Eccentric regularization: minimizing hyperspherical energy without explicit projection

Xuefeng Li and Alan Blair
School of Computer Science and Engineering
University of New South Wales, Sydney, 2052, Australia
{xuefeng.li1,a.blair}@unsw.edu.au

***Abstract*—Several regularization methods have recently been introduced which force the latent activations of an autoencoder or deep neural network to conform to either a Gaussian or hyperspherical distribution, or to minimize the implicit rank of the distribution in latent space. In the present work, we introduce a simple and novel regularizing loss function which simulates a pairwise repulsive force between items and an attractive force of each item toward the origin. We show that minimizing this loss function in isolation achieves a hyperspherical distribution, and demonstrate its effectiveness as a regularizer for an image autoencoder. Moreover, a reduction in the regularization parameter leads to a modest increase in the eccentricity of the distribution in latent space. This enhances image generation, and allows the eigenvectors of the covariance matrix to be extracted as deep principal components, which can be used for data analysis, image generation, visualization and downstream classification.**

I. Introduction

In recent years a number of regularization methods have been introduced which force the latent activations of an autoencoder or deep neural network to conform to either a hyperspherical or Gaussian distribution, in order to encourage diversity in the latent vectors, or to minimize the implicit rank of the distribution in latent space.

Variational Autoencoders (VAE) [1] and related variational methods such as β-VAE [2] force the latent distribution to match a known prior distribution by minimizing the Kullback-Leibler divergence. Normally, a standard Gaussian distribution is used as the prior, but alternatives such as the hyperspherical distribution have also been investigated in the literature due to certain advantages [3]. More recently, deterministic alternatives have been proposed such as Wasserstein Autoencoder (WAE) [4], VQ-VAE [5] and RAE [6].

Several existing methods encourage diversity by maximizing pairwise dissimilarity between items, drawing inspiration in part from a 1904 paper by J.J. Thomson [7] in which various classical models are proposed for maintaining the electrons of an atom in an appropriate formation around the nucleus. Hyperspherical Energy Minimization [8] has been used to regularize the hidden unit activations in deep networks, with a Thomson-like loss function which projects each vector onto a hypersphere and simulates a repulsive force pushing these projected items away from each other and spreading them evenly around the sphere. Other methods use an implicit projection by optimizing for the cosine similarity between vectors [9]–[11].

Recently, new compressive techniques have been developed with the aim of minimizing the intrinsic dimension of the latent space. This can be achieved by introducing projection mappings [12] or additional linear layers between the encoder and decoder [13], or by other methods such as optimizing for a variational lower bound on the mutual information between datapoints [14].

The word *eccentricity* (sometimes also called "unevenness" [15]) can be used as a general term for the extent to which the spectrum of the covariance matrix in latent space differs from that of a hyperspherical or standard normal distribution. Methods such as VAE aim to give equal importance to all latent features, while compressive techniques like IRMAE [13] explicitly encourage some features to become dominant while others diminish.

In the present work, we introduce a simple and novel regularizing loss function and show that it is minimized when the latent vectors conform to a hyperspherical distribution. We demonstrate the effectiveness of this loss function as a regularizer for an image autoencoder. We find that, for our proposed method as well as for the Wasserstein Autoencoder (WAE) [4], improved Fréchet Inception scores can be achieved when the scaling parameter is reduced, allowing the eccentricity of the distribution in latent space to moderately increase. This increased eccentricity additionally enables the eigenvectors of the covariance matrix to be extracted as deep principal components, which can be used for data analysis, image generation, visualization and downstream classification.

II. Eccentric Loss Function

We consider a family of loss functions $l_{\mu,M}$ on a set of n items $\mathbf{z}_i \in \mathbb{R}^d$ of the form:

$$l_{\mu,M}(\{\mathbf{z}_i\}) = \frac{1}{n(n-1)} \sum_{\mathbf{z}_i \neq \mathbf{z}_j} K_{\mu,M}(\mathbf{z}_i, \mathbf{z}_j)$$

where

$$K_{\mu,M}(\mathbf{z}_i, \mathbf{z}_j) = \big(\frac{||\mathbf{z}_i||^2 + ||\mathbf{z}_j||^2}{2}\big) - \mu M \log\big(1 + \frac{||\mathbf{z}_i - \mathbf{z}_j||^2}{M}\big).$$

Note that the gradient of $K_{\mu,M}$ effectively simulates a repulsive force between all pairs of items $\mathbf{z}_i$, $\mathbf{z}_j$, equal to

$2\mu(\mathbf{z}_i - \mathbf{z}_j)/\big(1 + ||\mathbf{z}_i - \mathbf{z}_j||^2/M\big)$, and an attractive force $(-\mathbf{z}_i)$ of each item $\mathbf{z}_i$ toward the origin.

Equation (1) ostensibly has two free parameters μ and M, but we intend to set M as a function of μ and the dimension d in such a way that the loss function is minimized on a hyperspherical distribution of radius approximately $\sqrt{d}$, which is a close approximation (in the sense of Wasserstein distance) to the Standard Normal distribution. The relationship between μ, M and the radius ρ of the stationary spherical distribution is determined by this theorem:

Theorem 1: Let S_ρ denote the uniform distribution on a hypersphere of radius ρ in dimension $d \geq 2$, and let $\Gamma()$ denote the Gamma Function. Then S_ρ is a stationary point for $l_{\mu,M}$, provided the following expression is equal to $\frac{1}{\mu}$:

$$\frac{M}{\rho^2\sqrt{\pi}} \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d-1}{2})} \int_{u=1}^{1+\frac{4\rho^2}{M}} \frac{(\frac{M}{2\rho^2}(u-1))^{\frac{d-1}{2}}(2 - \frac{M}{2\rho^2}(u-1))^{\frac{d-3}{2}}}{u} du.$$

Proof: (see Appendix)

Our aim is to choose M in such a way that the radius ρ of the stable spherical distribution for $l_{\mu,M}$ is very close to $\sqrt{d}$. Henceforth we will use l_μ to denote $l_{\mu,M}$ with M given by:

$$M = 2d(1 + \frac{1}{2\mu(d-1)})/(2\mu - 1).$$

(The motivation for this choice is explained in the Appendix).

In order to test the accuracy of this approximation, we set M as above and compute ρ using numerical integration and bisection, for each dimension d and for all values of μ between 1 and $2d + 1$ (in increments of 0.01). Figure 1 shows the percentage difference between ρ and $\sqrt{d}$, maximized over μ, for d between 2 and 300. We see that ρ is within 0.1% of $\sqrt{d}$ for $d \geq 12$, within 0.01% for $d \geq 38$, and within 0.001% for $d \geq 117$.

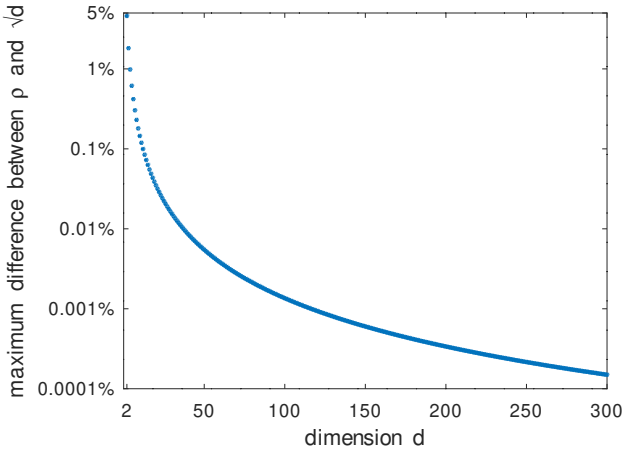


Fig. 1. Maximum percentage difference between ρ and $\sqrt{d}$.

