,b_2}\mathbf{I})^{-1}\overline{\mathbf{h}}(\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}, \mathcal{V})}{b_1+b_2}, \\
&\overline{\mathbf{h}}(\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}, \mathcal{V}) = \\
&\left(\frac{1}{b_1}\sum_{i=1}^{b_1} k(\mathbf{b}_{\mathbb{X}}, \mathbf{v}_j) - \frac{1}{b_2}\sum_{j=1}^{b_2} k(\mathbf{b}_{\mathbb{Y}}, \mathbf{v}_j)\right)_{j=1}^J \in \mathbb{R}^J \\
&\mathbf{v}_j \in \mathcal{V} = \{\mathbf{v}_{\mathbb{X}}; \mathbf{v}_{\mathbb{Y}}\}.
\end{aligned} \tag{2}$$

The overall idea is to use T_θ to map $\zeta^{\mathbb{X}}$ and $\zeta^{\mathbb{Y}}$ away from each other such that $\mathbf{z}^{\mathbb{X}}$ and $\mathbf{z}^{\mathbb{Y}}$ are separated in latent space as shown in Figure 2.

Intuitively, we calculate the ME-test statistic on batches of latent representations with randomly subsampled prototypes for each label and train T_θ to "pull" latent representations away from each other by maximizing the ME-test statistic as an objective. By maximizing $\Lambda(\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}, \mathcal{V})$ with respect to $\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}$ we are maximizing the statistical difference between $\mathbf{z}^{\mathbb{X}}$ and $\mathbf{z}^{\mathbb{Y}}$ resulting in T_θ mapping apart $\zeta^{\mathbb{X}}$ and $\zeta^{\mathbb{Y}}$. T_θ is updated by first calculating $\Lambda(\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}, \mathcal{V})$ using subsampled prototypes $\mathbf{v}_{\mathbb{X}}, \mathbf{v}_{\mathbb{Y}}$ from $\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}$ and then taking the gradient of $\Lambda(\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}, \mathcal{V})$ with respect to θ, detailed in Figure 2. We train T_θ on all objectives related to the generative model, denoted $\mathcal{L}_G$, to ensure that $\mathbf{z}^{\mathbb{X}}$ and $\mathbf{z}^{\mathbb{Y}}$ are separated under the condition that they still retain a meaningful representation to the generator G. The loss for T_θ is then

$$\mathcal{L}_{T_\theta}(\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}, \mathcal{X}) = \mathcal{L}_G(\mathcal{X}) + \eta_\Lambda \Lambda(\mathbf{b}'_{\mathbb{X}}, \mathbf{b}'_{\mathbb{Y}}, \mathcal{V}) \tag{3}$$

where $\mathcal{X}$ are any input required of the generative model objective and $\eta_\Lambda < 0$ is a hyperparameter controlling the effect of the ME-objective. In a VAE context, T_θ can be the encoder or a mapping between the encoder and decoder.

To prevent unbounded $\Lambda(\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}, \mathcal{V})$, the domain of $\mathbf{z}$ is bounded to $[-C, C]^d, C \in \mathbb{R}$ by applying a $C\tanh(\frac{\cdot}{C})$ transform to $T_\theta(\zeta)$.

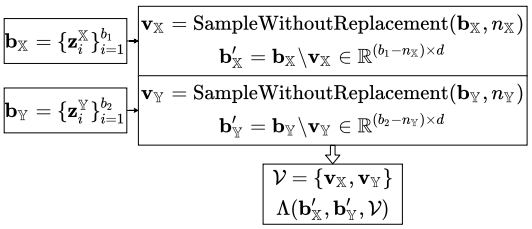

Figure 2: Calculating the ME-objective

3.2.1 Choice of generative model: IntroVAE

We apply the ME-objective to the VAE variation IntroVAE[10], which is a VAE with an additional modified GAN objective. IntroVAE has demonstrated to offer high quality image synthesis on larger images coupled with a minimalistic VAE architecture. This was favored over [19, 20] which either offered too complicated setups or convoluted latent representations. IntroVAE extends the training procedure of VAEs by training the encoder and generator on

