Latent Bernoulli Autoencoder

ICML 2020

Jiri Fajtl[1], Vasileios Argyriou[1],
Dorothy Monekosso[2] and Paolo Remagnino[1]

[1]Kingston University, London, UK
[2]Leeds Beckett University, Leeds, UK

August 15, 2020

Motivation

Questions:

- Can we realize a deterministic autoencoder to learn discrete latent space with a competitive performance?
- How to sample from latent space?
- How to interpolate between given samples in this latent space?
- Can we modify sample attributes in the latent space and how?
- What are the simplest possible solutions to the above?

Why discrete representations?

- Gating, hard attention, memory addressing
- Compact representation for storage, compression
- Encoding for energy models such as Hopfield memory[1] or HTM[2]
- Interpretability

Latent Bernoulli Autoencoder LBAE

- We propose a simple, deterministic encoder-decoder model that learns multivariate Bernoulli distribution in the latent space by binarization of continuous activations
- For N-dimensional latent space the information bottleneck of a typical autoencoder is in LBAE replaced with tanh() followed by binarization $f_b() \in \{-1, 1\}^N$ with unit gradient surrogate function $f_s()$ for backward pass

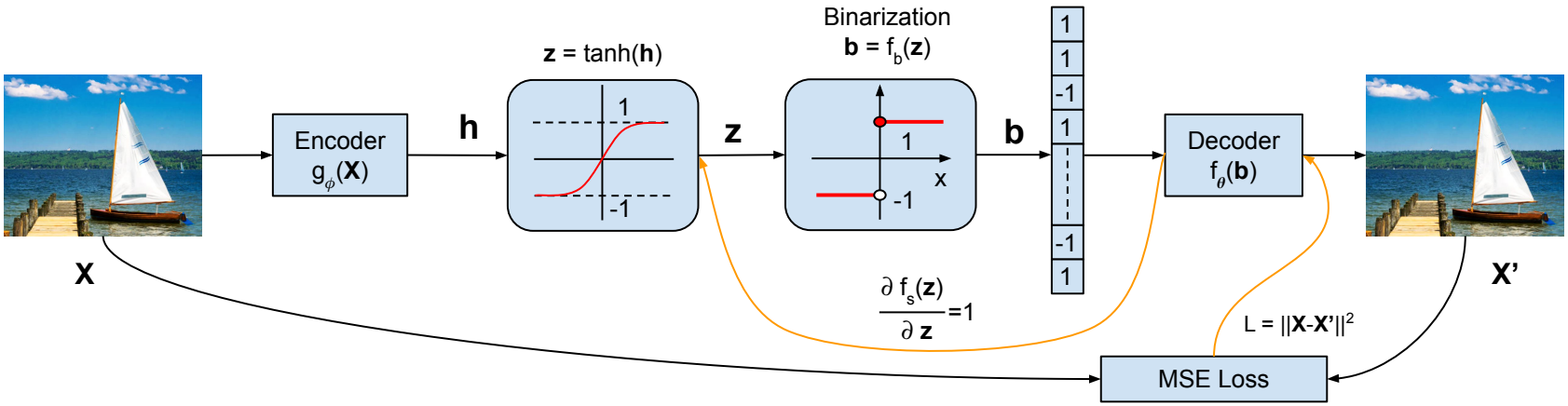
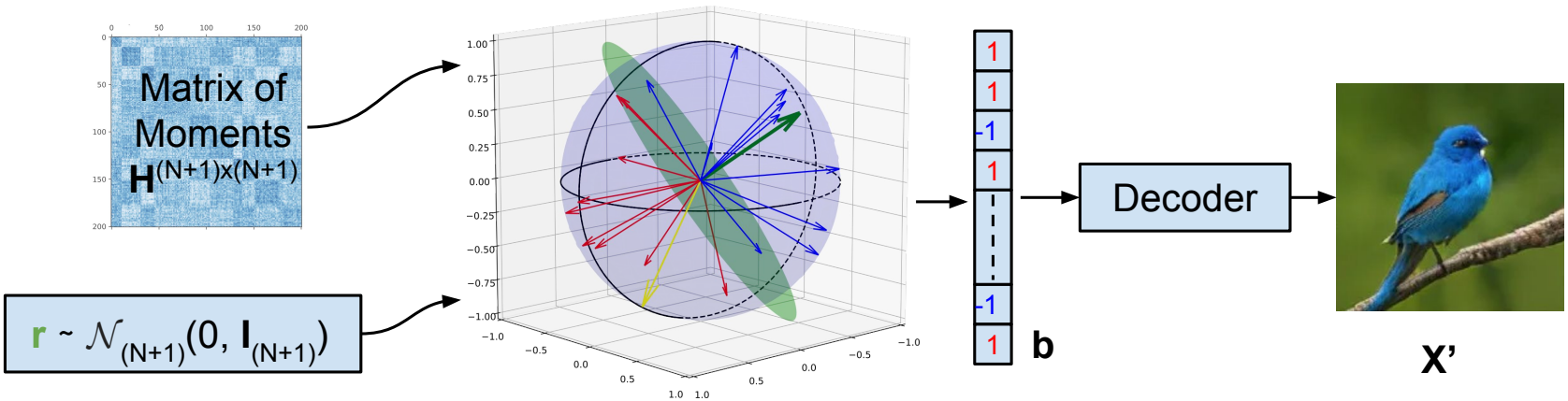


