Rate-Distortion Optimization Guided Autoencoder for Isometric Embedding in Euclidean Latent Space

Keizo Kato [1] **Jing Zhou** [2] **Tomotake Sasaki** [1] **Akira Nakagawa** [1]

Abstract

To analyze high-dimensional and complex data in the real world, deep generative models, such as variational autoencoder (VAE) embed data in a low-dimensional space (latent space) and learn a probabilistic model in the latent space. However, they struggle to accurately reproduce the probability distribution function (PDF) in the input space from that in the latent space. If the embedding were isometric, this issue can be solved, because the relation of PDFs can become tractable. To achieve isometric property, we propose Rate-Distortion Optimization guided autoencoder inspired by orthonormal transform coding. We show our method has the following properties: (i) the Jacobian matrix between the input space and a Euclidean latent space forms a constantly-scaled orthonormal system and enables isometric data embedding; (ii) the relation of PDFs in both spaces can become tractable one such as proportional relation. Furthermore, our method outperforms state-of-the-art methods in unsupervised anomaly detection with four public datasets.

1. Introduction

Capturing the inherent features of a dataset from high-dimensional and complex data is an essential issue in machine learning. Generative model approach learns the probability distribution of data, aiming at data generation, unsupervised learning, disentanglement, etc. (Tschannen et al., 2018). It is generally difficult to directly estimate a probability density function (PDF) $P_{\boldsymbol{x}}(\boldsymbol{x})$ of high-dimensional data $\boldsymbol{x} \in \mathbb{R}^M$. Instead, one promising approach is to map $\boldsymbol{x}$ to a low-dimensional latent variable $\boldsymbol{z} \in \mathbb{R}^N$ ($N < M$), and capture PDF $P_{\boldsymbol{z}}(\boldsymbol{z})$. Variational autoencoder (VAE) is a widely used generative model to capture $\boldsymbol{z}$ as a probabilistic model with univariate Gaussian priors (Kingma & Welling, 2014). For a more flexible estimation of $P_{\boldsymbol{z}}(\boldsymbol{z})$, successor models have been proposed, such as using Gaussian mixture model (GMM) (Zong et al., 2018), combining univariate Gaussian model and GMM (Liao et al., 2018), etc.

In tasks where the quantitative analysis is vital, $P_{\boldsymbol{x}}(\boldsymbol{x})$ should be reproduced from $P_{\boldsymbol{z}}(\boldsymbol{z})$. For instance, in anomaly detection, the anomaly likelihood is calculated based on PDF value of data sample (Chalapathy & Chawla, 2019). However, the embedding of VAEs is not isometric; that is, the distance between data points $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ is inconsistent to the distance of corresponding latent variables $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$ (Chen et al., 2018; Shao et al., 2018; Geng et al., 2020). Obviously mere estimation of $P_{\boldsymbol{z}}(\boldsymbol{z})$ cannot be the substitution of the estimation for $P_{\boldsymbol{x}}(\boldsymbol{x})$ under such situation. As McQueen et al. (2016) mentioned, for a reliable data analysis, the isometric embedding in low-dimensional space is necessary. In addition, to utilize the standard PDF estimation techniques, the latent space is preferred to be a Euclidean space. Despite of its importance, this point is not considered even in methods developed for the quantitative analysis of PDF (Johnson et al., 2016; Zong et al., 2018; Liao et al., 2018; Zenati et al., 2018; Song & Ou, 2018).

According to the Nash embedding theorem, an arbitrary smooth and compact Riemannian manifold $\mathcal{M}$ can be embedded in a Euclidean space $\mathbb{R}^N$ ($N \geq \dim \mathcal{M} + 1$, sufficiently large) isometrically (Han & Hong, 2006). On the other hand, the manifold hypothesis argues that real-world data presented in a high-dimensional space concentrate in the vicinity of a much lower dimensional manifold $\mathcal{M}_{\boldsymbol{x}} \subset \mathbb{R}^M$ (Bengio et al., 2013). Based on these theories, it is expected that the input data $\boldsymbol{x}$ can be embedded isometrically in a low-dimensional Euclidean space $\mathbb{R}^N$ when $\dim \mathcal{M}_{\boldsymbol{x}} < N \ll M$. Although the existence of the isometric embedding was proven, the method to achieve it has been challenging. Some previous works have proposed algorithms to do that (McQueen et al., 2016; Bernstein et al., 2000). Yet, they do not deal with high-dimensional input data, such as images. Another thing to consider is the distance on $\mathcal{M}_{\boldsymbol{x}}$ may be defined by the data tendency with an appropriate metric function. For instance, we can choose the binary cross entropy (BCE) for binary data and structured

[1]FUJITSU LABORATORIES LTD. [2]Fujitsu R&D Center Co., Ltd.. Correspondence to: kato.keizo, anaka <@fujitsu.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

similarity (SSIM) for image. As a whole, our challenge is to develop a deep generative model that guarantees the isometric embedding even for the high-dimensional data observed around $\mathcal{M}_{\boldsymbol{x}}$ endowed with a variety of metric function.

Mathematically, the condition of isometric embedding to Euclidean space is equivalent to that the columns of the Jacobian matrix between two spaces form an orthonormal system. When we turn our sight to conventional image compression area, orthonormal transform is necessary for an efficient compression. This is proven by rate-distortion (RD) theory (Berger, 1971) . Furthermore, the empirical method for optimal compression with orthonormal transform coding is established as rate-distortion optimization (RDO) (Sullivan & Wiegand, 1998). It is intuitive to regard data embedding to a low-dimensional latent space as an analogy of efficient data compression. Actually, deep learning based image compression (DIC) methods with RDO (Ballé et al., 2018; Zhou et al., 2019) have been proposed and they have achieved good compression performance. Although it is not discussed in Ballé et al. (2018); Zhou et al. (2019), we guess that behind the success of DIC, there should be theoretical relation to RDO of convetional transform coding.

Hence, in this study, we investigate the theoretical property and dig out the proof that RDO guides deep-autoencoders to have the orthonormal property. Based on this finding, we propose a method that enables isometric data embedding and allows a comprehensive data analysis, named Rate-Distortion Optimization Guided Autoencoder for Generative Analysis (RaDOGAGA). We show the validity of RaDOGAGA in the following steps.

(1) We show that RaDOGAGA has the following properties both theoretically and experimentally.

- The Jacobian matrix between the data observation space (inner product space endowed with a metric tensor) and latent space forms a constantly-scaled orthonormal system. Thus, data can be embedded in a Euclidean latent space isometrically.
- Thanks to the property above, the relation of $P_{\boldsymbol{z}}(\boldsymbol{z})$ and $P_{\boldsymbol{x}}(\boldsymbol{x})$ can become tractable one (e.g., proportional relation). Thus, PDF of $\boldsymbol{x}$ in the data observation space can be estimated by maximizing log-likelihood of parametric PDF $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ in the low-dimensional Euclidean space.

(2) Thanks to (1), RaDOGAGA outperforms the current state-of-the-art method in unsupervised anomaly detection task with four public datasets.

Isometric Map and Notions of Differential Geometry

Here, we explain notions of differential geometry adopted to our context. Given two Riemannian manifolds $\mathcal{M} \subset \mathbb{R}^M$ and $\mathcal{N} \subset \mathbb{R}^N$, a map $g : \mathcal{M} \to \mathcal{N}$ is called isometric if

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle_{\boldsymbol{p}} = \langle \mathrm{d}g(\boldsymbol{v}), \mathrm{d}g(\boldsymbol{w}) \rangle_{g(\boldsymbol{p})} \tag{1}$$

holds. Here, $\boldsymbol{v}$ and $\boldsymbol{w}$ are tangent vectors in $T_{\boldsymbol{p}}\mathcal{M}$ (tangent space of $\mathcal{M}$ at $\boldsymbol{p} \in \mathcal{M}$) represented as elements of $\mathbb{R}^M$ and $\mathrm{d}g$ is the differential of g (this can be written as a Jacobian matrix). $\langle \boldsymbol{v}, \boldsymbol{w} \rangle_{\boldsymbol{p}} = \boldsymbol{v}^\top \boldsymbol{A}_{\mathcal{M}}(\boldsymbol{p}) \boldsymbol{w}$, where $\boldsymbol{A}_{\mathcal{M}}(\boldsymbol{p}) \in \mathbb{R}^{M \times M}$ is a metric tensor represented as a positive define matrix. The inner product in the right side is also defined by another metric tensor $\boldsymbol{A}_{\mathcal{N}}(\boldsymbol{q}) \in \mathbb{R}^{N \times N}$. $\boldsymbol{A}_{\mathcal{M}}(\boldsymbol{p})$ or $\boldsymbol{A}_{\mathcal{N}}(\boldsymbol{q})$ is an identity matrix for a Euclidean case and the inner product becomes the standard one (the dot product).