$$\begin{aligned} &\mathcal{L}_E(\mathbf{x}, \mathbf{z}) = \text{ELBO}\left(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}\right) \\ &+ \left[m - D_{KL}\left(q_{\boldsymbol{\phi}}\left(\mathbf{z} \mid \mathbf{x}\right) \| p(\mathbf{z})\right)\right]^+ \\ &\mathcal{L}_G(\mathbf{x}, \mathbf{z}) = D_{KL}\left(q_{\boldsymbol{\phi}}\left(\mathbf{z} \mid \mathbf{x}\right) \| p(\mathbf{z})\right) \\ &+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x} \mid \mathbf{z}\right)\right] \end{aligned} \tag{4}$$

where $[m - x]^+ = \text{ReLU}(0, m - x)$ serves as a saddle point objective for the encoder p_θ and decoder $q_{\boldsymbol{\phi}}$ using the KL-divergence between approximative posterior and prior of $\mathbf{z}$ $D_{KL}\left(q_{\boldsymbol{\phi}}\left(\mathbf{z} \mid \mathbf{x}\right) \| p(\mathbf{z})\right)$. DIF is then applied by adding the ME-objective to the encoder loss:

$$\mathcal{L}(\mathbf{x}, \mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}) = \mathcal{L}_E(\mathbf{x}, \mathbf{z}) + \eta_\Lambda \Lambda(\mathbf{b}'_{\mathbb{X}}, \mathbf{b}'_{\mathbb{Y}}, \mathcal{V}). \tag{5}$$

It should be noted that we take T_θ to be the encoder and consequently ζ's are taken to be images. When training DIF, we use both losses specified in eq. (3) and eq. (5).

3.3 Isolating Features

To isolate features, we take our binary classifier $f(\mathbf{z})$ to be a L^1 regularized logistic regression classifier. We can then exploit the shrinkage properties of the L^1 regularization to isolate the most important features. We then select features by their magnitude $|w_i|$, where w_i is the i:th element in the linear weights $\mathbf{w} = [w_1, \ldots, w_n]$.

We linearly separate the latent representations by choosing a linear kernel $k(\mathbf{x}, \mathbf{y}) = \nu \mathbf{x}^\top \mathbf{y}$ with hyperparameter $\nu > 0$ for our ME-objective. With $\mathbf{z}$ bounded by a $\tanh(\cdot)$ transform, it follows that $\Lambda(\mathbf{b}_{\mathbb{X}}, \mathbf{b}_{\mathbb{Y}}, \mathcal{V})$ with a linear kernel also is bounded [11].

4 Experiments

4.1 Data

We conduct our experiments on CelebHQ (separating females with 18943 samples and males with 11057 samples, $256 \times 256 \times 3$), COVID-19[1] chest x-rays (separating healthy with 1602 samples and COVID-19 infected with 434 samples, $256 \times 256 \times 1$) and MNIST (separating 3s with 7141 samples and 8s with 6825 samples, upsampled to $64 \times 64 \times 1$).

All datasets are divided into training (90%) and testing (10%) sets. We use the same architecture for all generative models and their respective components.

4.2 Evaluation

We evaluate our experiments in the following way on test data:

1. **Generative model performance**: We estimate the log-likelihood and calculate the ELBO on test data together with FID scores [1] for fake samples.
2. **Latent traversals**: We traverse from an image in P to an image in Q and judge if the transition occurs smoothly without artifacts. This evaluation is done to ensure the generative model does not overfit.
3. **Predictive performance** We compare our L^1 regularized classifiers to a CNN classifier (using the identical architecture as the encoder) fitted directly on images and measure performance in Area Under the Curve (AUC).