Figure: Black forward pass, yellow backward pass

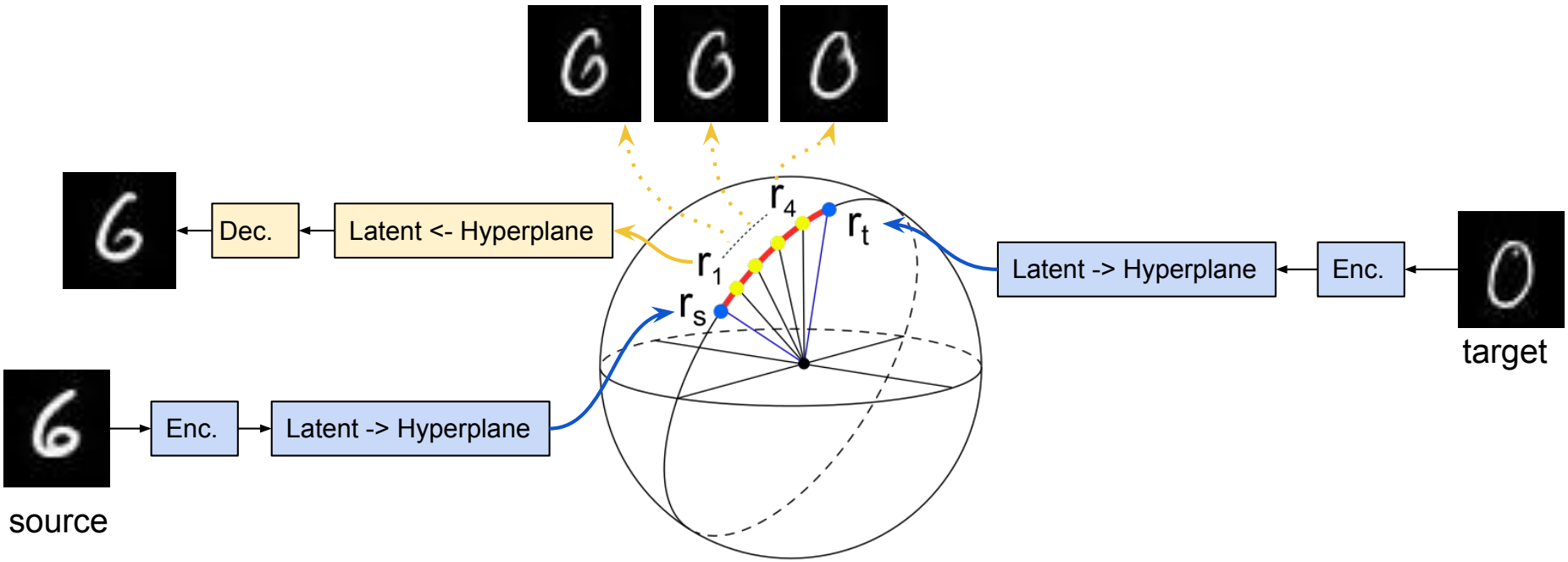
Sampling From the Bernoulli Distribution

- Without enforcing any prior on the latent space the learned distribution is unknown
- We parametrize the distribution by its first two moments learned from latents encoded on the training data
- Dimensions of the binary latent space are relaxed into vectors on a unit hypersphere given the first two moments
- A random Bernoulli vector with the distribution of the latent space is generated by randomly splitting the hypersphere and assigning logical ones to latent dimensions represented by vectors in one hemisphere and zeros to the rest (encoded as $\{-1, 1\}$)



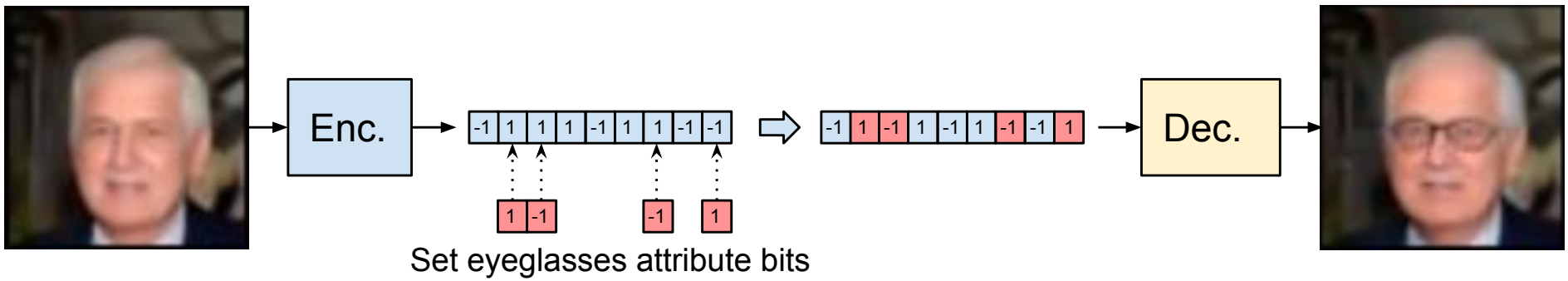
Interpolation in Latent Space

- Given latent representations of two images, generate latents producing interpolation in the image space
- For source and target latents we find hyperplanes on the hypersphere
- Divide the angle between source and target hyperplane normals into T steps and for each produce a new hyperplane
- Decode these hyperplanes into latents and then to images