We slightly abuse the terminology and call a map g isometric if the following relation holds for some constant $C > 0$:

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle_{\boldsymbol{p}} = C \langle \mathrm{d}g(\boldsymbol{v}), \mathrm{d}g(\boldsymbol{w}) \rangle_{g(\boldsymbol{p})}, \tag{2}$$

since Eq. (1) is achieved by replacing g with $\tilde{g} = (1/\sqrt{C})g$.

2. Related Work

Flow-based model: Flow-based generative models generate images with astonishing quality (Kingma & Dhariwal, 2018; Dinh et al., 2015). Flow mechanism explicitly takes the Jacobian of $\boldsymbol{x}$ and $\boldsymbol{z}$ into account. The transformation function $\boldsymbol{z} = f(\boldsymbol{x})$ is learned, calculating and storing the Jacobian of $\boldsymbol{x}$ and $\boldsymbol{z}$. Unlike ordinary autoencoders, which reverse $\boldsymbol{z}$ to $\boldsymbol{x}$ with function $g(\cdot)$ different from $f(\cdot)$, inverse function transforms $\boldsymbol{z}$ to $\boldsymbol{x}$ as $\boldsymbol{x} = f^{-1}(\boldsymbol{z})$. Since the model stores Jacobian, $P_{\boldsymbol{x}}(\boldsymbol{x})$ can be estimated from $P_{\boldsymbol{z}}(\boldsymbol{z})$. However, in these approaches, the form of $f(\cdot)$ is limited so that the explicit calculation of Jacobian is manageable, such as $f(\cdot)$ cannot reduce the dimension of $\boldsymbol{x}$.

Data interpolation with autoencoders: For a smooth data interpolation, in Chen et al. (2018); Shao et al. (2018), a function learns to map latent variables to a geodesic (shortest path in a manifold) space, in which the distance corresponds to the metric of the data space. In Geng et al. (2020); Pai et al. (2019), a penalty for the anisometricity of a map is added to training loss. Although these approaches may remedy scale inconsistency, they do not deal with PDF estimation. Furthermore, the distance for the input data is assumed to be a Euclidean distance and the cases for other distances are not considered.

Deep image compression (DIC) with RDO: RD theory is a part of Shannon's information theory for lossy compression which formulates the optimal condition between information rate and distortion. The signal is decorrelated by orthonormal transformation such as Karhunen-Loève transform (KLT) (Rao & Yip, 2000) and discrete cosine transform (DCT). In RDO, a cost $L = R + \lambda D$ is minimized at given Lagrange parameter λ. Recently, DIC methods with RDO (Ballé et al., 2018; Zhou et al., 2019) have been proposed. In these works, instead of orthonormal transform in the conventional lossy compression method, a deep autoencoder is trained for RDO. In the next section, we explain the idea of

RDO guided autoencoder and its relationship with VAE.

3. Overview of RDO Guided Approach

3.1. Derivation from RDO in Transform Coding

Figure 1 shows the overview of our idea based on the RDO inspired by transform coding. In the transform coding, the optimal method to encode data with Gaussian distribution is as follows (Goyal, 2001). First, the data are transformed deterministically to decorrelated data using orthonormal transforms such as Karhunen-Loève transform (KLT) and discrete cosine transform (DCT). Then these decorrelated data are quantized stochastically with uniform quantizer for all channels such that the quantization noise for each channel is equal. Lastly optimal entropy encoding is applied to quantized data where the rate can be calculated by the logarithm of symbol's estimated probability. From this fact, we have an intuition that the columns of the Jacobian matrix of the autoencoder forms an orthonormal system if the data were compressed based on RDO with a uniform quantized noise and parametric distribution of latent variables. Inspired by this, we propose autoencoder which scales latent variables according to the definition of metrics of data.

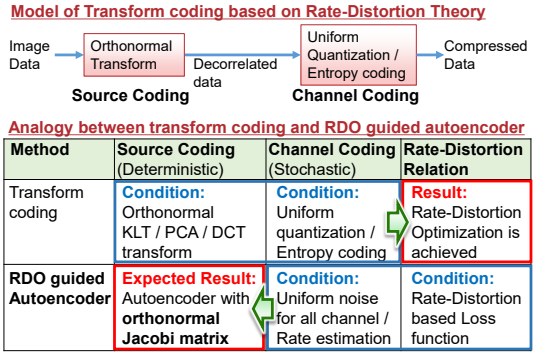


Figure 1. Overview of our idea. Orthonormal transformation and uniform quantization noise result in an RDO. Our idea is that uniform quantization noise and RDO make an antoencoder to be orthonormal.

3.2. Relationship with VAE

There is a number of VAE studies taking RD trade-off into account. In VAEs, it is common to maximize ELBO instead of maximizing log-likelihood of $P_{\boldsymbol{x}}(\boldsymbol{x})$ directly. In beta-VAE (Higgins et al., 2017), the objective function L_{VAE} is described as $L_{VAE} = L_{rec} - \beta L_{kl}$. Here, L_{kl} is the KL divergence between the encoder output and prior distribution, usually a Gaussian distribution. By changing β, the rate-distortion trade-off at desirable rate can be realized as discussed in Alemi et al. (2018).

Note that the beta-VAE and the RDO in image compression are analogous to each other. That is, β^{-1}, $-L_{kl}$, and L_{rec} correspond to λ, a rate R, and a distortion D respectively. However, the probability models of latent variables are quite different. VAE uses a fixed prior distribution. This causes a nonlinear scaling relationship between real data and latent variables. Figure 2 shows the conditions to achieve RDO in both VAE and RaDOGAGA. In VAE, for RDO condition, a nonlinear scaling of the data distribution is necessary to fit prior. To achieve it, Brekelmans et al. (2019) suggested to precisely control noise as a posterior variance for each channel.

As proven in Rolínek et al. (2019), in the optimal condition, the Jacobian matrix of VAE forms an orthogonal system, but the norm is not constant. In RaDOGAGA, uniform quantization noises are added to all channels. Instead, a parametric probability distribution should be estimated as a prior. As a result, the Jacobian matrix forms an orthonormal system because both orthogonality and scaling normalization are simultaneously achieved. As discussed above, the precise noise control in VAE and parametric prior optimization in RaDOGAGA are essentially the same. Accordingly, complexities in methods are estimated to be at the same degree.

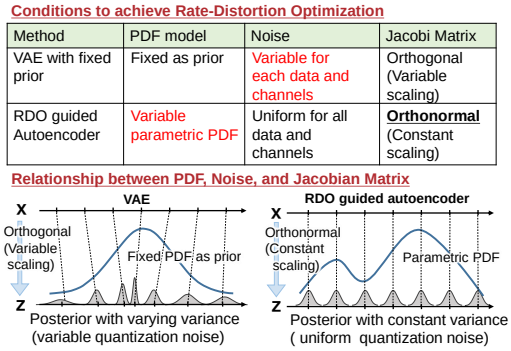


Figure 2. The condition of RDO in VAE and our method. In VAE, to fit the fixed prior (blue line), data are transformed anisometrically with precisely controlled noise as a posterior variance (gray area). A wider distribution of noise makes the PDF of transformed data smaller. In our method, a parametric prior distribution is estimated, and data is transformed isometrically with uniform noise.