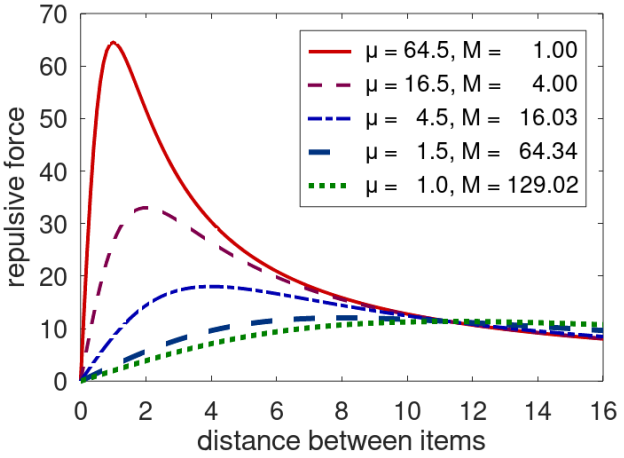


Fig. 2. Repulsive force between pairs of items, based on their distance, for dimension $d = 64$.

Figure 2 shows the magnitude of the repulsive force between any two items based on the distance between them, for various values of μ (and corresponding M). Note that the repulsive force increases until a distance of $\sqrt{M}$ and decreases thereafter.

III. Regularization and Tolerance to Eccentricity

We are primarily interested in the eccentric loss function for the purpose of regularization, as part of an overall loss function of the form

$$\text{loss} = \text{loss}_{\text{other}} + \lambda l_\mu(\{\mathbf{z}_i\}).$$

The scaling factor λ and the parameter μ regulate both the local flexibility of the distribution and its tolerance to eccentricity.

Figure 2 shows the magnitude of the repulsive force between any two items based on the distance between them. Note that the repulsive force increases until a distance of $\sqrt{M}$ and decreases thereafter. At one extreme, where $\mu = d + \frac{1}{2}$ and $M \simeq 1$, the repulsive force is strongest for nearby items, thus forcing them to spread out evenly on both a local and global scale. At the other extreme, where $\mu = 1$ and $M \simeq 2d + 1$, each item is influenced most strongly by items that are far away, thus forcing the items to spread out on a global scale but allowing some flexibility on a local scale.

A. Example: MNIST in Two Dimensions

Figure 3 illustrates the behavior of this regularization in the case where $\text{loss}_{\text{other}}$ is the L_2 distance between the original and reconstructed images for an autoencoder trained on the MNIST dataset, with $d = 2$ (details are given in the next section). When λ is large (top row) the distribution conforms closely to a sphere of radius $\sqrt{d}$. As λ decreases (middle row) the distribution deviates from the sphere but starts to roughly approximate a standard normal distribution. When λ decreases further (lowest row) the eccentricity also increases, allowing greater variance in some directions than others.

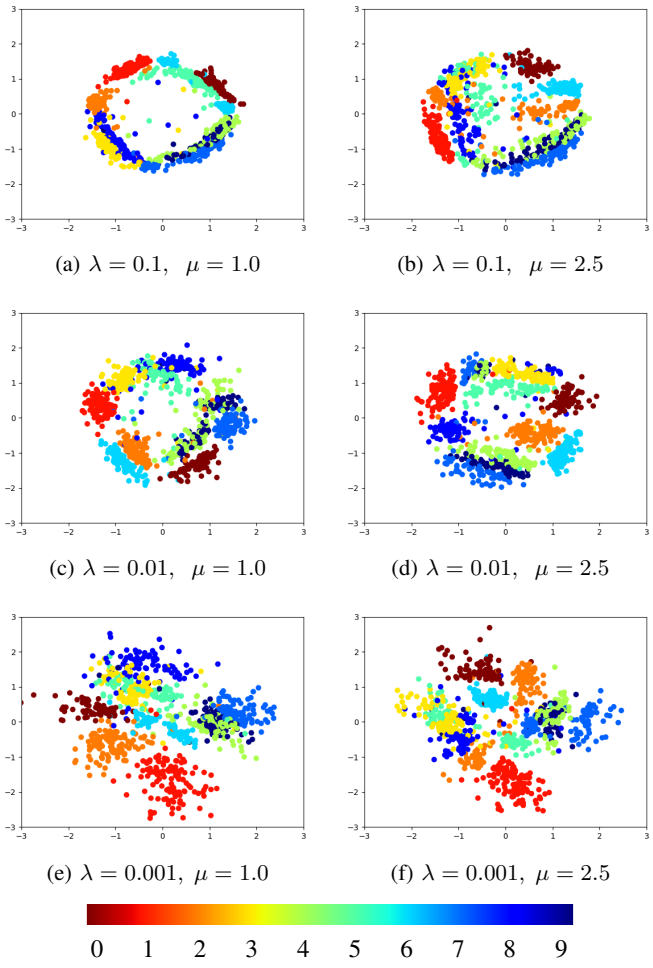


Fig. 3. Distribution in latent space of a random selection of test items, for Eccentric Autoencoder trained on the MNIST dataset using two latent dimensions, with $\lambda = 0.1$, 0.01 or 0.001 and $\mu = 1.0$ or 2.5. The horizontal and vertical axes are the principal components corresponding to the larger and smaller eigenvalue of the covariance matrix.

B. Intuitive Description

We find experimentally that this tolerance of our loss function to increased eccentricity occurs for any dimension d, provided the trace of the covariance matrix is approximately equal to d. This can be explained intuitively as follows: Consider a point $\mathbf{z} = (\sqrt{d}, 0, \ldots, 0)$ on the sphere $\mathcal{S}_{\sqrt{d}}$ of radius $\sqrt{d}$ (see Figure 4). Since the density of $\mathcal{S}_{\sqrt{d}}$ is heavily concentrated in vectors nearly perpendicular to $\mathbf{z}$, we can think of a "typical" point $\mathbf{z}_\perp \in \mathcal{S}_{\sqrt{d}}$ as being nearly perpendicular to $\mathbf{z}$ and exerting a repulsive force on $\mathbf{z}$ whose component in the direction of $\mathbf{z}$ is approximately

$$2\mu\,\mathbf{z}/\big(1 + ||\,\mathbf{z} - \mathbf{z}_\perp||^2/M\big) \simeq \mathbf{z}\big(2\mu/(1 + 2d/M)\big) \simeq \mathbf{z}\,.$$

When these forces are aggregated, the transverse components cancel each other out while the radial components combine to exactly balance the attractive force of $(-\mathbf{z})$. The influence of these "typical" points is particularly dominant when $\mu = 1$, because in this case the pairwise repulsive force also reaches its maximum at a distance of approximately $\sqrt{2d}$ (see Figure 2).