[1] **Disclaimer**: The results presented for the COVID-19 x-ray data are for illustration purposes and further work is needed to draw any medically relevant conclusions based on them.

4. **Isolating features**: We judge the extracted features qualitatively on how distinct the features are and how the latent transitions look. We analyze the 3 largest features w_i by magnitude $|w_i|$ for a fix λ_{L^1}. Strong classification performance for a large λ_{L^1} indicates validity of the isolated features. We also report % of non sparse features, which is calculated as $\frac{\text{number of} w_i \geq 0.1 \max |w|}{\text{total features}}$.

4.3 Benchmarks

Our benchmarks include vanilla IntroVAE, as well as a method inspired by [17], detailed below.

A logistic separation objective As an ablation study in order to demonstrate importance of using an ME-based objective, we also consider an objective akin to logistic regression applied to the latent space. Namely, instead of using ME, we take Λ to be a logistic regression objective (i.e. binary cross-entropy loss) $\mathcal{L}(w^\top C \tanh\left(\frac{T_\theta(\zeta)}{C}\right), y_{P/Q})$ for a linear decision boundary $w^\top C \tanh\left(\frac{T_\theta(\zeta)}{C}\right)$, where we keep the weight vector fixed $w = \mathbf{1}_{1\times d}$ but optimize the encoder parameters θ. Here, $y_{P/Q} \in \{0, 1\}$ is the class indicator for ζ. We want T_θ to map $\{\zeta_i^{\mathbb{X}}\}_{i=1}^{n_1}$ and $\{\zeta_j^{\mathbb{Y}}\}_{j=1}^{n_2}$ to their respective sides of a constant hyperplane w and thus minimize this objective. In contrast to the ME-objective, this method imposes the desired structure directly through a fix hyperplane rather than "pulling" latent representations apart through a kernel.

5 Results

We present the quantitative comparisons in Table 1. As classification performance tends to drop when $\lambda_{L^1} = 0.1$ for vanilla IntroVAE, we investigate sparsity levels at this value.

Image Traversals Smooth artifact-free image traversals between latent representations in Figure 3 indicate that no model is overfitting.

Isolating Features We present isolated features using L^1 regularized logistic classification in Figure 4. DIF finds features that seemingly controls

Figure 3: Image traversals for samples from P to samples from Q.

facial hair, the width of numbers respectively, and non-organ matter (non-heart grey matter) in the lungs. These changes occur in an isolated fashion compared to the other methods where one element seemingly affect multiple visual features. In particular, we find that the non-organ matter is consistent with medical results reported in [4, 14].

Figure 4: Comparison of isolated features for all three data sets. We traverse each feature between (-C,C), the interval of our tanh transform.

We present the test accuracy of a L^1-regularized classification trained on the latent representations for different regularization values $\lambda_{L^1} \in \{0.0, 0.001, 0.01, 0.1\}$ in Table 1. We find that lasso regression surprisingly outperforms a CNN for CelebHQ and the COVID-19 dataset for both DIF and the logistic benchmark, where DIF yields the most gains in classification performance. Models trained with DIF also retain performance for sparser classifiers. As the percentage of non-sparse features are higher for DIF, this suggests that DIF finds more meaningful and robust features that dis-