4. METHOD AND THEORETICAL PROPERTIES

4.1. Method

Our method is based on the RDO of the autoencoder for the image compression proposed in Ballé et al. (2018) with some modifications. In Ballé et al. (2018), the cost function

$$L = R + \lambda D \tag{3}$$

consists of (i) reconstruction error D between input data and decoder output with noise to latent variable and (ii) rate R of latent variable. This is analogous to beta-VAE where $\lambda = \beta^{-1}$.

Figure 3 depicts the architecture of our method. The details are given in the following. Let $\boldsymbol{x}$ be an M-dimensional input data, $\mathbb{R}^M$ be a data observation space endowed with a metric function $D(\cdot\ ,\ \cdot)$, and $P_{\boldsymbol{x}}(\boldsymbol{x})$ be the PDF of $\boldsymbol{x}$. Let $f_\theta(\boldsymbol{x})$, $g_\phi(\boldsymbol{z})$, and $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ be the parametric encoder, decoder, and PDF of the latent variable with parameters θ, ϕ, and ψ. Note that both of the encoder and decoder are deterministic, while the encoder of VAE is stochastic.

First, the encoder converts the input data $\boldsymbol{x}$ to an N-dimensional latent variable $\boldsymbol{z}$ in a Euclidean latent space $\mathbb{R}^N$, and then the decoder converts $\boldsymbol{z}$ to the decoded data $\hat{\boldsymbol{x}} \in \mathbb{R}^M$:

$$\boldsymbol{z} = f_\theta(\boldsymbol{x}), \quad \hat{\boldsymbol{x}} = g_\phi(\boldsymbol{z}). \tag{4}$$

Let $\boldsymbol{\epsilon} \in \mathbb{R}^N$ be a noise vector to emulate uniform quantization, where each component is independent from others and has an equal mean 0 and an equal variance σ^2:

$$\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ ..\epsilon_N),\ E\left[\epsilon_i\right] = 0,\ E\left[\epsilon_i \epsilon_j\right] = \delta_{ij}\sigma^2. \tag{5}$$

Here, δ_{ij} denotes the Kronecker's delta. Given the sum of latent variable $\boldsymbol{z}$ and noise $\boldsymbol{\epsilon}$, another decoder output $\breve{\boldsymbol{x}} \in \mathbb{R}^M$ is obtained as

$$\breve{\boldsymbol{x}} = g_\phi(\boldsymbol{z} + \boldsymbol{\epsilon}) \tag{6}$$

with the same parameter ϕ used to obtain $\hat{\boldsymbol{x}}$. This is analogous to the stochastic sampling and decoding procedure in VAE.

The cost function is defined based on Eq. (3) with some modifications as follows:

$$L = -\log(P_{\boldsymbol{z},\psi}(\boldsymbol{z})) + \lambda_1 h\left(D\left(\boldsymbol{x}, \hat{\boldsymbol{x}}\right)\right) + \lambda_2 D\left(\hat{\boldsymbol{x}}, \breve{\boldsymbol{x}}\right). \tag{7}$$

The first term corresponds to the estimated rate of the latent variable. We can use arbitrary probabilistic model as $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$. For example, Ballé et al. (2018) uses univariate independent (factorized) model $P_{\boldsymbol{z},\psi}(\boldsymbol{z}) = \prod_{i=1}^{N} P_{z_i,\psi}(z_i)$. In this work, a parametric function $c_\psi(z_i)$ outputs cumulative distribution function of z_i. A rate for quantized symbol is calculated by $c_\psi(z+\frac{1}{2}) - c_\psi(z-\frac{1}{2})$, assuming the symbol is quantized with the side length of 1. A model based on GMM like Zong et al. (2018) is another instance.

The second and the third term in Eq. (7) is based on the decomposition $D\left(\boldsymbol{x}, \breve{\boldsymbol{x}}\right) \simeq D\left(\boldsymbol{x}, \hat{\boldsymbol{x}}\right) + D\left(\hat{\boldsymbol{x}}, \breve{\boldsymbol{x}}\right)$ shown in Rolínek et al. (2019). The second term in Eq. (7) purely calculate reconstruction loss as an autoencoder. In the RDO, the consideration is trade-off between rate (the first term) and the distortion by the quantization noise (the third term). By this decomposition, we can avoid the interference between better reconstruction and RDO trade-off durning the training. The weight λ_1 controls the degree of reconstruction, and λ_2 ($\simeq \beta^{-1}$ of beta-VAE) controls a scaling between data and latent spaces respectively.

The function $h(\cdot)$ in the second term of Eq. (7) is a monotonically increasing function. In experiments in this paper, we use $h(d) = \log(d)$. In the theory shown in Appendix A, better reconstruction provide much rigid orthogonality. We find $h(d) = \log(d)$ is much more appropriate for this purpose than $h(d) = d$ as detailed in Appendix C.

The properties of our method shown in the rest of this paper hold for a variety of metric function $D(\cdot\ ,\ \cdot)$, as long as it can be approximated by the following quadratic form in the neighborhood of $\boldsymbol{x}$:

$$D(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{\Delta x}) \simeq \boldsymbol{\Delta x}^\top\ \boldsymbol{A}(\boldsymbol{x})\ \boldsymbol{\Delta x}. \tag{8}$$

Here, $\boldsymbol{\Delta x}$ is an arbitrary infinitesimal variation of $\boldsymbol{x}$, and $\boldsymbol{A}(\boldsymbol{x})$ is an $M \times M$ positive definite matrix depending on $\boldsymbol{x}$ that corresponds to a metric tensor. When $D(\cdot\ ,\ \cdot)$ is the square of the Euclidean distance, $\boldsymbol{A}(\boldsymbol{x})$ is the identity matrix. For another instance, a cost with structure similarity (SSIM (Wang et al., 2004)) and binary cross entropy (BCE) can also be approximated by a quadratic form as explained in Appendix D. By deriving parameters that minimize the average of Eq. (7) according to $\boldsymbol{x} \sim P_{\boldsymbol{x}}(\boldsymbol{x})$ and $\boldsymbol{\epsilon} \sim P_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})$, the encoder, decoder, and probability distribution of the latent space are trained as

$$\theta, \phi, \psi = \underset{\theta,\phi,\psi}{\arg\min}(E_{\boldsymbol{x}\sim P_{\boldsymbol{x}}(\boldsymbol{x}),\ \boldsymbol{\epsilon}\sim P_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})}[\ L\ ]). \tag{9}$$

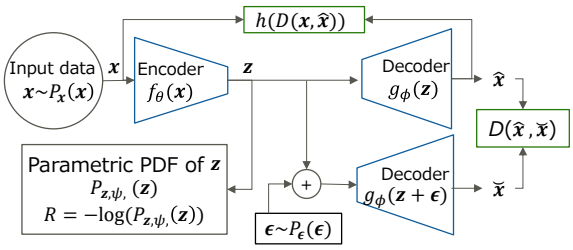


Figure 3. Architecture of RaDOGAGA

4.2. Theoretical Properties

In this section, we explain the theoretical properties of the method. To show the essence in a simple form, we first (formally) consider the case $M = N$. The theory for $M > N$ is then outlined. All details are given in Appendices.