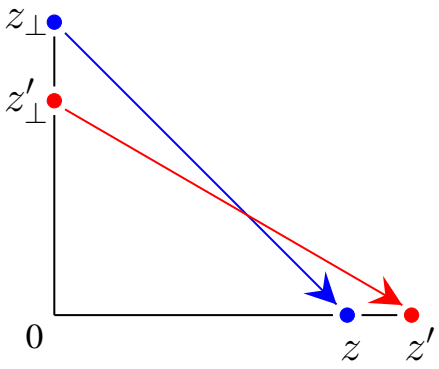


Fig. 4. The point $\mathbf{z} = (\sqrt{d}, 0, \ldots, 0)$ is repelled by a "typical" point $\mathbf{z}_\perp$ on the sphere $\mathcal{S}_{\sqrt{d}}$ with a force whose radial component approximately balances the attractive force $(-\mathbf{z})$; when the sphere $\mathcal{S}_{\sqrt{d}}$ is transformed to an ellipse with covariance $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$ with $\mathrm{Trace}(\Sigma) = \sum_i \sigma_i^2 = d$, the transformed points $\mathbf{z}'$, $\mathbf{z}'_\perp$ satisfy $||\,\mathbf{z}' - \mathbf{z}'_\perp|| \simeq ||\,\mathbf{z} - \mathbf{z}_\perp||$ and the radial component continues to approximate the (new) attractive force $(-\mathbf{z}')$.

Now suppose that the sphere $\mathcal{S}_{\sqrt{d}}$ is transformed into an ellipse with covariance Σ, where $\mathrm{Trace}(\Sigma) = d$, and assume for simplicity that $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$ is diagonal. The transformation $\Sigma^{\frac{1}{2}}$ maps $\mathbf{z}$ to $\mathbf{z}' = (\sigma_1\sqrt{d}, 0, \ldots, 0)$, and the condition $\sum_i \sigma_i^2 = \mathrm{Trace}(\Sigma) = d$ ensures that the "typical" point $\mathbf{z}_\perp$ will be mapped to a point $\mathbf{z}'_\perp$ for which $||\,\mathbf{z}' - \mathbf{z}'_\perp|| \simeq ||\,\mathbf{z} - \mathbf{z}_\perp||$. But, due to the change in angle, the radial component of the repulsive force will change from $(-\mathbf{z})$ to $(-\mathbf{z}')$, keeping it in balance with the attractive force.

C. Minimum Loss Value

When $\mu = 1$, we find experimentally that the minimum value of the loss function l_μ^d obtainable through gradient descent can be approximated by

$$l_\mu^d \simeq d(1 - 2\log 2) \simeq -0.386\,d\,.$$

This can be understood using the same intuition as above, namely that two random points $\mathbf{z}_i, \mathbf{z}_j$ on $\mathcal{S}_{\sqrt{d}}$ are likely to be nearly perpendicular, so an approximate value for l_μ on $\mathcal{S}_{\sqrt{d}}$ is given by:

$$\begin{aligned} K_{\mu,M}(\mathbf{z}_i, \mathbf{z}_j) &= \big(\frac{||\mathbf{z}_i||^2 + ||\mathbf{z}_j||^2}{2}\big) - \mu M \log\big(1 + \frac{||\mathbf{z}_i - \mathbf{z}_j||^2}{M}\big) \\ &\simeq \frac{d+d}{2} - (1)(2d{+}1)\log(1 + \frac{2d}{2d+1}) \\ &\simeq d(1 - 2\log 2). \end{aligned}$$

IV. Autoencoder Experiments

In this section, we explore the eccentric loss function as a method for regularizing the latent variables of an autoencoder. The loss function in this case is:

$$\mathrm{loss} = \mathrm{loss}_{\mathrm{recon}} + \lambda\, l_\mu(\{\mathbf{z}_i\}),$$

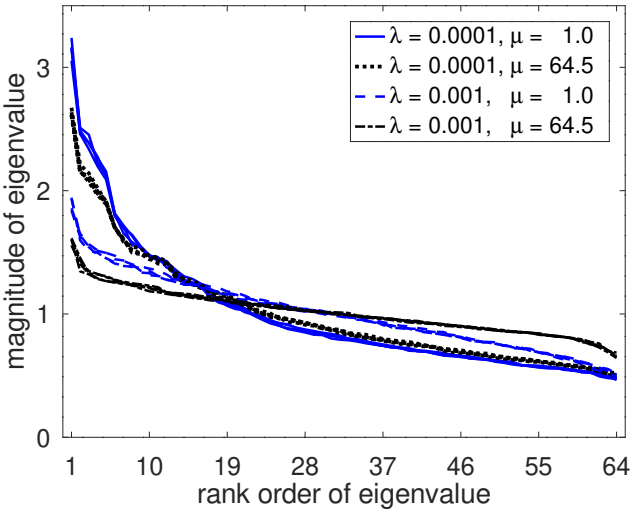


Fig. 5. Spectrum of latent variable covariance matrix for Eccentric Autoencoders trained on CelebA; four indepedent runs are superimposed for each combination of λ and μ.

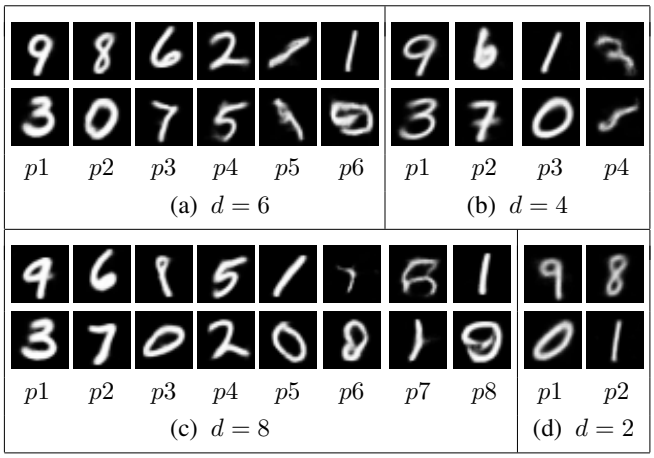


Fig. 6. Paired eigen-digits for deep principal components of Eccentric Autoencoders trained on MNIST with latent dimension 6, 4, 8 and 2.

where $\text{loss}_{\text{recon}}$ is the L_2 distance between the original and reconstructed image, for a training batch $\{\mathbf{z}_i\}$.

Specifically, we train Eccentric Autoencoders on:

(a) MNIST dataset [16] for 1000 epochs using dimension $d = 2, 4, 6, 8$ with $\lambda = 10^{-3}, \mu = 1$;

(b) CelebA [17] for 100 epochs using $\lambda = 10^{-4}, 10^{-3}$, $\mu = 1, 1.5, 4.5, 16.5, 64.5$, with dimension $d = 64$.

As far as possible, we try to use the same network structure and hyperparameters as [4], with the Adam optimizer [18], batch size 100, learning rate of 10^{-4} and weight decay of 10^{-6} (see Appendix for details). Each MNIST run takes approximately 3 hours on a GeForce GTX 1080 Ti; each CelebA run takes approximately 6 hours on one node of a V100 GPU.

Figure 5 shows the eigenvalues for the covariance matrix of the CelebA images in latent space, for different values of λ and μ. We see that the eigenvalues become more uniform (closer to 1.0) as λ increases and, to a lesser extent, as μ increases. When λ and μ decrease, the distribution becomes more eccentric, with eigenvalues ranging as high as 3.24 and as low as 0.47. As discussed in the previous section, the sum of the eigenvalues (indicated by the area under the curve) is approximately equal to the dimension d. Note that in each case the curves from four independent runs match each other almost exactly, indicating that the spectrum is largely invariant from one run to another.