Changing Attributes

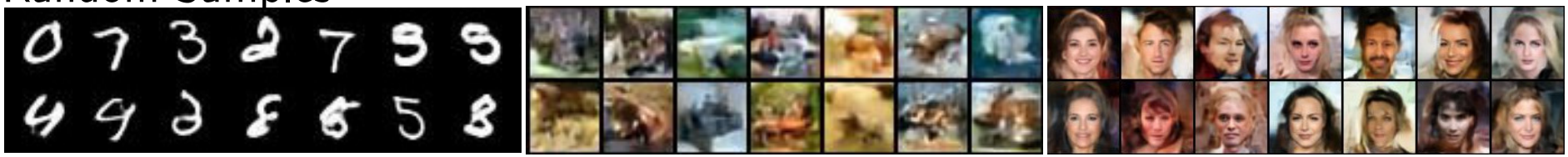
- Statistically significant attributes of the training data can be identified in the latent space e.g. images of faces with eyeglasses
- No need to train the LBAE in a conditional setting
- Collect latents of samples with the given attribute and find highly positively and negatively correlated latent bits
- The attribute is then modified by changing these bits in the latent vector



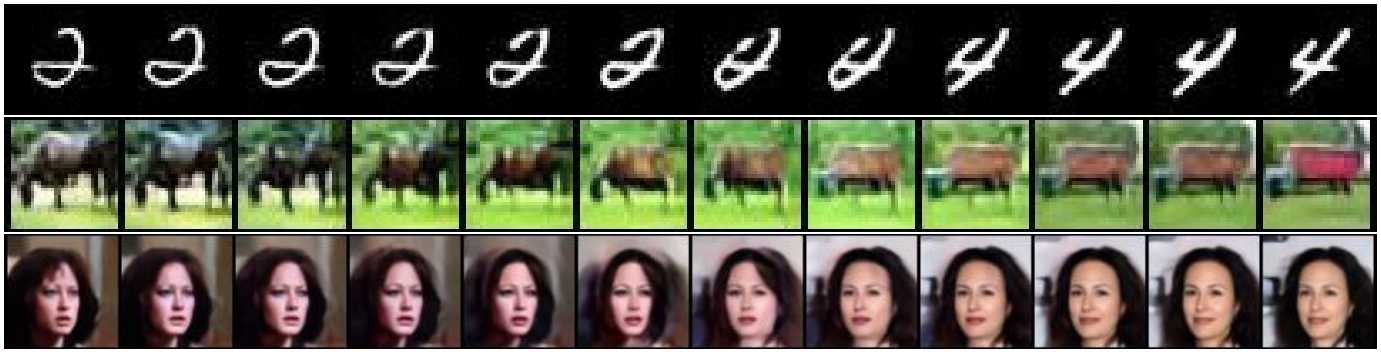
Results

Reconstruction on test datasets

Random Samples



Interpolation on test datasets



Adding *eyeglasses* and *goatee* CelebA attributes on test dataset

Quantitative Results at the end of the presentation

Deep Dive

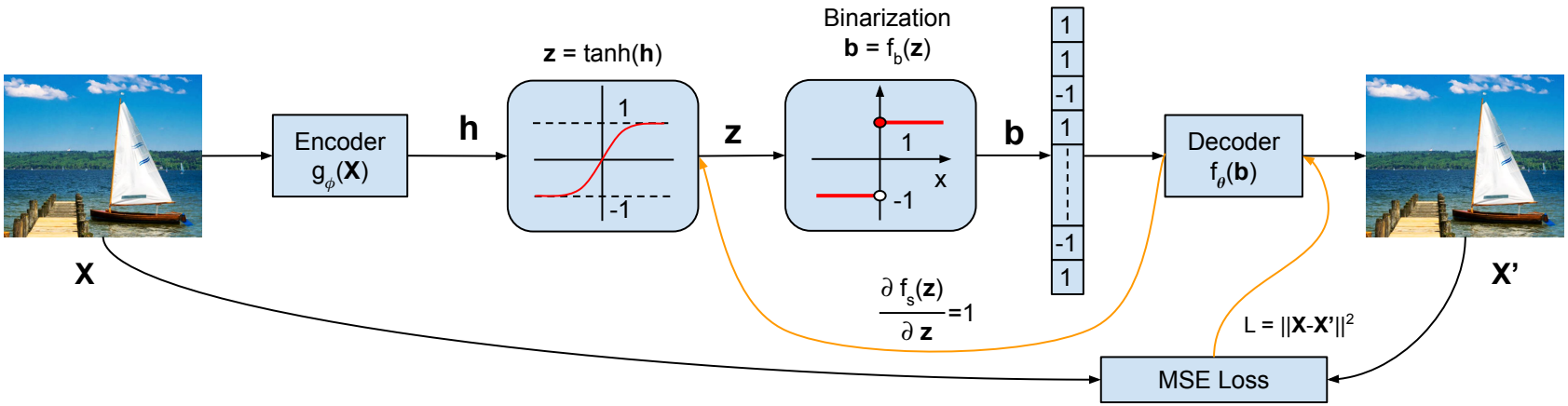
- Learning Bernoulli latent space
- Sampling correlated multivariate Bernoulli latents
- Interpolation in latent space
- Changing sample attributes
- Quantitative & qualitative results
- Conclusion

Learning Bernoulli Latent Space

- Problematic with grandient based methods , not differentiable - no backprop
- Leave non differentiable binarization fcn in the forward pass and bypass it during backprop. Proposed earlier by Hinton & Bengio.
- But the convergence is slow or impossible without limiting the magnitude of the error gradient in the encoder
- Limiting the activation to $[-1, 1]$ with tanh() alleviates this issue

Learning Bernoulli Latent Space

- For N-dimensional latent space we replace the information bottleneck of a typical autoencoder with tanh() followed by binarization $f_b(z_i) = \{1, \text{if } z_i \geq 0 \text{ and } -1 \text{ otherwise}\}$ with unit gradient surrogate function $f_s()$ for backward pass