We begin with examining the condition to minimize the loss function analytically, assuming that the reconstruction part

is trained enough so that $\boldsymbol{x} \simeq \hat{\boldsymbol{x}}$. In this case, the second term in Eq. (7) can be ignored. Let $\boldsymbol{J}(\boldsymbol{z}) = \partial \boldsymbol{x}/\partial \boldsymbol{z} = \partial g_\phi(\boldsymbol{z})/\partial \boldsymbol{z} \in \mathbb{R}^{N\times N}$ be the Jacobian matrix between the data space and latent space, which is assumed to be full-rank at every point. Then, $\breve{\boldsymbol{x}} - \hat{\boldsymbol{x}}$ can be approximated as $\acute{\boldsymbol{\epsilon}} = \sum_{i=1}^N \epsilon_i (\partial \boldsymbol{x}/\partial z_i)$ through the Taylor expansion. By applying $E[\epsilon_i \epsilon_j] = \sigma^2 \delta_{ij}$ and Eq. (8), the expectation of the third term in Eq. (7) is rewritten as

$$\underset{\boldsymbol{\epsilon}\sim P_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})}{E}\left[\acute{\boldsymbol{\epsilon}}^\top \boldsymbol{A}(\boldsymbol{x})\acute{\boldsymbol{\epsilon}}\right] = \sigma^2 \sum_{j=1}^N \left(\frac{\partial \boldsymbol{x}}{\partial z_j}\right)^\top \boldsymbol{A}(\boldsymbol{x}) \left(\frac{\partial \boldsymbol{x}}{\partial z_j}\right). \quad (10)$$

As is well known, the relation between $P_{\boldsymbol{z}}(\boldsymbol{z})$ and $P_{\boldsymbol{x}}(\boldsymbol{x})$ in such case is described as $P_{\boldsymbol{z}}(\boldsymbol{z}) = |\det(\boldsymbol{J}(\boldsymbol{z}))| P_{\boldsymbol{x}}(\boldsymbol{x})$. The expectation of L in Eq. (7) is thus approximated as

$$\begin{aligned}\underset{\boldsymbol{\epsilon}\sim P_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})}{E}[L] \simeq &-\log(|\det(\boldsymbol{J}(\boldsymbol{z}))|) - \log(P_{\boldsymbol{x}}(\boldsymbol{x})) \\ &+ \lambda_2 \sigma^2 \sum_{j=1}^N \left(\frac{\partial \boldsymbol{x}}{\partial z_j}\right)^\top \boldsymbol{A}(\boldsymbol{x}) \left(\frac{\partial \boldsymbol{x}}{\partial z_j}\right). \quad (11)\end{aligned}$$

By differentiating Eq. (11) by $\partial \boldsymbol{x}/\partial z_j$, the following equation is derived as a condition to minimize the expected loss:

$$2\lambda_2 \sigma^2 \boldsymbol{A}(\boldsymbol{x}) \left(\frac{\partial \boldsymbol{x}}{\partial z_j}\right) = \frac{1}{\det(\boldsymbol{J}(\boldsymbol{z}))} \tilde{\boldsymbol{J}}(\boldsymbol{z})_{:,j}, \quad (12)$$

where $\tilde{\boldsymbol{J}}(\boldsymbol{z})_{:,j} \in \mathbb{R}^N$ is the j-th column vector of the cofactor matrix of $\boldsymbol{J}(\boldsymbol{z})$. Due to the trait of cofactor matrix, $(\partial \boldsymbol{x}/\partial z_i)^\top \tilde{\boldsymbol{J}}(\boldsymbol{z})_{:,j} = \delta_{ij} \det(\boldsymbol{J}(\boldsymbol{z}))$ holds. Thus, the following relationship is obtained by multiplying Eq. (12) by $(\partial \boldsymbol{x}/\partial z_i)^\top$ from the left and rearranging the results:

$$\left(\frac{\partial \boldsymbol{x}}{\partial z_i}\right)^\top \boldsymbol{A}(\boldsymbol{x}) \left(\frac{\partial \boldsymbol{x}}{\partial z_j}\right) = \frac{1}{2\lambda_2 \sigma^2} \delta_{ij}. \quad (13)$$

This means that *the columns of the Jacobian matrix of two spaces form a constantly-scaled orthonormal system with respect to the inner product defined by $\boldsymbol{A}(\boldsymbol{x})$ for all $\boldsymbol{z}$.*

Given tangent vectors $\boldsymbol{v}_z$ and $\boldsymbol{w}_z$ in the tangent space of $\mathbb{R}^N$ at $\boldsymbol{z}$ represented as elements of $\mathbb{R}^N$, let $\boldsymbol{v}_x$ and $\boldsymbol{w}_x$ be the corresponding tangent vectors represented as elements of $\mathbb{R}^M = \mathbb{R}^N$. The following relation holds due to Eq. (13), which means that *the map is isometric in the sense of Eq. (2)*:

$$\begin{aligned}\boldsymbol{v}_x \boldsymbol{A}(\boldsymbol{x}) \boldsymbol{w}_x &= \sum_{i=0}^N \sum_{j=0}^N \left(\frac{\partial \boldsymbol{x}}{\partial z_i} v_{zi}\right)^\top \boldsymbol{A}(\boldsymbol{x}) \left(\frac{\partial \boldsymbol{x}}{\partial z_j} w_{zj}\right) \\ &= \frac{1}{2\lambda_2 \sigma^2} \sum_{i=0}^N v_{z_i} w_{z_i} = \frac{1}{2\lambda_2 \sigma^2} \boldsymbol{v}_z \cdot \boldsymbol{w}_z. \quad (14)\end{aligned}$$

Since $f_\theta(\cdot)$ and $g_\phi(\cdot)$ acts like the inverse functions of each other when restricted on the input data, isometric property holds for both.

Even for the case $M > N$, equations in the same form as Eqs. (13) and (14) can be derived essentially in the same manner (Appendix A); that is, *RaDOGAGA achieves isometric data embedding for the case $M > N$ as well.*

Now let us proceed to PDF estimation. First, we (formally) consider the case $M = N$ as before. Note that Eq. (13) can be expressed as follows: $\boldsymbol{J}(\boldsymbol{z})^\top \boldsymbol{A}(\boldsymbol{x}) \boldsymbol{J}(\boldsymbol{z}) = (1/2\lambda_2\sigma^2)\boldsymbol{I}_N$ ($\boldsymbol{I}_N$ is the $N \times N$ identity matrix). We have the following equation by taking the determinants of both sides of this and using the properties of the determinant: $|\det(\boldsymbol{J}(\boldsymbol{z}))| = (1/2\lambda_2\sigma^2)^{N/2} \det(\boldsymbol{A}(\boldsymbol{x}))^{-1/2}$. Note that $\det(\boldsymbol{A}(\boldsymbol{x})) = \prod_{j=1}^N \alpha_j(\boldsymbol{A}(\boldsymbol{x}))$, where $0 < \alpha_1(\boldsymbol{A}(\boldsymbol{x})) \leq \cdots \leq \alpha_N(\boldsymbol{A}(\boldsymbol{x}))$ are the eigenvalues of $\boldsymbol{A}(\boldsymbol{x})$. Thus, $P_{\boldsymbol{z}}(\boldsymbol{z})$ and $P_{\boldsymbol{x}}(\boldsymbol{x})$ are related in the following form:

$$P_{\boldsymbol{x}}(\boldsymbol{x}) = \left(\frac{1}{2\lambda_2\sigma^2}\right)^{-\frac{N}{2}} \left(\prod_{j=1}^N \alpha_j(\boldsymbol{A}(\boldsymbol{x}))\right)^{\frac{1}{2}} P_{\boldsymbol{z}}(\boldsymbol{z}). \quad (15)$$