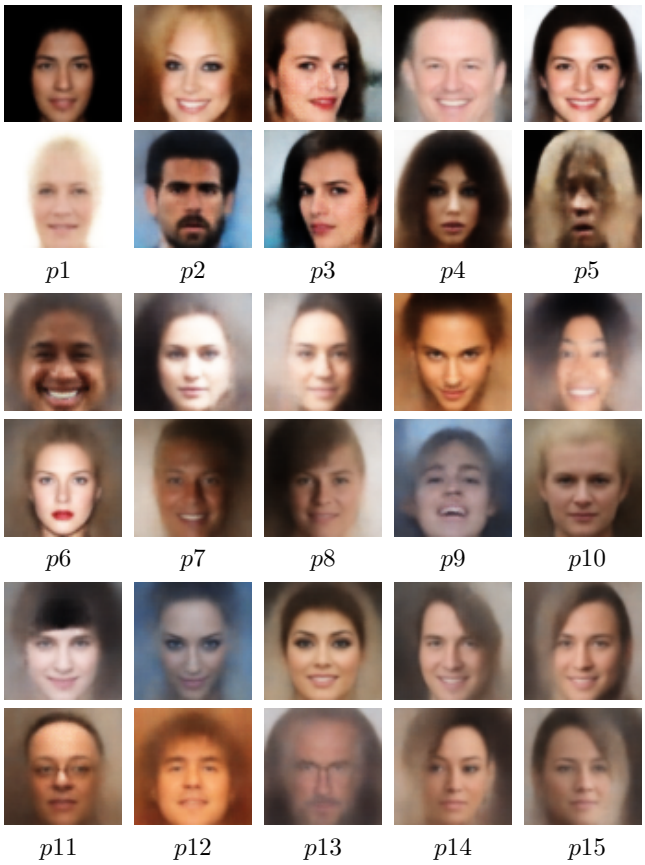


Fig. 7. Paired eigen-faces corresponding to first fifteen deep principal components for Eccentric Autoencoder trained on CelebA dataset.

A. Principal Components and Visualization

If we shift and rotate the latent vectors of the training items so that their mean becomes zero and their coordinate axes lie along the eigenvectors of the covariance matrix in decreasing order, we can extract pairs of deep eigen-digits or eigen-faces by choosing appropriately scaled positive and negative basis vectors along each of these principal axes. Eigen-digits for MNIST with different latent dimension are shown in Figure 6, and eigen-faces for CelebA are shown in Figure 7.

This decomposition could potentially serve as the basis for a visualization tool or an interactive interface for generating desired images. Suppose, for example, that we wish to further explore the effect of components $p2, p3, p6, p9, p10$ and $p11$

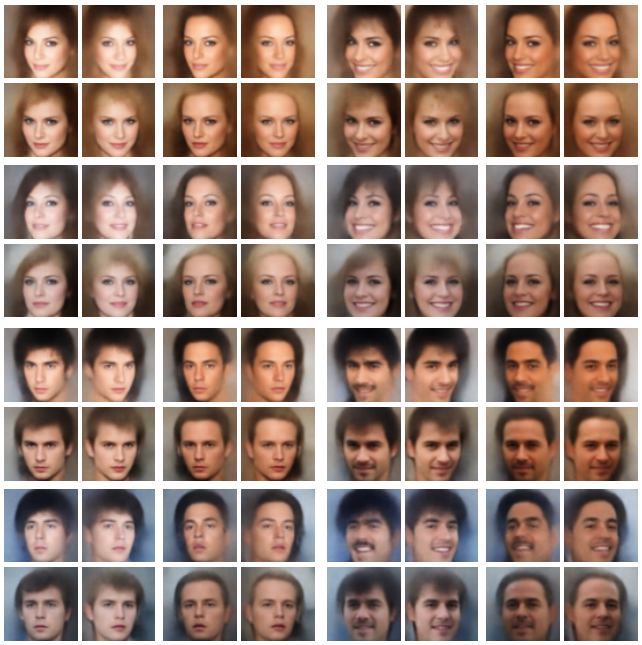
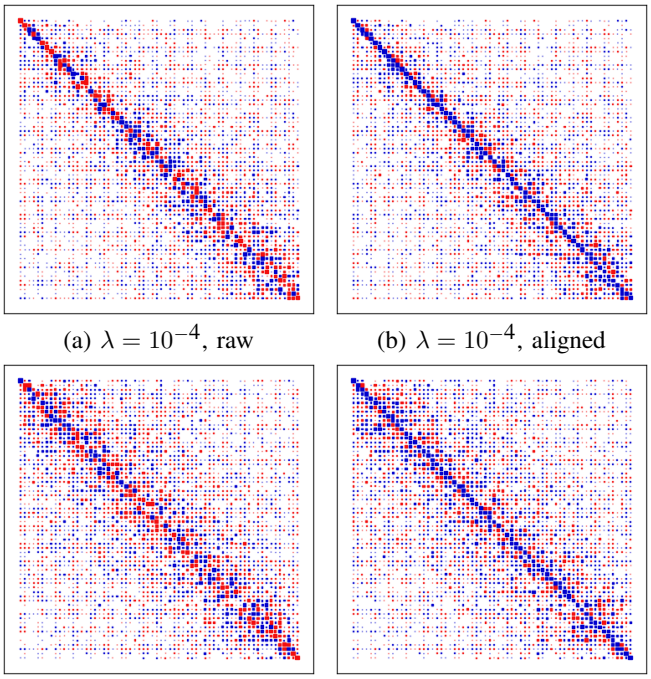


Fig. 8. Images generated from principal components $p6$, $p11$, $p3$ (horizontally) and $p2$, $p10$, $p9$ (vertically).



(a) $\lambda = 10^{-4}$, raw (b) $\lambda = 10^{-4}$, aligned

(c) $\lambda = 10^{-3}$, raw (d) $\lambda = 10^{-3}$, aligned

Fig. 9. Cross correlation between deep principal components for Eccentric Autoencoders with $\mu = 1$, trained independently on the CelebA dataset from different random initial weights, before the alignment procedure (left) and afterwards (right).

in Figure 7. We could generate the plot shown in Figure 8 by making the selected coordinates positive and negative in all possible combinations.

When contemplating such a visualization tool, we believe there are certain advantages to be gained from allowing unevenness or eccentricity in the eigenvalues of the covariance matrix. For example, it may concentrate the intrinsic variation more heavily into the first few principal components, thus reducing the cognitive load for the human user and alerting them to the relative prominence of the various latent factors. Secondly, it may provide a greater degree of consistency from one run to another, allowing a relatively stable set of principal components to be extracted when the algorithm is re-run with different initial weights.

B. Representation Alignment and Visualization

The full mapping to the rotated latent space can be considered as an encoding which maps each input image to a vector $(p_1, \ldots, p_d)$ where p_k is the value of the coordinate in the direction of the kth (deep) principal component. We are interested in the question: How canonical is this encoding? In other words, if we train two autoencoders independently from different random weights, producing two encodings $E_1 : \text{Img} \mapsto (p_1, \ldots, p_d)$ and $E_2 : \text{Img} \mapsto (q_1, \ldots, q_d)$, how similar are these two mappings?

The images on the left side of Figure 9 illustrate the correlation between the principal components of two such mappings. We note that it is possible for principal components corresponding to close eigenvalues to be permuted, for example $p2$ and $p3$ in Figure 6 (a) and (c), which would appear as slightly off-diagonal terms in Figure 9. It is also possible for the positive and negative eigenvectors to be inverted, for example $p4$ in Figure 6 (a) and (c), which would show up in Figure 9 as diagonal terms which are negative (red) rather than positive (blue).