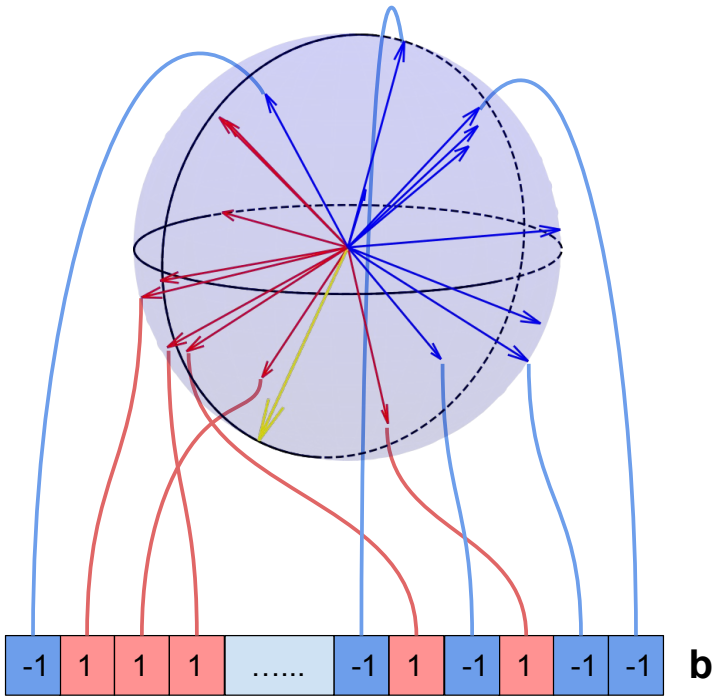
- We found lower overfitting with the binarization compared to an identical AE with similar bit-size continuous latents
- Quantization noise helps with regularisation

Latent Space Representation

- Without enforcing any prior on the latent space the learned distribution is unknown
- How to parametrize the latent distribution? GMM, KDE, autoregressive models, ...?
- Marginal Bernoulli distribution has a limit on information carried by single dimension given by its unimodal distribution with expectation $p = \mathbb{E}[b]$
- Most information is carried by higher moments
- We parametrize the latent distribution by its first and second non-central moments learned from latents encoded on the training dataset
- Our method is based on random hyperplane rounding proposed by Goemans-Williamson for the MAX-CUT [3] algorithm

Latent Space Representation

- Relax latent dimensions into unit vectors on a hypersphere
- Set angles between the vectors to be proportional to covariances of corresponding latent dimensions
- Add a boundary vector (yellow) representing the expected value of the distribution



Latent Space Parametrization

- Let us consider a matrix $\mathbf{Y} \in \{-1, 1\}^{(N \times K)}$ of K N-dimensional latents encoded on the training dataset
- Parametrize the latent space distribution by first two moments as:

$$\mathbf{M} = \begin{bmatrix} \mathbb{E}[\mathbf{Y}\mathbf{Y}^{\mathsf{T}}] & \mathbb{E}[\mathbf{Y}] \\ \mathbb{E}[\mathbf{Y}]^{T} & 1 \end{bmatrix}, \mathbf{M} \in [-1, 1]^{(N+1)\times(N+1)}$$

- Generate $N + 1$ unit length vectors on a sphere $S^{(N+1)}$ organized as rows in matrix $\mathbf{V} \in \mathbb{R}^{(N+1)\times(N+1)}, \forall i \in [1, .., N + 1], \|\mathbf{V}_i\| = 1$
- Setup angles $\alpha_{i,j}$ between pair of vectors (V_i, V_j) as:
  - $\alpha_{i,j} \to 0$ for high positive covariance
  - $\alpha_{i,j} \to \pi$ for high negative covariance
  - $\alpha_{i,j} \approx \frac{\pi}{2}$ for independent dimensions

Latent Space Parametrization

- Relate covariances in $\mathbf{M}$ to the angle $\alpha_{i,j}$ and scalar product $\langle \mathbf{V}_i, \mathbf{V}_j \rangle$

$$\frac{1}{2}(M_{i,j} + 1) = 1 - \frac{\alpha_{i,j}}{\pi} = 1 - \frac{\cos^{-1}(\langle \mathbf{V}_i, \mathbf{V}_j \rangle)}{\pi}$$

- Get $\mathbf{V}$ as a function of $\mathbf{M}$

$$H_{i,j} = \cos\left(\frac{\pi}{2}(1 - M_{i,j})\right)$$

where $\mathbf{H}$ is a Gram matrix $H_{i,j} = \langle \mathbf{V}_i, \mathbf{V}_j \rangle$

$$\mathbf{H} = \mathbf{V}\mathbf{V}^T \quad s.t. \quad \mathbf{H} \succcurlyeq 0,$$

where $\mathbf{V}$ is a row-normal lower triangular matrix after Cholesky decomposition with rows being the desired unit vectors on $S^{(N+1)}$.