To consider the relationship of $P_{\boldsymbol{z}}(\boldsymbol{z})$ and $P_{\boldsymbol{x}}(\boldsymbol{x})$ for $M > N$, we follow the manifold hypothesis and assume the situation where the data $\boldsymbol{x}$ substantially exist in the vicinity of a low-dimensional manifold $\mathcal{M}_{\boldsymbol{x}}$, and $\boldsymbol{z} \in \mathbb{R}^N$ can sufficiently capture its feature. In such case, we can regard that the distribution of $\boldsymbol{x}$ away from $\mathcal{M}_{\boldsymbol{x}}$ is negligible and the ratio of $P_{\boldsymbol{z}}(\boldsymbol{z})$ and $P_{\boldsymbol{x}}(\boldsymbol{x})$ is equivalent to that of the volumes of corresponding regions in $\mathbb{R}^N$ and $\mathbb{R}^M$. This ratio is shown to be $J_{sv}(\boldsymbol{z})$, the product of the singular values of $\boldsymbol{J}(\boldsymbol{z})$, and we get the relation $P_{\boldsymbol{z}}(\boldsymbol{z}) = J_{sv}(\boldsymbol{z}) P_{\boldsymbol{x}}(\boldsymbol{x})$. We can further show that $J_{sv}(\boldsymbol{z})$ is also $(1/2\lambda_2\sigma^2)^{N/2}(\prod_{j=1}^N \alpha_j(\boldsymbol{A}(\boldsymbol{x})))^{-1/2}$ under a certain condition that includes the case $\boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{I}_M$ (see Appendix B). Consequently, Eq. (15) holds even for the case $M > N$. In such case, $P_{\boldsymbol{z}}(\boldsymbol{z})$ and $(\prod_{j=1}^N \alpha_j(\boldsymbol{A}(\boldsymbol{x})))^{-1/2} P_{\boldsymbol{x}}(\boldsymbol{x})$, the probability distribution function of $\boldsymbol{x}$ modified by a metric depending scaling, becomes proportional. As a result, when we obtain a parameter ψ attaining $P_{\boldsymbol{z},\psi}(\boldsymbol{z}) \simeq P_{\boldsymbol{z}}(\boldsymbol{z})$ by training, $P_{\boldsymbol{x}}(\boldsymbol{x})$ is proportional to $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ with a metric depending scaling $(\prod_{j=1}^N \alpha_j(\boldsymbol{A}(\boldsymbol{x})))^{1/2}$ as:

$$P_{\boldsymbol{x}}(\boldsymbol{x}) \propto \left(\prod_{j=1}^N \alpha_j(\boldsymbol{A}(\boldsymbol{x}))\right)^{\frac{1}{2}} P_{\boldsymbol{z},\psi}(\boldsymbol{z}). \quad (16)$$

In the case of $\boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{I}_M$ (or more generally $\kappa \boldsymbol{I}_M$ for a constant $\kappa > 0$), $P_{\boldsymbol{x}}(\boldsymbol{x})$ is simply proportional to $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$:

$$P_{\boldsymbol{x}}(\boldsymbol{x}) \propto P_{\boldsymbol{z},\psi}(\boldsymbol{z}). \quad (17)$$

5. Experimental Validations

Here, we show the properties of our method experimentally. In Section 5.1, we examine the isometricity of the map as in Eq. (2) with real data. In Section 5.2, we confirm the proportionality of PDFs as in Eq. (15) with toy data. In Section 5.3, the usefulness is validated in anomaly detection.

5.1. Isometric Embedding

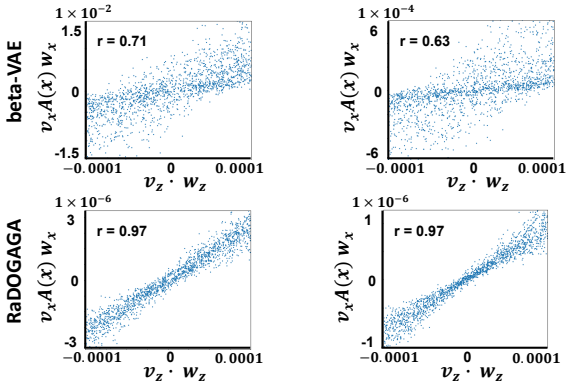
In this section, we confirm that our method can embed data in the latent space isometrically. First, a randomly picked data point $\boldsymbol{x}$ is mapped to $\boldsymbol{z}(= f_\theta(\boldsymbol{x}))$. Then, let $\boldsymbol{v}_z$ be a small tangent vector in the latent space. The corresponding tangent vector in the data space $\boldsymbol{v}_x$ is approximated by $g(\boldsymbol{z}+\boldsymbol{v}_z)-g(\boldsymbol{z})$. Given randomly generated two different tangent vectors $\boldsymbol{v}_z$ and $\boldsymbol{w}_z$, $\boldsymbol{v}_z \cdot \boldsymbol{w}_z$ is compared with ${\boldsymbol{v}_x}^\top \boldsymbol{A}(\boldsymbol{x})\boldsymbol{w}_x$. We use the CelebA dataset (Liu et al., 2015)[1] that consists of 202,599 celebrity images. Images are center-cropped with a size of 64 x 64.

5.1.1. CONFIGURATION

In this experiment, factorized distributions (Ballé et al., 2018) are used to estimate $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ [2]. The encoder part is constructed with four convolution (CNN) layers and two fully connected (FC) layers. For CNN layers, the kernel size is 9×9 for the first one and 5×5 for the rest. The dimension is 64, stride size is 2, and activation function is the generalized divisive normalization (GDN) (Ballé et al., 2016), which is suitable for image compression, for all layers. The dimensions of FC layers are 8192 and 256. For the first one, $softplus$ function is attached. The decoder part is the inverse form of the encoder. For comparison, we evaluate beta-VAE with the same form of autoencoder with 256-dimensional $\boldsymbol{z}$. In this experiment, we test two different metrics; MSE, where $\boldsymbol{A}(\boldsymbol{x}) = \frac{1}{M}\boldsymbol{I}_M$, and $1-SSIM$, where $\boldsymbol{A}(\boldsymbol{x}) = \left(\frac{1}{2{\mu_x}^2}\boldsymbol{W}_m + \frac{1}{2{\sigma_x}^2}\boldsymbol{W}_v\right)$. $\boldsymbol{W}_m \in \mathbb{R}^{M\times M}$ is a matrix such that all elements are $\frac{1}{M^2}$ and $\boldsymbol{W}_v = \frac{1}{M}\boldsymbol{I}_M - \boldsymbol{W}_m$. Note that, in practice, $1-SSIM$ for an image is calculated with a small window. In this experiment the window size is 11×11, and this local calculation is performed for the entire image with the stride size of 1. The cost is the average of local values. For the second term in Eq. (7), $h(d)$ is $\log(d)$ and $\boldsymbol{A}(\boldsymbol{x}) = \frac{1}{M}\boldsymbol{I}_M$. For beta-VAE, we set β^{-1} as 1×10^5 and 1×10^4 regarding to the training with MSE and $1-SSIM$ respectively. For RaDOGAGA, (λ_1, λ_2) is (0.1, 0.1) and (0.2, 0.1). Optimization is done by Adam optimizer (Kingma & Ba, 2014) with learning rate 1×10^{-4}. All models are trained so that the $1-SSIM$ between the input and reconstructed images is approximately 0.05.

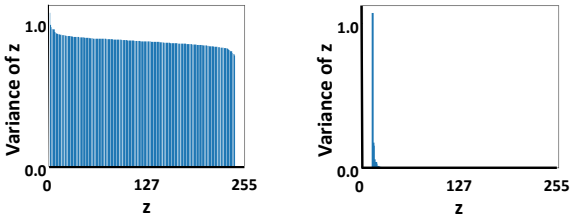
5.1.2. RESULTS

Figure 4 depicts $\boldsymbol{v}_z \cdot \boldsymbol{w}_z$ (horizontal axis) and ${\boldsymbol{v}_x}^\top \boldsymbol{A}(\boldsymbol{x})\boldsymbol{w}_x$ (vertical axis). The top row is the result of beta-VAE and the bottom row shows that of our method. In our method,



(a) MSE (b) $1-SSIM$

Figure 4. Plot of $\boldsymbol{v}_z \cdot \boldsymbol{w}_z$ (horizontal axis) and ${\boldsymbol{v}_x}^\top \boldsymbol{A}(\boldsymbol{x})\boldsymbol{w}_x$ (vertical axis). In beta-VAE (top row), the correlation is weak whereas in our method (bottom row) we can observe proportionality.



(a) beta-VAE (b) RaDOGAGA

Figure 5. Variance of $\boldsymbol{z}$. In beta-VAE, variances of all dimensions are trained to be 1. In RaDOGAGA, the energy is concentrated in few dimensions.