In order to resolve these ambiguities and try to bring the two embeddings into approximate alignment, we use the Hungarian Algorithm [19] to find a permutation which maximizes the trace of the absolute (training set) cross-covariance between the two mappings, and then flip the signs of individual components to make the cross-covariance matrix positive along the diagonal. This alignment procedure can be visualized by comparing the (test set) correlations on the left of Figure 9 with those on the right. It has the effect of bringing the correlation closer to the diagonal, and turning it from negative (red) to positive (blue). The correlation appears to be more heavily concentrated along the diagonal for $\lambda = 10^{-4}$ than for $\lambda = 10^{-3}$. The appearance of a 2-by-2 block along the diagonal with three blue squares and one red square, such as in the top left corner of Figure 9(d), is indicative of a rotation in the 2-dimensional subspace formed by these two components. The alignment between representations from different runs can be quantified by measuring the average angle between corresponding vectors in the two representations (shown in Table 1, and discussed in the next subsection).

Algorithm	λ	ε	angle(raw)	angle(align)	FI score
EAE	10^{-2}	0.12	81°± 1°	69°± 1°	53.7 ± 1.2
EAE	10^{-3}	0.29	69°± 2°	54°± 1°	50.2 ± 0.5
EAE	3×10^{-4}	0.49	59°± 1°	47°± 1°	48.7 ± 1.0
EAE	10^{-4}	0.54	53°± 2°	43°± 1°	49.5 ± 0.9
EAE	10^{-5}	0.71	55°± 5°	39°± 1°	50.3 ± 0.8
EAE	10^{-6}	0.75	50°± 2°	40°± 1°	55.0 ± 1.7
VAE	10^{-3}	0.25	76°± 2°	63°± 1°	50.0 ± 0.7
VAE	3×10^{-4}	0.41	71°± 2°	58°± 1°	53.7 ± 1.9
VAE	10^{-4}	0.56	59°± 1°	46°± 1°	57.7 ± 1.0
VAE	10^{-5}	0.71	51°± 2°	40°± 1°	57.8 ± 0.2
WAE	10^{-1}	0.19	79°± 1°	65°± 1°	53.5 ± 0.8
WAE	10^{-2}	0.29	62°± 2°	49°± 1°	49.2 ± 1.4
WAE	3×10^{-3}	0.45	53°± 4°	40°± 1°	49.4 ± 0.6
WAE	10^{-3}	0.59	51°± 3°	40°± 2°	50.0 ± 1.0
WAE	3×10^{-4}	0.69	52°± 2°	40°± 1°	51.2 ± 0.6
WAE	10^{-4}	0.76	53°± 5°	41°± 1°	54.2 ± 0.4
AE	0	0.71	48°± 1°	38°± 1°	55.8 ± 0.1

TABLE I
ECCENTRICITY (ε) OF DISTRIBUTION IN LATENT SPACE, MEAN ANGLE (RAW AND ALIGNED) BETWEEN INDEPENDENTLY TRAINED PRINCIPAL COMPONENT ENCODINGS, AND FRÉCHET INCEPTION (FI) SCORE, FOR AUTOENCODER MODELS TRAINED ON CELEBA WITH DIFFERENT VALUES OF THE SCALING PARAMETER λ.

C. Eccentricity, Alignment and Fréchet Inception Score

Trained autoencoders can be used for image generation, with latent vectors chosen randomly from either a standard normal or multivariate Gaussian distribution. The Fréchet Inception (FI) score has become a standard tool for measuring how well the distribution of generated images conforms to that of an unseen set of test images. In the case where the latent vectors are chosen from a standard normal distribution, Tolstikhin et al. [4] report an FI score of 63 for VAE, 55 for WAE-MMD and 42 for WAE-GAN (which includes a discriminator). In this section, we use a scaling parameter to modulate the eccentricity of the distribution, and explore image generation using latent vectors chosen from a multivariate Gaussian whose mean and covariance match those of the encoded latent vectors from the training images.

Table I shows the mean angle between encodings of the same (test set) images, averaged across six pairwise combinations of four runs of our Eccentric Autoencoder (EAE), for different choices of λ, along with the Fréchet Inception score and the eccentricity, which we define to be the standard deviation of the eigenvalues of the covariance matrix divided by their mean. For comparison, we include VAE [1] and WAE-MMD [4], as well as a plain autoencoder (AE) with no regularization other than weight decay.

Each of the three regularizers shows a similar pattern. When λ is (relatively) large, the distribution in latent space becomes close to both the spherical and the standard normal distribution. As λ decreases, the distribution becomes less constrained and gradually closer to that of an unregulated autoencoder (AE), which has an eccentricity ε of 0.71, and generates images with an FI score of 55.8.

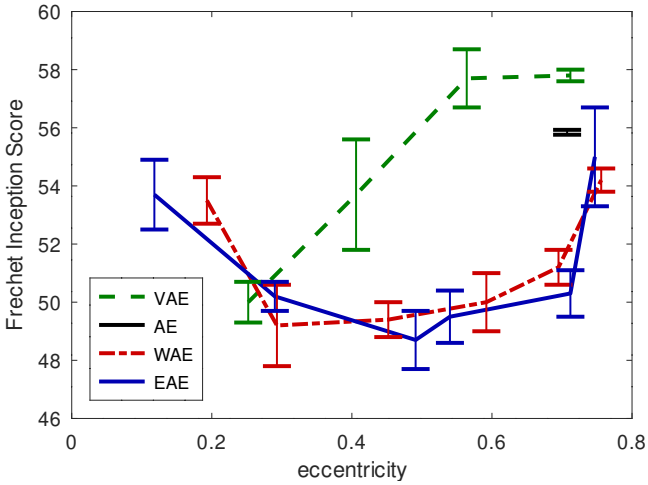


Fig. 10. Relationship between eccentricity (ε) and Fréchet Inception (FI) score, for models listed in TABLE I.

Our Eccentric Autoencoder (EAE) and the Wasserstein Autoencoder (WAE) both achieve a minimum FI score of around 49 or 50 for ε in the range of 0.3 to 0.7 (see Figure 10). The FI score increases to around 54 when ε is low (over-regulation) and when it is high (under-regulation). For the Variational Autoencoder (VAE) the FI score is around 50 for $\varepsilon \simeq 0.25$ but increases to 58 when $\varepsilon \simeq 0.56$. It should be noted that we are only comparing different methods for regularizing autoencoders, and that considerably better image generation can now be achieved by other methods such as 2-Stage VAE [20], BigGAN [21] and diffusion models [22].

As explained in the previous subsection, modulating for increased eccentricity may have the additional benefit of allowing deep principal components to be extracted in a manner that is relatively invariant from one run to another. This is quantified in the 5th column of Table 1, where the mean angle between corresponding latent vectors across different runs reaches a minimum of 39 or 40 degrees for ε greater than about 0.45 (WAE) or 0.6 (EAE).

D. Downstream Classification

Another way to test the efficacy of an autoencoder is to train a classifier on the latent variables using different sized subsets of labeled training items and measure the performance. Table II shows the result of training a KNN classifier on subsets of labeled training items of the specified size, averaged across three autoencoder runs and 20 random subsets for each run. For comparison, the results for $d = 8$ and training size ≥ 100 are very close (in fact, within the margin of error) to those reported for IRMAE in [13] although in that case an MLP with two fully connected layers of dimension 128 was used for classification rather than KNN.