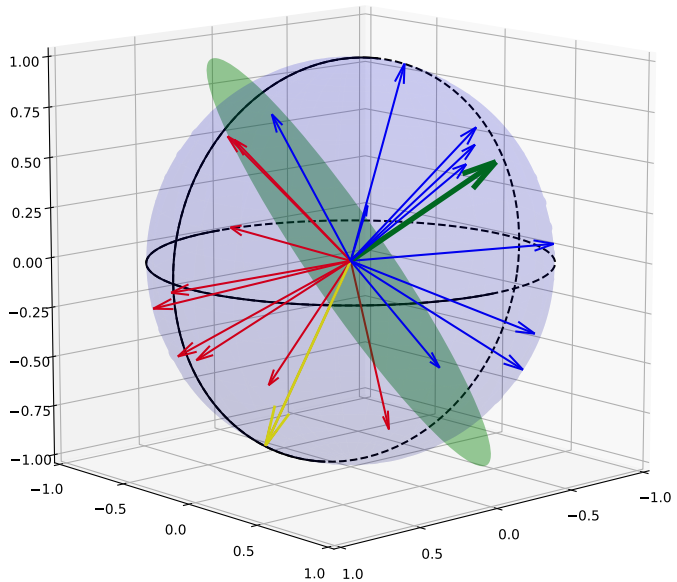
Sampling Correlated Multivariate Bernoulli Latents

- Generate random hyperplane through the center of $S^{(N+1)}$ (green)

$$\mathbf{r} \sim \mathcal{N}_{(N+1)}(0, \mathbf{I}_{(N+1)})$$

- Set positive states (red) to dimensions represented by vectors in hemisphere shared by the boundary vector $\mathbf{V}_{N+1}$ (yellow) and negative to the rest

$$b_i = \begin{cases} 1, \text{ if } f_b(\langle \mathbf{V}_i, \mathbf{r} \rangle) = f_b(\langle \mathbf{V}_{(N+1)}, \mathbf{r} \rangle) \\ -1, \text{ otherwise} \end{cases} \quad , \forall i \in [1, .., N]$$

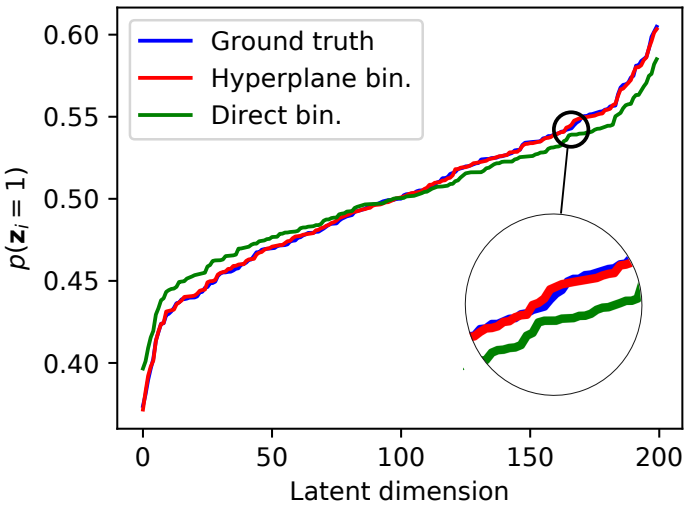


Sampling Correlated Multivariate Bernoulli Latents

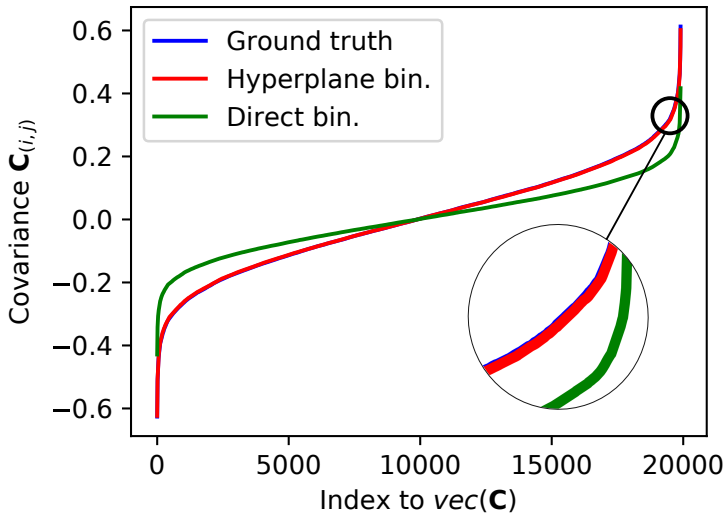
Why not sample from multivariate normal distributions with rounding?

$$\Sigma = \mathbb{E}[\mathbf{Y}\mathbf{Y}^{\mathsf{T}}] - \mathbb{E}[\mathbf{Y}]\,\mathbb{E}[\mathbf{Y}]^{T}, \quad \mathbf{z} \sim \mathcal{N}_N(0, \mathbf{I}_N)$$
$$\mathbf{b} = f_b(\mathbf{L}\mathbf{z} + \mathbb{E}[\mathbf{Y}]), \quad \mathbf{b} \in \{-1, 1\}^N,$$

where $\Sigma = \mathbf{L}\mathbf{L}^{T}$ is a lower triangular Cholesky decomposition.