dataset-model	fake-FID (n=32)	log-likelihood	ELBO	$\lambda_{L^1} = 0$	$\lambda_{L^1} = 0.001$	$\lambda_{L^1} = 0.01$	$\lambda_{L^1} = 0.1$	$\lambda_{L^1} = 0.1$ %Non sparse	CNN
CelebHQ-DIF	**103.64±4.177**	**-2239.154±0.593**	**-2422.622±1.508**	**0.983±0.002**	0.981±0.001	0.98±0.001	0.97±0.004	9.4%±4.9%	0.967±0.006
CelebHQ-Vanilla	107.184±3.612	-2301.921±0.903	-2512.501±3.796	0.948±0.004	0.945±0.002	0.92±0.003	0.744±0.015	2.7%±0.2%	0.967±0.006
CelebHQ-logistic	114.514±6.227	-2299.671±0.834	-2471.561±1.473	0.967±0.002	0.97±0.001	0.94±0.003	0.769±0.004	2.9%±0.0%	0.967±0.006
MNIST-DIF	**87.112±2.535**	-71.487±0.03	-74.486±0.047	0.992±0.001	0.992±0.001	0.989±0.002	0.939±0.001	29.2%±3.6%	**0.994±0.007**
MNIST-Vanilla	92.574±0.89	-72.077±0.038	-75.936±0.078	0.989±0.002	0.988±0.002	0.989±0.002	0.917±0.003	37.5%±0.0%	**0.994±0.007**
MNIST-logistic	88.894±8.272	**-64.5±0.01**	**-69.809±0.068**	0.989±0.001	0.987±0.003	0.989±0.002	0.909±0.002	43.8%±0.0%	**0.994±0.007**
Covid-DIF	**214.266±1.426**	-929.991±1.596	-1122.549±10.712	**0.995±0.005**	0.989±0.003	0.979±0.01	0.952±0.01	10.0%±12.9%	0.983±0.009
Covid-Vanilla	253.288±2.564	-761.591±0.151	-891.08±2.046	0.977±0.004	0.968±0.005	0.968±0.021	0.888±0.025	2.3%±0.2%	0.983±0.009
Covid-logistic	247.358±7.338	**-652.715±0.131**	**-759.155±2.906**	0.874±0.027	0.852±0.012	0.916±0.013	0.937±0.0	1.1%±0.2%	0.983±0.009

Table 1: All classification results in test data AUC, averaged over 5 runs. Sparser models do better with DIF representation, which implies that DIF indeed imposes a linearly separable latent space. DIF generally seems to improve image quality by the lower FID-scores reported for fake images. The improved image generation quality is potentially caused by the structure imposed by DIF in latent space.

criminate between P and Q.

6 Analysis

Does DIF achieve its aim of separation in latent space? We see that DIF achieves a very good separation on the CelebHQ dataset in Figure 5. On MNIST, a separation objective is not needed as the latent representations are naturally separated by looking at the latent space of the Vanilla model.

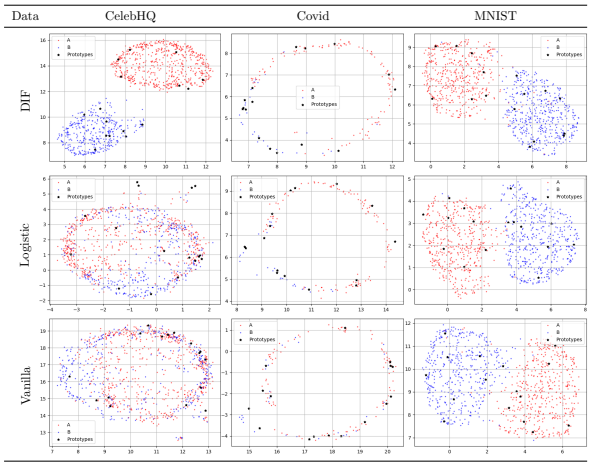

Figure 5: UMAP visualization of $\mathbf{z}$ for test data. Red points belong to P, blue points to Q and black points denote prototypes.

Is DIF more robust to adversarial attacks? As DIF correctly imposes a linearly separable structure on the latent space, we hypothesize this improves resilience towards adversarial attacks and report this in Figure 6. We use the adversarial attacks proposed in [6] to test the robustness.