	10(1)	100(1)	1000(5)	10000(10)	60000(15)
$d=2$	39.0 ± 7.2	19.1 ± 2.6	12.4 ± 1.3	10.9 ± 0.9	10.6 ± 0.8
$d=4$	36.0 ± 6.6	12.1 ± 1.9	5.4 ± 0.6	4.0 ± 0.3	3.7 ± 0.3
$d=6$	33.7 ± 7.3	10.0 ± 1.3	4.1 ± 0.3	3.0 ± 0.1	2.6 ± 0.1
$d=8$	34.0 ± 5.1	9.5 ± 1.1	3.7 ± 0.2	2.4 ± 0.1	2.0 ± 0.1

TABLE II
ERROR RATE FOR DOWNSTREAM CLASSIFICATION ON MNIST WITH DIFFERENT NUMBERS OF LABELED TRAINING ITEMS, USING A KNN CLASSIFIER (NUMBER OF NEIGHBORS SPECIFIED IN PARENTHESES).

λ	μ	DT(1)	DT(8)	KNN(20)
10^{-4}	1.0	16.68 ± 0.12	14.42 ± 0.11	13.15 ± 0.03
10^{-4}	64.5	17.65 ± 0.33	14.56 ± 0.16	13.14 ± 0.04
10^{-3}	1.0	17.61 ± 0.37	15.20 ± 0.19	13.30 ± 0.01
10^{-3}	64.5	17.90 ± 0.32	15.52 ± 0.17	13.43 ± 0.02

TABLE III
CLASSIFICATION ERROR AVERAGED OVER 40 ATTRIBUTES FOR CELEBA, USING DECISION TREE OF DEPTH 1 (STUMP), DECISION TREE WITH MAXIMUM DEPTH 8, AND KNN WITH 20 NEIGHBORS.

Attribute	Component	Information Gain
3. Attractive	$p2$	50% → 65%
19. Heavy Makeup	$p2$	60% → 76%
20. High Cheek Bones	$p2$	52% → 68%
21. Male	$p2$	61% → 80%
22. Mouth Slightly Open	$p6$	50% → 66%
32. Smiling	$p6$	50% → 67%
37. Wearing Lipstick	$p2$	52% → 79%

TABLE IV
ATTRIBUTES FOR WHICH A SINGLE PRINCIPAL COMPONENT PROVIDES A SIGNIFICANT INFORMATION GAIN.

Table III shows the classification error averaged across 40 standard binary features for CelebA, using a Decision Tree with depth 1 (also known as a Stump), a Decision Tree with depth 8, and a KNN with 20 neighbors. As a baseline, choosing the majority class in each case would achieve an error rate of 20.0%. Attributes for which a single component provides a significant information gain are listed in Table IV, with reference to the principal components shown in Figure 7. According to Table IV, the images in the upper half of Figure 8 should appear more attractive, more female, with higher cheek bones, wearing lipstick and heavier makeup, compared to those in the lower half. Those in the right half should be smiling with mouth slightly open, compared to those on the left.

V. CONCLUSION

We have introduced the Eccentric Loss function and shown that it reaches its minimum on a hyperspherical distribution in dimension d with radius very close to $\sqrt{d}$.

By adjusting the scaling factor, we can force the items to adhere closely to the hypersphere, or we can enable more flexibility and tolerance of eccentricity, thus allowing the latent factors to be stratified according to their relative prominence, with potential benefits for data visualization and analysis.

Although we have demonstrated our method on image data, it could in principle be applied to other kinds of dataset, with an appropriate choice of generator and encoder network. In order to avoid explicit tuning, the scaling factor λ could instead be regulated dynamically to achieve a certain pre-determined target value for the loss function, or the eccentricity.

In future work, we plan to explore the use of eccentric regularization to encourage diversity of latent features in deep classification networks, and as a stabilizing component for recurrent network architectures.

REFERENCES

[1] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[2] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning basic visual concepts with a constrained variational framework," in *ICLR*, 2017.

[3] T. R. Davidson, L. Falorsi, N. D. Cao, T. Kipf, and J. M. Tomczak, "Hyperspherical variational auto-encoders," in *UAI*, A. Globerson and R. Silva, Eds., 2018, pp. 856–865.

[4] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf, "Wasserstein auto-encoders," in *ICLR*, 2018.

[5] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *NeurIPS*, 2017, pp. 6306–6315.

[6] P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. J. Black, and B. Schölkopf, "From variational to deterministic autoencoders," in *ICLR*, 2020.

[7] J. J. Thomson, "On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 7, no. 39, pp. 237–265, 1904.

[8] W. Liu, R. Lin, Z. Liu, L. Liu, Z. Yu, B. Dai, and L. Song, "Learning towards minimum hyperspherical energy," in *Advances in Neural Information Processing Systems*, 2018, pp. 6222–6233.

[9] Y. Yu, Y.-F. Li, and Z.-H. Zhou, "Diversity regularized machine," in *Int'l Joint Conference on Artificial Intelligence (IJCAI)*, 2011, pp. 1603–1608.

[10] Y. Bao, H. Jiang, L. Dai, and C. Liu, "Incoherent training of deep neural networks to de-correlate bottleneck features for speech recognition," in *ICASSP*, 2013, pp. 6980–6984.

[11] P. Xie, Y. Deng, and E. Xing, "Diversifying restricted Boltzmann machine for document modeling," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1315–1324.

[12] R. Lin, W. Liu, Z. Liu, C. Feng, Z. Yu, J. M. Rehg, L. Xiong, and L. Song, "Regularizing neural networks via minimizing hyperspherical energy," in *CVPR*, 2020, pp. 6917–6927.

[13] L. Jing, J. Zbontar *et al.*, "Implicit rank-minimizing autoencoder," in *NeurIPS*, 2020, pp. 14 736–46.

[14] A. Grover and S. Ermon, "Uncertainty autoencoders: Learning compressed representations via variational information maximization," in *AISTATS*, 2019, pp. 2514–2524.

[15] P. Xie, A. Singh, and E. P. Xing, "Uncorrelation and evenness: a new diversity-promoting regularizer," in *ICML*, 2017, pp. 3811–3820.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[17] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015, pp. 3730–3738.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[19] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[20] B. Dai and D. Wipf, "Diagnosing and enhancing VAE models," *arXiv preprint arXiv:1903.05789*, 2019.

[21] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *ICLR*, 2019.

[22] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *NeurIPS*, 2021, pp. 8780–8794.

APPENDIX

A. Proof of Theorem 1

The Spherical Distribution $\mathcal{S}_\rho$ will be a stationary point provided the integral, over $\mathcal{S}_\rho$, of the gradient of $K(\mathbf{z}_0, \mathbf{z}_\rho)$ is equal to zero for any arbitrary point $\mathbf{z}_0$ on the sphere, i.e.

$$\int_{\mathbf{z}_\rho \in \mathcal{S}_\rho} \nabla_{\mathbf{z}_0} K(\mathbf{z}_0, \mathbf{z}_\rho)\, d\mathbf{z}_\rho = 0.$$

Equivalently,

$$\int_{\mathcal{S}_\rho} \frac{2\mu(\mathbf{z}_0 - \mathbf{z}_\rho)}{1 + \frac{||\mathbf{z}_0 - \mathbf{z}_\rho||^2}{M}}\, d\mathbf{z}_\rho = \mathbf{z}_0.$$