(a) Sorted marginal probabilities

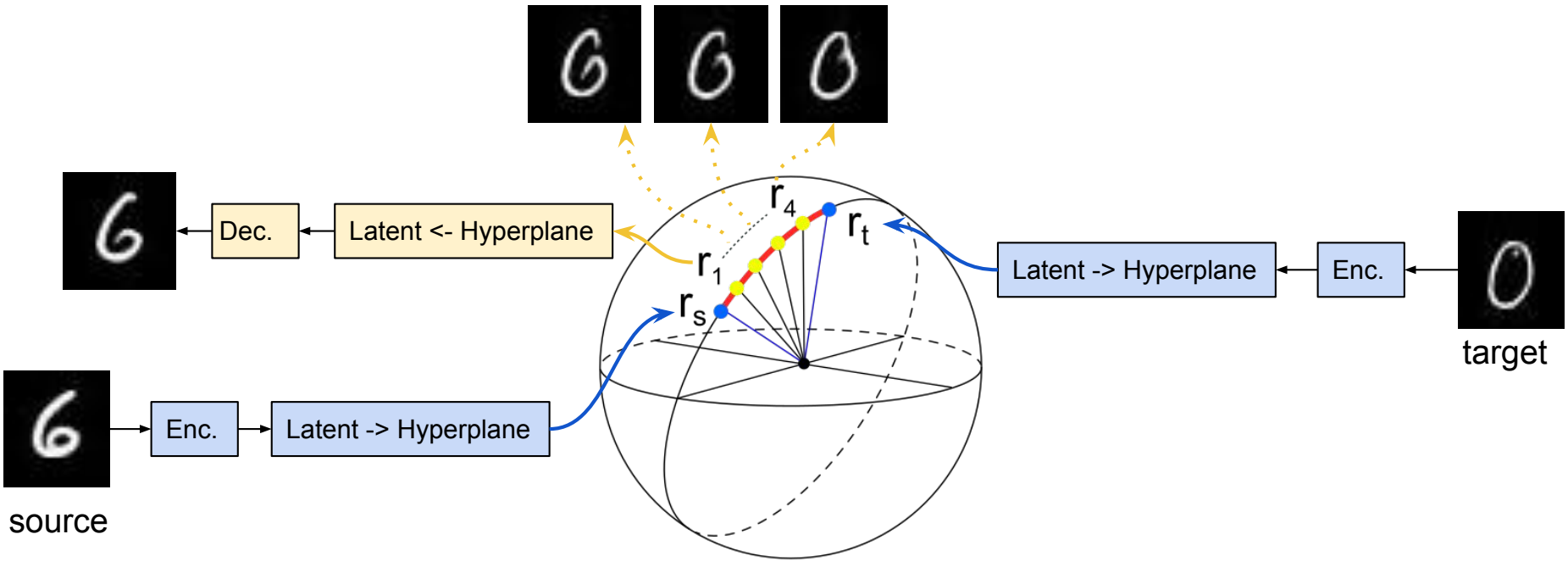


(b) Vectorized, sorted covariances.

Ground truth (GT) vs LBAE sampling vs normal dist. sampling. GT and LBAE sampling appear identical. Note that GT (blue) is mostly hidden behind the red.

Interpolation in Bernoulli Latent Space

- Encode source and target images to latents **s** and **t**
- For each find a hyperplane $\mathbf{r}_s$ and $\mathbf{r}_t$ that generates original latents
- Get T equally spaced vectors $\mathbf{r}_i, i \in [1, ..., T]$ between $\mathbf{r}_s$ and $\mathbf{r}_t$
- For each hyperplane with normal $\mathbf{r}_i$ generate a latent and decode it to an image

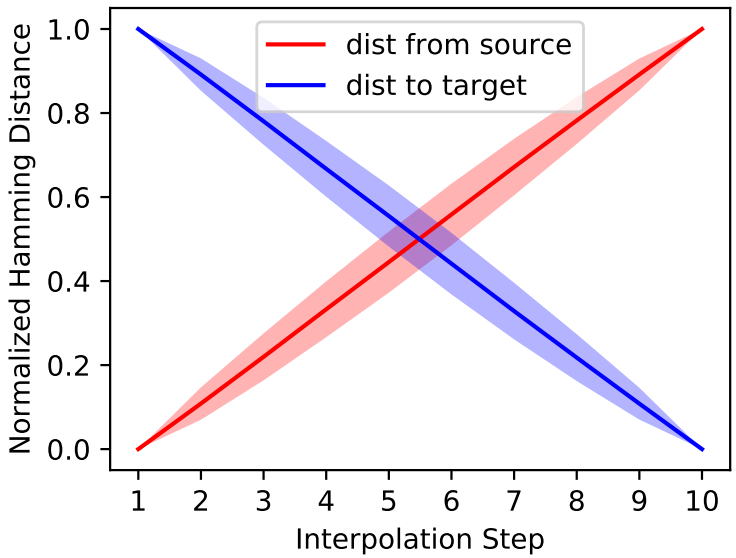


Interpolation - Latent to Hyperplane Inversion

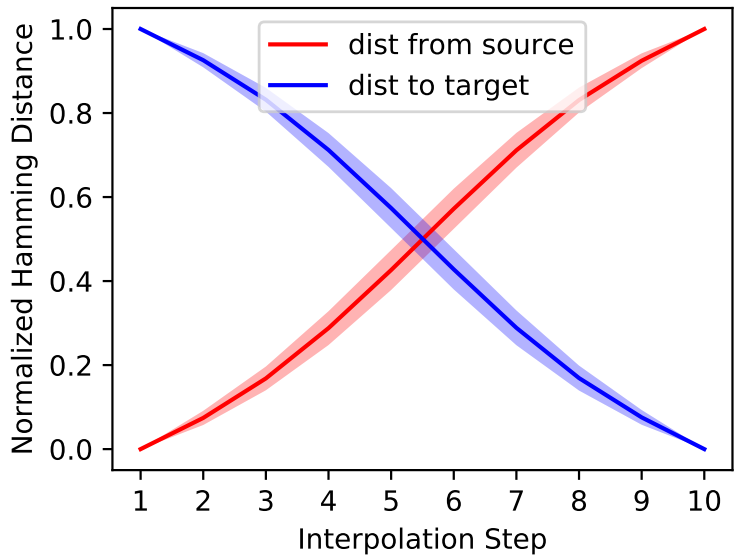
- The hyperplane position on $S^{(N+1)}$ for any given latent is not unique
- Hyperplane with a least square fit between positive and negative states is degenerated in some sense
- Interpolation between such hyperplanes produces exact copies of the source latent till the midpoint where it instantly flips to the target.
- We find the hyperplane normal for a given latent as a line through the center, closest to the centroids of its positive and negative state vectors in $\mathbf{V}$

Interpolation - Latent to Hyperplane Inversion

Hamming distance of the latents interpolated by our method changes almost linearly between the source and target.