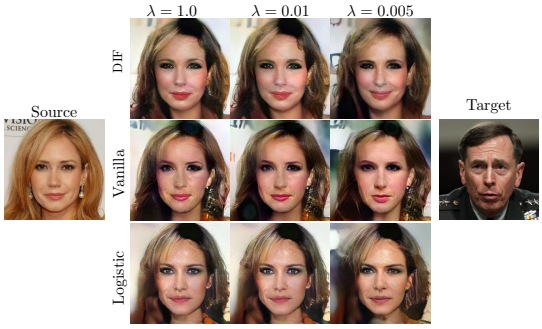

Figure 6: λ denotes the strength of the adversarial attack. A smaller lambda implies a stronger attack. Both DIF and the Logistic benchmarks exhibits robustness towards adversarial attacks, while the Vanilla version starts to exhibit masculine features at $\lambda = 0.005$.

7 Conclusion

We combined deep generative modeling with a supervised objective based on kernel-based two-sample testing (ME objective) to encourage the separation between the classes in the latent space. As a result, individual latent dimensions are more interpretable and capture class-discerning image features. Latent traversals for DIF between images of different classes exhibit smooth transitions, indicating a smooth latent space. We note that a linear classifier can then be applied to the resulting latent representation, in some cases resulting in the performance superior to CNNs on test data. Further, imposing a linear structure on the latent space improves robustness against adversarial attacks.

References

[1] S. Barratt and R. Sharma. A note on the inception score. *ArXiv*, abs/1801.01973, 2018.

[2] Y. Bengio and Y. LeCun. Convolutional networks for images, speech, and time-series. 11 1997.

[3] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, and C. Rudin. This looks like that: Deep learning for interpretable image recognition, 2018.

[4] D. Cozzi, M. Albanesi, E. Cavigli, C. Moroni, A. Bindi, S. Luvarà, S. Lucarini, S. Busoni, L. N. Mazzoni, and V. Miele. Chest x-ray in new coronavirus disease 2019 (covid-19) infection: findings and correlation with clinical outcome. *La radiologia medica*, 125(8):730–737, Aug 2020.

[5] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining explanations: An overview of interpretability of machine learning, 2018.

[6] G. Gondim-Ribeiro, P. Tabacof, and E. Valle. Adversarial attacks on variational autoencoders, 2018.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

[8] A. Gosiewska and P. Biecek. Lifting interpretability-performance trade-off via automated feature engineering, 2020.

[9] P. Hase, C. Chen, O. Li, and C. Rudin. Interpretable image recognition with hierarchical prototypes, 2019.

[10] H. Huang, Z. Li, R. He, Z. Sun, and T. Tan. Introvae: Introspective variational autoencoders for photographic image synthesis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 52–63, Red Hook, NY, USA, 2018. Curran Associates Inc.

[11] W. Jitkrittum, Z. Szabo, K. Chwialkowski, and A. Gretton. Interpretable Distribution Features with Maximum Testing Power. 5 2016.

[12] U. Johansson, C. Sönströd, U. Norinder, and H. Boström. Trade-off between accuracy and interpretability for predictive in silico modeling. *Future medicinal chemistry*, 3:647–63, 04 2011.

[13] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. 12 2013.

[14] W. Kong and P. Agarwal. Chest imaging appearance of covid-19 infection. *Radiology: Cardiothoracic Imaging*, 2(1):e200028, 2020.

[15] Z. C. Lipton and S. Tripathi. Precise recovery of latent vectors from generative adversarial networks, 2017.

[16] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations, 2018.

[17] D. Lopez-Paz and M. Oquab. Revisiting classifier two-sample tests, 2016.

[18] S. Lundberg and S.-I. Lee. A Unified Approach to Interpreting Model Predictions. 5 2017.

[19] S. Pidhorskyi, D. Adjeroh, and G. Doretto. Adversarial latent autoencoders, 2020.

[20] A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2, 2019.

[21] D. J. Rezende and S. Mohamed. Variational Inference with Normalizing Flows. 5 2015.

[22] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?". In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 1135–1144, New York, New York, USA, 2016. ACM Press.

[23] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, May 2019.