By symmetry, the integral on the left hand side is a vector in the same direction as $\mathbf{z}_0$. We compute its magnitude using the angle $\theta = \cos^{-1}(-z/\rho)$, where z is the component of $\mathbf{z}_\rho$ in the direction of $\mathbf{z}_0$. The locus of points $\mathbf{z}_\rho \in \mathcal{S}_\rho$ with for which this angle is between θ and $\theta + d\theta$ is an interval of length $\rho\, d\theta$ crossed with a sphere of dimension $(d-2)$ with radius $\rho \sin\theta$ and surface area $(2\pi^{\frac{d-1}{2}}/\Gamma(\frac{d-1}{2}))(\rho\sin\theta)^{d-2}$. For each point $\mathbf{z}_\rho$, the component of $(\mathbf{z}_0 - \mathbf{z}_\rho)$ in the direction of $\mathbf{z}_0$ is $\mathbf{z}_0(1+\cos\theta)$. Furthermore,

$$||\mathbf{z}_0 - \mathbf{z}_\rho||^2 = 4\rho^2 \sin^2(\frac{\pi - \theta}{2}) = 2\rho^2(1+\cos\theta).$$

Rewriting $(\sin\theta)^{d-2}$ as $(1-\cos^2\theta)^{\frac{d-3}{2}}(\sin\theta)$ and dividing by the surface area $\rho^{d-1}2\pi^{\frac{d}{2}}/\Gamma(\frac{d}{2})$ for normalization, we have

$$\begin{aligned}
&\int_{\mathcal{S}_\rho} \frac{2\mu(\mathbf{z}_0 - \mathbf{z}_\rho)}{1 + \frac{||\mathbf{z}_0 - \mathbf{z}_\rho||^2}{M}}\, d\mathbf{z}_\rho \\
&= \mathbf{z}_0 \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d-1}{2})} \int_{\theta=0}^{\pi} \frac{2\mu(1+\cos\theta)}{\sqrt{\pi}} \frac{(1-\cos^2\theta)^{\frac{d-3}{2}}}{1 + \frac{2\rho^2(1+cos(\theta))}{M}} \sin\theta\, d\theta \\
&= \mathbf{z}_0 \frac{2\mu}{\sqrt{\pi}} \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d-1}{2})} \int_{\theta=0}^{\pi} \frac{(1+\cos\theta)^{\frac{d-1}{2}}(1-\cos\theta)^{\frac{d-3}{2}}}{1 + \frac{2\rho^2(1+cos(\theta))}{M}} \sin\theta\, d\theta.
\end{aligned}$$

We change variables to $u = 1 + \frac{1+\cos\theta}{a}$, where $a = M/(2\rho^2)$, $\rho = \sqrt{M/2a}$. Then

$$\cos\theta = a(u-1) - 1, \quad du = -\frac{1}{a}\sin\theta\, d\theta.$$

So

$$\begin{aligned}
&\int_{\mathcal{S}_\rho} \frac{2\mu(\mathbf{z}_0 - \mathbf{z}_\rho)}{1 + \frac{||\mathbf{z}_0 - \mathbf{z}_\rho||^2}{M}}\, d\mathbf{z}_\rho \\
&= \mathbf{z}_0 \frac{2a\mu}{\sqrt{\pi}} \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d-1}{2})} \int_{u=1}^{1+\frac{2}{a}} \frac{(a(u-1))^{\frac{d-1}{2}}(2-a(u-1))^{\frac{d-3}{2}}}{u}\, du.
\end{aligned}$$

Hence, the spherical distribution $\mathcal{S}_\rho$ is stationary, provided the expression on the right hand side is equal to $\mathbf{z}_0$.

B. Finding an Approximate Value for M

If d and μ are given, we would like to choose M in such a way that the radius ρ of the stable spherical distribution for $l_{\mu,M}$ is very close to $\sqrt{d}$. Let us define

$$f_{d,a}(u) = \frac{2a}{\sqrt{\pi}} \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d-1}{2})} \frac{(a(u-1))^{\frac{d-1}{2}}(2-a(u-1))^{\frac{d-3}{2}}}{u}$$

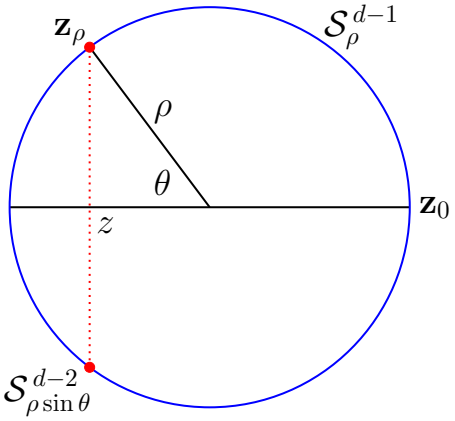


Fig. 11. The integral of $\nabla_{\mathbf{z}_0} K(\mathbf{z}_0, \mathbf{z}_\rho)$ over a $(d-1)$-dimensional sphere of radius ρ can be calculated using a series of $(d-2)$-dimensional spheres of radius $\rho\sin\theta$.

and let a_0 be the value of a for which $\int_{u=1}^{1+\frac{2}{a}} f_{d,a}(u)\, du = \frac{1}{\mu}$. It follows from Theorem 1 that $\rho = \sqrt{M/2a_0} = \sqrt{d}$ when $M = 2da_0$. We can use the following Lemma to find a good approximation a_1 for a_0 and hence a good choice $M = 2da_1$ for M.

Lemma 1:

(a) For $d \geq 3$ and $0 < a < 2$, $\int_{u=1}^{1+\frac{2}{a}} u\, f_{d,a}(u)\, du = 2$.

(b) For $d \geq 4$, the value $u_{(d,a)}$ of u for which $f_{d,a}(u)$ is maximal satisfies the equation

$$a = \big(1 + \frac{1}{u_{(d,a)}(d-3)+1}\big)/\big(u_{(d,a)} - 1\big).$$

Proof: (See below)

If we assume that $f_{d,a}$ is approximately Gaussian, then the mean value of u when averaged with weighting $f(u)$ should be approximately equal to the value $u_{(d,a)}$ at which $f(u)$ is maximal, i.e.

$$\big(\int_{u=1}^{1+\frac{2}{a}} u\, f(u)\, du\big)/\big(\int_{u=1}^{1+\frac{2}{a}} f(u)\, du\big) \simeq u_{(d,a)}.$$

If a were chosen such that $u_{(d,a)} = 2\mu$, we would have

$$\int_{u=1}^{1+\frac{2}{a}} f(u)\, du \simeq \big(\int_{u=1}^{1+\frac{2}{a}} u\, f(u)\, du\big)/u_{(d,a)} = \frac{2}{2\mu} = \frac{1}{\mu}.$$

From Part (b) of the Lemma, this corresponds to

$$a \simeq \big(1 + \frac{1}{2\mu(d-3)+1}\big)/(2\mu - 1).$$

In practice, the distribution is slightly skewed, and the true mean is a bit larger than $u_{(d,a)}$. In order to correct for this difference, we replace $(2\mu(d-3)+1)$ with $2\mu(d-1)$ in the above formula, giving us

$$M = 2da \simeq 2da_1 = 2d(1 + \frac{1}{2\mu(d-1)})/(2\mu - 1).$$

Proof of Lemma 1:

(a) We prove equivalently that $F_{d,a} = G_{d,a}$ where

$$F_{d,a} = \int_{u=1}^{1+\frac{2}{a}} (a(u-1))^{\frac{d-1}{2}} (2-a(u-1))^{\frac{d-3}{2}}\, du,$$
$$G_{d,a} = \frac{\sqrt{\pi}}{a} \frac{\Gamma(\frac{d-1}{2})}{\Gamma(\frac{d}{2})}\,.$$