(a) MNIST



(b) CIFAR10 (CelebA is similar)

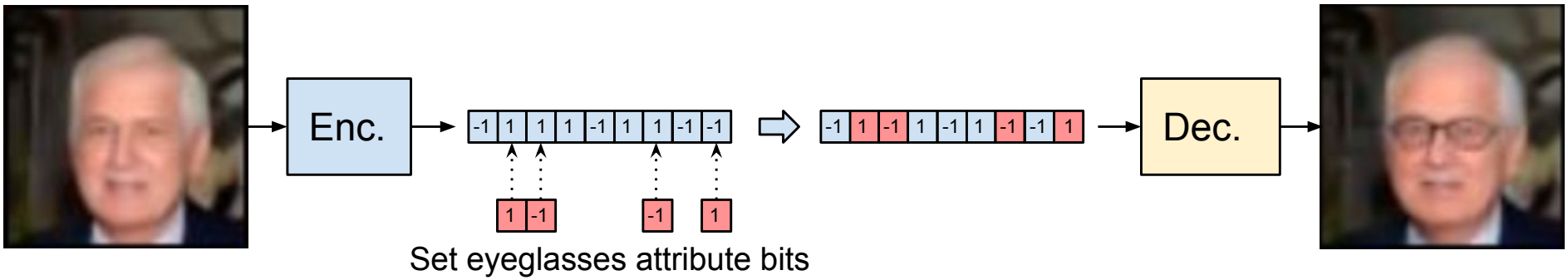
μ and σ of Hamming distances between interpolated latent at step k and source and target latents over 1k interpolations. Distances are normalized by the source-target distance.

Changing Attributes

- A simple method, no need to train the LBAE in a conditional setting
- Collect K latents $\mathbf{Y}^a \in \{-1, 1\}^{(N \times K)}$ with the attribute a
- Get $\mathbf{p} = \mathbb{E}[\mathbf{Y}^a], \mathbf{p} \in \mathbb{R}^N$
- To change the attribute a in an image represented by latent $\mathbf{b}$ set its bits b_i as such:

$$b_i = \begin{cases} 1, & \text{if} \quad p_i > D \\ -1, & \text{if} \quad p_i < -D \\ b_i, & \text{otherwise.} \end{cases}$$

- Threshold D determines how many bits will be modified
- Experimentally we found that $D = 0.1$ provides satisfactory results and set this value for all our experiments.



Quantitative Results

- Evaluated by FID[4], KID[5] and Precision/Recall[6] metrics with reference implementations[1][2][3]
- To compute FID and KID we use 10k reference and evaluation images

FID scores (lower is better)

	MNIST			CIFAR-10			CelebA		
	Reco.	Gen.	Int.	Reco.	Gen.	Int.	Reco.	Gen.	Int.
VAE [7]	18.26	19.21	18.21	57.94	106.37	88.62	39.12	48.12	44.49
WAE-MMD [7]	10.03	20.42	14.34	35.97	117.44	76.89	34.81	53.67	40.93
RAE-L2 [7]	10.53	22.22	14.54	32.24	80.8	62.54	43.52	51.13	45.98
VPGA [8]		11.67			**51.51**			**24.73**	
LBAE	**8.11**	**11.36**	**9.8**	**19.37**	53.55	**34.41**	**7.71**	34.95	**14.87**

Note that VPGA on CelebA almost entirely crop out the background, including parts of faces, which simplifies the underlying statistic.

[1] `https://github.com/bioinf-jku/TTUR`

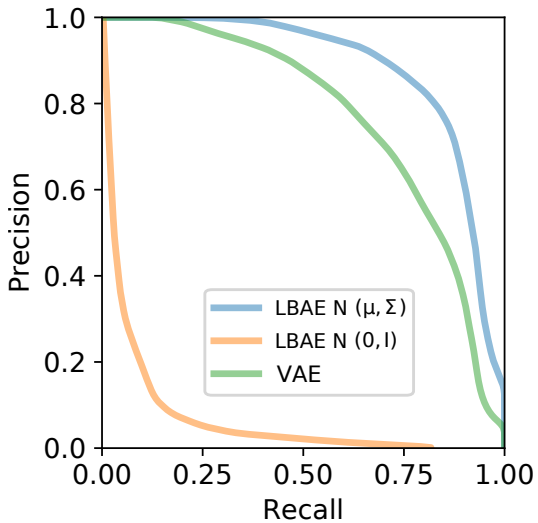
[2] `https://github.com/mbinkowski/MMD-GAN`

[3] `https://github.com/msmsajjadi/precision-recall-distributions`

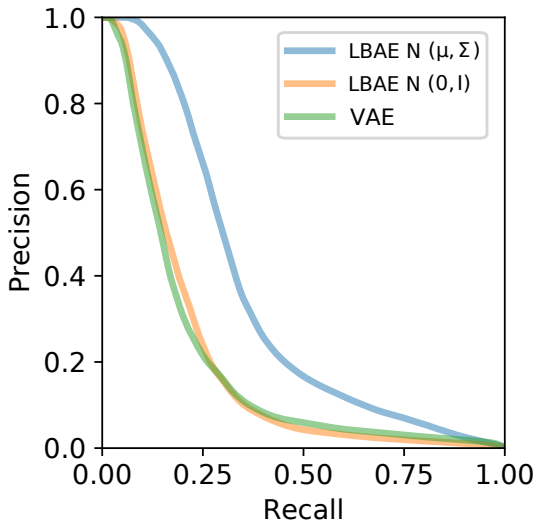
Quantitative Results

Precision/Recall (higher is better)