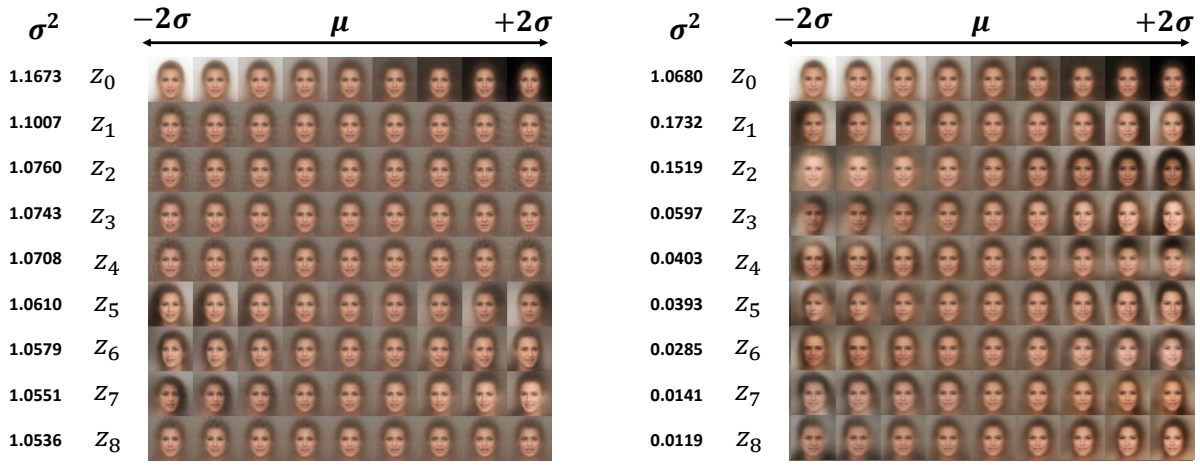
$\boldsymbol{v}_z \cdot \boldsymbol{w}_z$ and ${\boldsymbol{v}_x}^\top \boldsymbol{A}(\boldsymbol{x})\boldsymbol{w}_x$ are almost proportional regardless of the metric function. The correlation coefficients r reach 0.97, whereas that of beta-VAE are around 0.7. It can be seen that our method enables isometric embedding to a Euclidean space even with this large scale real dataset. For interested readers, we provide the experimental results with the MNIST dataset in Appendix F.

5.1.3. CONSISTENCY TO NASH EMBEDDING THEOREM

As explained in Introduction, the Nash embedding theorem and manifold hypothesis are behind our exploration of the isometric embedding of input data. Here, the question is whether the trained model satisfied the condition that $\dim \mathcal{M}_{\boldsymbol{x}} < N$. With RaDOGAGA, we can confirm it by observing the variance of each latent variable. Because the Jacobian matrix forms an orthonormal system, RaDOGAGA can work like principal component analysis (PCA) and evaluates the importance of each latent variable. The theoretical proof for this property is described in Appendix E. Figure 5 shows the variance of each dimension of the model trained with MSE. The variance concentrates on the few dimensions. This means that $\mathbb{R}^N$ is large enough to represent the data. Figure 6 shows the decoder outputs when

[1] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

[2] Implementation is done with a library for TensorFlow provided at https://github.com/tensorflow/compression with default parameters.



(a) beta-VAE (b) RaDOGAGA

Figure 6. Latent space traversal of variables with top-9 variance. In beta-VAE, some latent variables do not influence the visual so much even though they have almost the same variance. In RaDOGAGA, all latent variables with large variance have important information for image.

each component z_i is traversed from -2σ to 2σ, fixing the rest of $\boldsymbol{z}$ as the mean. Note the index i is arranged in a descending order of σ^2. Here, σ^2 and μ for the i-th dimension of $\boldsymbol{z}(= f_\theta(\boldsymbol{x}))$ are $Var[z_i]$ and $E[z_i]$ respectively with all data samples. From the top, each row corresponds to z_0, z_1, z_2 ..., and the center column is mean. In Fig. 6b, the image changes visually in any dimension of $\boldsymbol{z}$, whereas in Fig. 6a, depending on the dimension i, there are cases where no significant changes can be seen (such as z_1, z_2, z_3, and so on). In summary, we can qualitatively observe that $Var[z_i]$ corresponds to the eigenvalue of PCA; that is, a latent variable with a larger σ have bigger impact on image.

These results suggest that the important information to express data are concentrated in the lower few dimensions and the dimension number of 256 is large enough to satisfy $\dim \mathcal{M}_{\boldsymbol{x}} < N$. To confirm the sufficiency of the dimension is difficult in beta-VAE because σ^2 should be 1 for all dimensions because it is trained to fit to the prior. However, some dimensions have a large impact on the image, meaning that σ does not work as the measure of importance.

We believe that this PCA-like trait is very useful for the interpretation of latent variables. For instance, if the metric function were designed so as to reflect semantics, important variables for a semantics are easily found. Furthermore, we argue that this is a promising way to capture the minimal feature to express data, which is one of the goals of machine learning.

5.2. PDF Estimation with Toy Data

In this section, we describe our experiment using toy data to demonstrate whether the probability density function of the input data $P_{\boldsymbol{x}}(\boldsymbol{x})$ and that of the latent variable estimated in the latent space $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ are proportional to each other as in theory. First, we sample data points $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_n, ..., \boldsymbol{s}_{10,000}) \in \mathbb{R}^{3\times 10,000}$ from three-dimensional GMM consists of three mixture-components with mixture weight $\boldsymbol{\pi} = (1/3, 1/3, 1/3)$, mean $\boldsymbol{\mu}_1 = (0,0,0)$, $\boldsymbol{\mu}_2 = (15,0,0)$, $\boldsymbol{\mu}_3 = (15,15,15)$, and covariance $\boldsymbol{\Sigma}_k = \mathrm{diag}(1,2,3)$. k is the index for the mixture-component. Then, we scatter $\boldsymbol{s}$ with uniform random noise $\boldsymbol{u} \in \mathbb{R}^{3\times 16}$, $u_{dm} \sim U_d\left(-\frac{1}{2}, \frac{1}{2}\right)$, where d and m are index for dimension of sampled data and scattered data. The $\boldsymbol{u}_d$s are uncorrelated with each other. We produce 16-dimensional input data as $\boldsymbol{x}_n = \sum_{d=1}^{3} \boldsymbol{u}_d s_{nd}$ with a sample number of 10,000 in the end. The appearance probability of input data $P_{\boldsymbol{x}}(\boldsymbol{x})$ is equals to a generation probability of $\boldsymbol{s}$.

5.2.1. CONFIGURATION

In the experiment, we estimate $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ using GMM with parameter ψ as in DAGMM (Zong et al., 2018). Instead of the EM algorithm , GMM parameters are learned using Estimation Network (EN), which consists of a multi-layer neural network. When the GMM consists of N-dimensional Gaussian distribution $\mathcal{N}_N(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with K mixture-components, and L is the size of batch samples, EN outputs the mixture-components membership prediction as a K-dimensional vector $\widehat{\boldsymbol{\gamma}}$ as follows:

$$\boldsymbol{p} = EN(\boldsymbol{z}; \psi), \widehat{\boldsymbol{\gamma}} = softmax(\boldsymbol{p}). \tag{18}$$

Then, k-th mixture weight $\widehat{\pi}_k$, mean $\widehat{\boldsymbol{\mu}}_k$, covariance $\widehat{\boldsymbol{\Sigma}}_k$, and entropy R of $\boldsymbol{z}$ are further calculated as follows:
$\widehat{\pi}_k = \sum_{l=1}^{L} \widehat{\gamma}_{lk}/L, \quad \widehat{\boldsymbol{\mu}}_k = \sum_{l=1}^{L} \widehat{\gamma}_{lk}\boldsymbol{z}_l / \sum_{l=1}^{L} \widehat{\gamma}_{lk},$
$\widehat{\boldsymbol{\Sigma}}_k = \sum_{l=1}^{L} \widehat{\gamma}_{lk}(\boldsymbol{z}_l - \widehat{\boldsymbol{\mu}}_k)(\boldsymbol{z}_l - \widehat{\boldsymbol{\mu}}_k)^\top / \sum_{l=1}^{L} \widehat{\gamma}_{lk},$

$$R = -\log\left(\sum_{k=1}^{K} \mathcal{N}_N(\boldsymbol{z}; \widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k)\right).$$