Change variables to

$$a(u-1) = \frac{2}{t^2+1}, \quad du = \frac{-4t}{a(t^2+1)^2}\, dt.$$

Then

$$F_{d,a} = \frac{1}{a}\int_0^\infty \Big(\frac{2}{t^2+1}\Big)^{\frac{d-1}{2}} \Big(\frac{2t^2}{t^2+1}\Big)^{\frac{d-3}{2}} \frac{4t}{(t^2+1)^2}\, dt$$
$$= \frac{2^d}{a}\int_0^\infty \frac{t^{d-2}}{(t^2+1)^d}\, dt.$$

If we set

$$J(k,n) = \int_0^\infty \frac{t^k}{(t^2+1)^n}\, dt,$$

then $J(0,1) = \frac{\pi}{2}$, $J(1,n) = \frac{1}{2(n-1)}$ for $n \geq 2$ and we can derive these recurrence relations for $n, k \geq 2$:

$$J(k,n) = \frac{k-1}{2(n-1)}\, J(k-2, n-1), \quad J(0,n) = \frac{2n-3}{2n-2}\, J(0, n-1).$$

It follows that when $d \geq 2$ is even,

$$F_{d,a} = \frac{2^d}{a} J(d-2,d) = \frac{2^d}{a} \frac{(d-3)!!(\frac{d}{2})!}{2^{\frac{d-2}{2}}(d-1)!}\, J(0, \tfrac{d+2}{2})$$
$$= \frac{2^d}{a} \frac{(d-3)!!(\frac{d}{2})!}{2^{\frac{d-2}{2}}(d-1)!} \frac{(d-1)!!}{d!!} \frac{\pi}{2}$$
$$= \frac{\pi}{a} \frac{(d-3)!!\; 2^{\frac{d}{2}}(\frac{d}{2})!(d-1)!!}{(d-2)!!\quad d\quad (d-1)!}$$
$$= \frac{\pi}{a} \frac{(d-3)!!}{(d-2)!!} = G_{d,a}\,.$$

When $d \geq 3$ is odd,

$$F_{d,a} = \frac{2^d}{a} J(d-2,d) = \frac{2^d}{a} \frac{(d-3)!!(\frac{d+1}{2})!}{2^{\frac{d-3}{2}}(d-1)!}\, J(1, \tfrac{d+3}{2})$$
$$= \frac{2}{a} \frac{(d-3)!!\; 2^{\frac{d+1}{2}}(\frac{d+1}{2})!}{(d-1)!\quad 2(\frac{d+1}{2})}$$
$$= \frac{2}{a} \frac{(d-3)!!\;(d+1)!!}{(d-1)!\;\;(d+1)}$$
$$= \frac{2}{a} \frac{(d-3)!!}{(d-2)!!} = G_{d,a}\,.$$

(b) When $d \geq 4$, the derivative of $f_{d,a}(u)$ is $C_{(d,a)} D_{(d,a)}$ where

$$C_{(d,a)} = \frac{-2a^2}{\sqrt{\pi}} \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{d-1}{2})} (a(u-1))^{\frac{d-3}{2}} (2-a(u-1))^{\frac{d-5}{2}},$$
$$D_{(d,a)} = (a(u-1)-1)(u(d-3)+1) - 1\,.$$

This derivative will be zero when $D_{(d,a)} = 0$, i.e. when

$$a = \big(1 + \frac{1}{u(d-3)+1}\big)/\big(u-1\big).$$

C. Pytorch code for Eccentric Loss function

```
def EccentricLoss( z, scale=1 ):
 x0 = torch.squeeze(z)
 x1 = x0.transpose(0,1)
 batch= x0.size()[0]
 dim  = x0.size()[1]
 norm = 2*dim*(1 + 1/(2*scale*(dim-1)))/(2*scale-1)
 xx = torch.bmm(x0.view(batch,1,dim),
                x0.view(batch,dim,1)).squeeze(2)
 xx0 = xx.expand(batch,batch)
 xx1 = xx0.transpose(0,1)
 xy = xx0 + xx1 - 2*torch.matmul(x0,x1)
 result = torch.sum(xx) - scale*norm* \
          torch.sum(torch.log(1+xy/norm))/batch
 return result/(batch-1)
```

D. Network Architectures

Encoder architecture for MNIST:

$$\begin{aligned} x \in \mathcal{R}^{32\times32\times1} &\to \mathrm{Conv}_{16}^{5(1)} \to \mathrm{BN} \to \mathrm{LeakyReLU}_{(0.1)} \\ &\to \mathrm{Conv}_{24}^{4(2)} \to \mathrm{BN} \to \mathrm{LeakyReLU}_{(0.1)} \\ &\to \mathrm{Conv}_{32}^{4(1)} \to \mathrm{BN} \to \mathrm{LeakyReLU}_{(0.1)} \\ &\to \mathrm{Conv}_{48}^{4(2)} \to \mathrm{BN} \to \mathrm{LeakyReLU}_{(0.1)} \\ &\to \mathrm{FC}_{64} \to \mathrm{FC}_d \end{aligned}$$

Decoder architecture for MNIST:

$$\begin{aligned} x \in \mathcal{R}^{d} &\to \mathrm{FSConv}_{48}^{3(1)} \to \mathrm{BN} \to \mathrm{LeakyReLU}_{(0.1)} \\ &\to \mathrm{FSConv}_{32}^{4(2)} \to \mathrm{BN} \to \mathrm{LeakyReLU}_{(0.1)} \\ &\to \mathrm{FSConv}_{24}^{4(1)} \to \mathrm{BN} \to \mathrm{LeakyReLU}_{(0.1)} \\ &\to \mathrm{FSConv}_{16}^{4(2)} \to \mathrm{BN} \to \mathrm{LeakyReLU}_{(0.1)} \\ &\to \mathrm{FSConv}_{16}^{5(1)} \to \mathrm{BN} \to \mathrm{LeakyReLU}_{(0.1)} \\ &\to \mathrm{FSConv}_{16}^{1(1)} \to \mathrm{BN} \to \mathrm{LeakyReLU}_{(0.1)} \\ &\to \mathrm{FSConv}_{1}^{1} \to \mathrm{Sigmoid} \end{aligned}$$

Encoder architecture for CelebA:

$$\begin{aligned} x \in \mathcal{R}^{64\times64\times3} &\to \mathrm{Conv}_{128}^{4(2)} \to \mathrm{BN} \to \mathrm{ReLU} \\ &\to \mathrm{Conv}_{256}^{4(2)} \to \mathrm{BN} \to \mathrm{ReLU} \\ &\to \mathrm{Conv}_{512}^{4(2)} \to \mathrm{BN} \to \mathrm{ReLU} \\ &\to \mathrm{Conv}_{1024}^{4(2)} \to \mathrm{BN} \to \mathrm{ReLU} \to \mathrm{FC}_{64} \end{aligned}$$

Decoder architecture for CelebA:

$$\begin{aligned} x \in \mathcal{R}^{64} &\to \mathrm{FC}_{8\times8\times1024} \\ &\to \mathrm{FSConv}_{512}^{4(2)} \to \mathrm{BN} \to \mathrm{ReLU} \\ &\to \mathrm{FSConv}_{256}^{4(2)} \to \mathrm{BN} \to \mathrm{ReLU} \\ &\to \mathrm{FSConv}_{128}^{4(2)} \to \mathrm{BN} \to \mathrm{ReLU} \\ &\to \mathrm{FSConv}_{3}^{1} \to \mathrm{Sigmoid} \end{aligned}$$