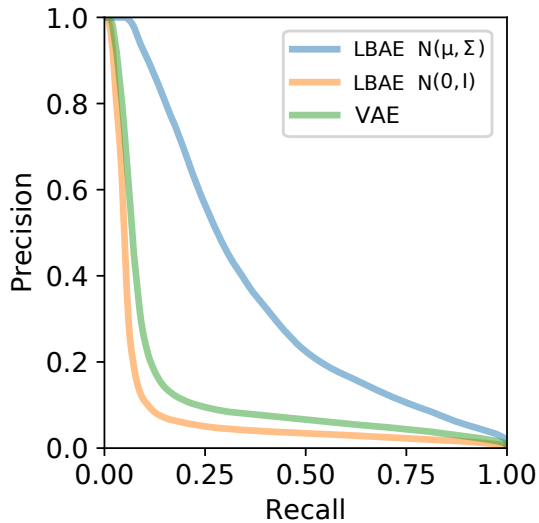
	MNIST	CIFAR-10	CelebA
VAE [7]	**0.96** / 0.92	0.25 / 0.55	0.54 / 0.66
WAE-MMD [7]	0.93 / 0.88	0.38 / 0.68	0.59 / 0.68
RAE-L2 [7]	0.92 / 0.87	0.41 / 0.77	0.36 / 0.64
LBAE	0.92 / **0.97**	**0.66 / 0.87**	**0.73 / 0.82**



(a) MNIST



(b) CIFAR-10

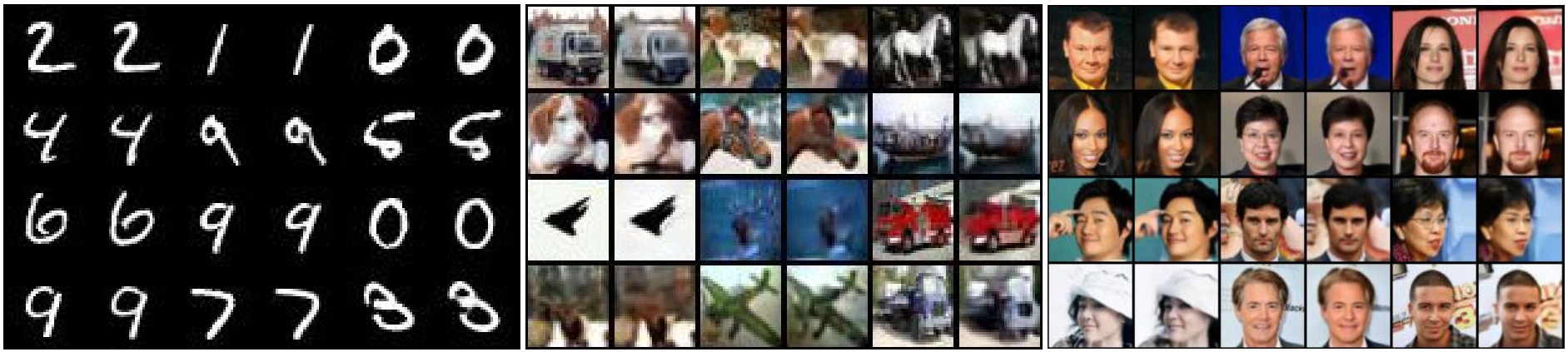


(c) CelebA

High precision and recall of LBAE signifies that the generated images represent the entire distribution and that their quality is close to the reference distribution.

Reconstruction & Random Samples

Reconstruction on test datasets



Random Samples

Interpolation

Interpolation on test datasets

Attributes Modification

Interpolation between CelebA test images (left) and the same images (right) with modified attributes (test dataset)

(a) Setting *eyeglasses* attribute.

(b) Setting *goatee* attribute.

More results in the supplemental material.

Conclusions

- We show that a simple deterministic, discrete latent autoencoder, trained with the straight-through estimator performs on a par with the current state of the art methods on common benchmarks CelebA, CIFAR-10 and MNIST
- We propose a closed form method for sampling from the Bernoulli latent space and a method for interpolation and attribute modification in this space
- Out method produces sharper images compared to VAE
- Does not suffer from mode collapse
- To our knowledge it is the first successful method that directly learns binary representations of images and allows for smooth interpolation in the discrete latent space

Thank You!

Contact: J.Fajtl@kingston.ac.uk

Paper & code: https://github.com/ok1zjf/lbae/

References I

[1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," *Proceedings of the National Academy of Sciences*, vol. 79, pp. 2554–2558, apr 1982.

[2] J. Hawkins and S. Ahmad, "Why neurons have thousands of synapses, a theory of sequence memory in neocortex," *Frontiers in Neural Circuits*, vol. 10, p. 23, 2016.

[3] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.

[4] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are gans created equal? a large-scale study," in *Advances in Neural Information Processing Systems*, pp. 700–709, 2018.

References II

[5] M. Bikowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," in *International Conference on Learning Representations*, 2018.

[6] M. S. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, "Assessing generative models via precision and recall," in *Advances in Neural Information Processing Systems*, pp. 5228–5237, 2018.

[7] P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. Black, and B. Scholkopf, "From variational to deterministic autoencoders," in *International Conference on Learning Representations*, 2020.

[8] Z. Zhang, R. Zhang, Z. Li, Y. Bengio, and L. Paull, "Perceptual generative autoencoders," in *International Conference on Learning Representations, Workshop DeepGenStruct*, 2019.