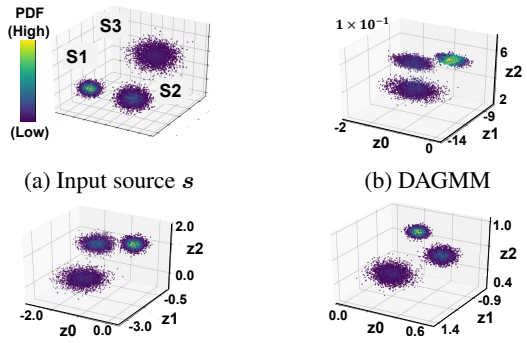
The overall network is trained by Eqs. (7) and (9). In this experiment, we set $D(\cdot\ ,\ \cdot)$ as the square of the Euclidean distance because the input data is generated obeying the PDF in the Euclidean space. We test two types of $h(\cdot)$, $h(d) = d$ and $h(d) = \log(d)$, and denote models trained with these $h(\cdot)$ as RaDOGAGA(d) and RaDOGAGA(log(d)) respectively. We used DAGMM as a baseline method. DAGMM also consists of an encoder, decoder, and EN. In DAGMM, to avoid falling into the trivial solution that the entropy is minimized when the diagonal component of the covariance matrix is 0, the inverse of the diagonal component $P(\widehat{\boldsymbol{\Sigma}}) = \sum_{k=1}^{K}\sum_{i=1}^{N} \widehat{\boldsymbol{\Sigma}}_{ki}^{-1}$ is added to the cost:

$$L = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2 + \lambda_1(-\log(P_{\boldsymbol{z},\psi}(\boldsymbol{z}))) + \lambda_2 P(\widehat{\boldsymbol{\Sigma}}). \quad (19)$$

The only differences between our model and DAGMM is that the regulation term $P(\widehat{\boldsymbol{\Sigma}})$ is replaced by $D(\hat{\boldsymbol{x}}, \breve{\boldsymbol{x}})$. The model complexity such as the number of parameters is the same. For both RaDOGAGA and DAGMM, the autoencoder part is constructed with FC layers with sizes of 64, 32, 16, 3, 16, 32, and 64. For all FC layers except for the last of the encoder and the decoder, we use $tanh$ as the activation function. The EN part is also constructed with FC layers with sizes of 10 and 3. For the first layer, we use $tanh$ as the activation function and dropout (ratio=0.5). For the last one, $softmax$ is used. (λ_1, λ_2) is set as $(1 \times 10^{-4}, 1 \times 10^{-9})$, $(1 \times 10^{6}, 1 \times 10^{3})$ and $(1 \times 10^{3}, 1 \times 10^{3})$ for DAGMM, RaDOGAGA(d) and RaDOGAGA(log(d)) respectively. We optimize all models by Adam optimizer with a learning rate of 1×10^{-4}. We set σ^2 as $1/12$.

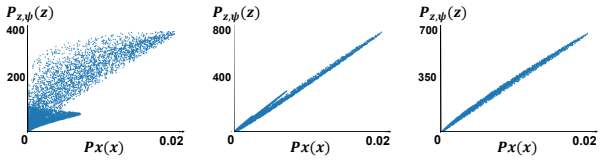
5.2.2. Results

Figure 7 displays the distribution of the input data source $\boldsymbol{s}$ and latent variable $\boldsymbol{z}$. Although both methods can capture that $\boldsymbol{s}$ is generated from three mixture-components, there is a difference in the PDFs. Since the data is generated from GMM, the value of the PDF gets higher as the sample gets closer to the centers of clusters. However, in DAGMM, this tendency looks distorted. This difference is further demonstrated in Fig. 8, which shows a plot of $P_{\boldsymbol{x}}(\boldsymbol{x})$ (horizontal axis) against $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ (vertical axis). In our method, $P_{\boldsymbol{x}}(\boldsymbol{x})$ and $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ are almost proportional to each other as in the theory, but we cannot observe such a proportionality in DAGMM. This difference is also quantitatively obvious. That is, correlation coefficients between $P_{\boldsymbol{x}}(\boldsymbol{x})$ and $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ are 0.882 (DAGMM), 0.997 (RaDOGAGA(d)), and 0.998 (RaDOGAGA(log(d))). We can also observe that, in RaDOGAGA(d), there is a slight distortion in its proportionality in the area of $P_{\boldsymbol{x}}(\boldsymbol{x}) < 0.01$. When $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ is sufficiently fitted, $h(d) = \log(d)$ makes $P_{\boldsymbol{x}}(\boldsymbol{x})$ and $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ be proportional more rigidly. More details are given in Appendix C.



(a) Input source $\boldsymbol{s}$ (b) DAGMM (c) RaDOGAGA(d) (d) RaDOGAGA(log(d))

Figure 7. Plot of the source of input data $\boldsymbol{s}$ and latent variables $\boldsymbol{z}$. The color bar located left of (a) corresponds to the normalized PDF. Both DAGMM and RaDOGAGA capture three mixture-components, but the PDF in DAGMM looks different from the input data source. Points with high PDF do not concentrate on the center of the cluster especially in the upper right one.



(a) DAGMM (b) Ours (d) (c) Ours (log(d))

Figure 8. Plot of $P_{\boldsymbol{x}}(\boldsymbol{x})$ vs $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$. In RaDOGAGA, $P_{\boldsymbol{x}}(\boldsymbol{x})$ and $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ are proportional while we cannot see that in DAGMM.

5.3. Anomaly Detection Using Real Data

We here examine whether the clear relationship between $P_{\boldsymbol{x}}(\boldsymbol{x})$ and $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ is useful in anomaly detection in which PDF estimation is the key issue. We use four public datasets[‡]: KDDCUP99, Thyroid, Arrhythmia, and KDDCUP-Rev. The (instance number, dimension, anomaly ratio(%)) of each dataset is (494021, 121, 20), (3772, 6, 2.5), (452, 274, 15), and (121597, 121, 20). The details of the datasets are given in Appendix G.

5.3.1. Experimental Setup

For a fair comparison with previous works, we follow the setting in Zong et al. (2018). Randomly extracted 50% of the data were assigned to the training and the rest to the testing. During the training, only normal data were used. During the test, the entropy R for each sample was calculated as the anomaly score, and if the anomaly score is higher than a threshold, it is detected as an anomaly. The threshold is determined by the ratio of the anomaly data in each data set. For example, in KDDCup99, data with R in the top 20 % is

[‡]Datasets can be downloaded at https://kdd.ics.uci.edu/ and http://odds.cs.stonybrook.edu.

Table 1. Average and standard deviations (in brackets) of Precision, Recall and F1

Dataset	Methods	Precision	Recall	F1
KDDCup	ALAD*	0.9427 (0.0018)	0.9577 (0.0018)	0.9501 (0.0018)
	INRF*	0.9452 (0.0105)	0.9600 (0.0113)	0.9525 (0.0108)
	GMVAE*	0.952	0.9141	0.9326
	DAGMM	0.9427 (0.0052)	0.9575 (0.0053)	0.9500 (0.0052)
	RaDOGAGA(d)	0.9550 (0.0037)	0.9700 (0.0038)	0.9624 (0.0038)
	RaDOGAGA(log(d))	**0.9563 (0.0042)**	**0.9714 (0.0042)**	**0.9638 (0.0042)**
Thyroid	GMVAE*	**0.7105**	0.5745	0.6353
	DAGMM	0.4656 (0.0481)	0.4859 (0.0502)	0.4755 (0.0491)
	RaDOGAGA(d)	0.6313 (0.0476)	0.6587 (0.0496)	0.6447 (0.0486)
	RaDOGAGA(log(d))	0.6562 (0.0572)	**0.6848 (0.0597)**	**0.6702 (0.0585)**
Arrythmia	ALAD*	0.5000 (0.0208)	0.5313 (0.0221)	0.5152 (0.0214)
	GMVAE*	0.4375	0.4242	0.4308
	DAGMM	0.4985 (0.0389)	0.5136 (0.0401)	0.5060 (0.0395)
	RaDOGAGA(d)	**0.5353 (0.0461)**	**0.5515 (0.0475)**	**0.5433 (0.0468)**
	RaDOGAGA(log(d))	0.5294 (0.0405)	0.5455 (0.0418)	0.5373 (0.0411)
KDDCup-rev	DAGMM	0.9778 (0.0018)	0.9779 (0.0017)	0.9779 (0.0018)
	RaDOGAGA(d)	0.9768 (0.0033)	0.9827 (0.0012)	0.9797 (0.0015)
	RaDOGAGA(log(d))	**0.9864 (0.0009)**	**0.9865 (0.0009)**	**0.9865 (0.0009)**

*Scores are cited from Zenati et al. (2018) (ALAD), Song & Ou (2018) (INRF), and Liao et al. (2018) (GMVAE).

detected as an anomaly. As metrics, precision, recall, and F1 score are calculated. We run experiments 20 times for each dataset split by 20 different random seeds.

5.3.2. BASELINE METHODS

As in the previous section, DAGMM is taken as a baseline. We also compare the scores of our method with the ones reported in previous works conducting the same experiments (Zenati et al., 2018; Song & Ou, 2018; Liao et al., 2018).

5.3.3. CONFIGURATION

As in Zong et al. (2018), in addition to the output from the encoder, $\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_2}{\|\boldsymbol{x}\|_2}$ and $\frac{\boldsymbol{x}\cdot\boldsymbol{x}'}{\|\boldsymbol{x}\|_2\|\boldsymbol{x}'\|_2}$ are concatenated to $\boldsymbol{z}$ and sent to EN. Note that $\boldsymbol{z}$ is sent to the decoder before concatenation. Other configuration a except for the hyperparameter is same as in the previous experiment. The hyperparameter for each dataset is described in Appendix G. The input data are max-min normalized.

5.3.4. RESULTS

Table 1 reports the average scores and standard deviations (in brackets). Compared to DAGMM, RaDOGAGA has a better performance regardless of types of $h(\cdot)$. Note that, our method does not increase model complexity at all. Simply introducing the RDO mechanism to the autoencoder is effective for anomaly detection. Moreover, RaDOGAGA achieves the highest performance compared to other methods. RaDOGAGA(log(d)) is superior to RaDOGAGA(d) except for Arrhythmia. This result suggests that a much rigid orthonormality can likely bring better performance.

6. Conclusion

In this paper, we propose RaDOGAGA which embeds data in a low-dimensional Euclidean space isometrically. With RaDOGAGA, the relation of latent variables and data is quantitatively tractable. For instance, $P_{\boldsymbol{z},\psi}(\boldsymbol{z})$ obtained by the proposed method is related to $P_{\boldsymbol{x}}(\boldsymbol{x})$ in a clear form, e.g., they are proportional when $\boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{I}$. Furthermore, thanks to these properties, we achieve a state-of-the-art performance in anomaly detection.

Although we focused on the PDF estimation as a practical task in this paper, the properties of RaDOGAGA will benefit a variety of applications. For instance, data interpolation will be easier because a straight line in the latent space is geodesic in the data space. It also may help the unsupervised or semi-supervised learning since the distance of $\boldsymbol{z}$ reliably reflects the distance of $\boldsymbol{x}$. Furthermore, our method will promote disentanglement because, thanks to the orthonormality, PCA-like analysis is possible.

To capture the essential features of data, it is important to fairly evaluate the significance of latent variables. Because isometric embedding ensures this fairness, we believe that RaDOGAGA will bring a *Breakthru* for generative analysis. As a future work, we explore the usefulness of this method in various tasks mentioned above.

Acknowledgement

We express our gratitude to Naoki Hamada, Ramya Srinivasan, Kentaro Takemoto, Moyuru Yamada, Tomoya Iwakura, and Hiyori Yoshikawa for constructive discussion.

References

Alemi, A., Poole, B., Fischer, I., Dillon, J., Saurous, R. A., and Murphy, K. Fixing a broken ELBO. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 159–168, 2018.

Ballé, J., Laparra, V., and Simoncelli, E. Density modeling of images using a generalized normalization transformation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. Variational image compression with a scale hyperprior. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Bengio, Y., Courville, C., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (8):798–1828, 2013.

Berger, T. (ed.). *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice Hall, 1971.

Bernstein, M., De Silva, V., Langford, J. C., and Tenenbaum, J. B. Graph approximations to geodesics on embedded manifolds. Technical report, Stanford University, 2000.

Brekelmans, R., Moyer, D., Galstyan, A., and Ver Steeg, G. Exact rate-distortion in autoencoders via echo noise. In *Advances in Neural Information Processing Systems*, pp. 3889–3900, 2019.

Chalapathy, R. and Chawla, S. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

Chen, N., , Klushyn, A., Kurle, Jiang, X., Bayer, J., and van der Smagt, P. Metrics for deep generative models. In *Proceedings of the International Conference on Artifcial Intelligence and Statistics (AISTATS)*, pp. 1540–1550, 2018.

Dinh, L., Krueger, D., and Bengio, Y. NICE: Non-linear independent components estimation. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshop*, 2015.

Dua, D. and Graff, C. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2019.

Geng, C., Wang, J., Chen, L., Bao, W., Chu, C., and Gao, Z. Uniform interpolation constrained geodesic learning on data manifold. *arXiv preprint arXiv:2002.04829*, 2020.

Goyal, V. K. Theoretical foundations of transform coding. *IEEE Signal Processing Magazine*, 18(5):9–21, 2001.

Han, Q. and Hong, J.-X. *Isometric Embedding of Riemannian Manifolds in Euclidean Spaces*. American Mathematical Society, 2006.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. β-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Horn, R. A. and Johnson, C. R. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2013.

Johnson, M., Duvenaud, D., Wiltschko, A. B., Datta, S. R., and Adams, R. P. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Liao, W., Guo, Y., Chen, X., and Li, P. A unified unsupervised Gaussian mixture variational autoencoder for high dimensional outlier detection. In *Proceedings of the IEEE International Conference on Big Data*, pp. 1208–1217, 2018.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015.

McQueen, J., Meila, M., and Joncas, D. Nearly isometric embedding by relaxation. In *Advances in Neural Information Processing Systems*, pp. 2631–2639, 2016.

Pai, G., Talmon, R., Alex, B., and Kimmel, R. DIMAL: Deep isometric manifold learning using sparse geodesic sampling. In *Proceedings of the IEEE Winter Conference*

on Applications of Computer Vision (WACV), pp. 819–828, 2019.

Rao, K. R. and Yip, P. (eds.). *The Transform and Data Compression Handbook*. CRC Press, Inc., 2000.

Rolínek, M., Zietlow, D., and Martius, G. Variational autoencoders pursue PCA directions (by accident). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12406–12415, 2019.

Shao, H., Kumar, A., and Fletcher, P. T. The Riemannian geometry of deep generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 315–323, 2018.

Song, Y. and Ou, Z. Learning neural random fields with inclusive auxiliary generators. *arXiv preprint arXiv:1806.00271*, 2018.

Strang, G. *Linear Algebra and its Applications*. Cengage Learning, 4th edition, 2006.

Sullivan, G. J. and Wiegand, T. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine*, 15(6):74–90, 1998.

Teoh, H. S. Formula for vector rotation in arbitrary planes in $\mathbb{R}^n$. http://eusebeia.dyndns.org/4d/genrot.pdf, 2005.

Tschannen, M., Bachem, O., and Lucic, M. Recent advances in autoencoder-based representation learning. In *Proceedings of the Third Workshop on Bayesian Deep Learning (NeurIPS 2018 Workshop)*, 2018.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Zenati, H., Romain, M., Foo, C.-S., Lecouat, B., and Chandrasekhar, V. Adversarially learned anomaly detection. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pp. 727–736, 2018.

Zhou, J., Wen, S., Nakagawa, A., Kazui, K., and Tan, Z. Multi-scale and context-adaptive entropy model for image compression. In *Proceedings of the Workshop and Challenge on Learned Image Compression (CVPR 2019 Workshop)*, pp. 4321–4324, 2019.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.