A. Derivations and proofs in Section 4.3

A.1. Proof of Lemma 1: Approximated expansion of the reconstruction loss

The approximated expansion of the reconstruction loss is mainly the same as Rolínek et al. (2019) except we consider a metric tensor $\boldsymbol{G}_{\boldsymbol{x}}$ which is a positive definite Hermitian matrix.

$\delta\breve{\boldsymbol{x}}$ and $\delta\hat{\boldsymbol{x}}$ denote $\breve{\boldsymbol{x}}-\boldsymbol{x}$ and $\hat{\boldsymbol{x}}-\breve{\boldsymbol{x}}$, respectively. Let $\delta z_j \sim \mathcal{N}(\delta z_j; 0, \sigma_{j(\boldsymbol{x})})$ be an added noise in the reparameterization trick where $z_j = \mu_{j(\boldsymbol{x})} + \delta z_j$. Then, $\delta\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}} - \breve{\boldsymbol{x}}$ is approximated as:

$$\delta\hat{\boldsymbol{x}} \simeq \sum_{j=1}^{n} \delta z_j \, \boldsymbol{x}_{\mu_j}. \tag{26}$$

Next, the reconstruction loss $D(\boldsymbol{x}, \hat{\boldsymbol{x}})$ can be approximated as follows.

$$\begin{aligned} D(\boldsymbol{x}, \hat{\boldsymbol{x}}) &= D(\boldsymbol{x}, \boldsymbol{x} + (\delta\breve{\boldsymbol{x}} + \delta\hat{\boldsymbol{x}})) \\ &\simeq {}^t(\delta\breve{\boldsymbol{x}} + \delta\hat{\boldsymbol{x}})\boldsymbol{G}_{\boldsymbol{x}}(\delta\breve{\boldsymbol{x}} + \delta\hat{\boldsymbol{x}}) \\ &= {}^t\delta\breve{\boldsymbol{x}}\, \boldsymbol{G}_{\boldsymbol{x}}\delta\breve{\boldsymbol{x}} + {}^t\delta\hat{\boldsymbol{x}}\, \boldsymbol{G}_{\boldsymbol{x}}\delta\hat{\boldsymbol{x}} + 2\, {}^t\delta\hat{\boldsymbol{x}}\, \boldsymbol{G}_{\boldsymbol{x}}\delta\breve{\boldsymbol{x}} \\ &\simeq D(\boldsymbol{x}, \breve{\boldsymbol{x}}) + D(\breve{\boldsymbol{x}}, \hat{\boldsymbol{x}}) + \sum_{j=1}^{n} 2\delta z_j \, {}^t\boldsymbol{x}_{\mu_j} \boldsymbol{G}_x \delta\breve{\boldsymbol{x}} \end{aligned} \tag{27}$$

Then, we evaluate the average of $D(\boldsymbol{x}, \hat{\boldsymbol{x}})$ over $\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x})$, i.e., $\delta z_j \sim \mathcal{N}(\delta z_j; 0, \sigma_{j(\boldsymbol{x})})$ for all j. Note that $E[\delta z_j \delta z_k] = {\sigma_{j(\boldsymbol{x})}}^2 \delta_{jk}$ where δ_{jk} is the Kronecker delta. First, the average of $D(\boldsymbol{x}, \breve{\boldsymbol{x}})$ in the last line of Eq. 27 is still $D(\boldsymbol{x}, \breve{\boldsymbol{x}})$ since this term does not depend on δz_j. Second, the average of $D(\breve{\boldsymbol{x}}, \hat{\boldsymbol{x}})$ in the last line of Eq. 27 is approximated as:

$$\begin{aligned} E_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[D(\breve{\boldsymbol{x}}, \hat{\boldsymbol{x}})\right] &\simeq E_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[{}^t\delta\hat{\boldsymbol{x}}\, \boldsymbol{G}_{\boldsymbol{x}}\delta\hat{\boldsymbol{x}}\right] \\ &\simeq E_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\Big(\sum_{j=1}^{n} \delta z_j \, {}^t\boldsymbol{x}_{\mu_j}\Big)\boldsymbol{G}_{\boldsymbol{x}}\Big(\sum_{k=1}^{n} \delta z_k \, \boldsymbol{x}_{\mu_k}\Big)\right] \\ &= \sum_{j=1}^{n}\sum_{k=1}^{n} E_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\delta z_j \delta z_k] \, {}^t\boldsymbol{x}_{\mu_j} \boldsymbol{G}_{\boldsymbol{x}} \boldsymbol{x}_{\mu_k} \\ &= \sum_{j=1}^{n} {\sigma_{j(\boldsymbol{x})}}^2 \, {}^t\boldsymbol{x}_{\mu_j} \boldsymbol{G}_x \boldsymbol{x}_{\mu_j}. \end{aligned} \tag{28}$$

Third, the average of the third term in the last line of Eq. 27, i.e., $\sum_{j=1}^{n} 2\delta z_j \, {}^t\boldsymbol{x}_{\mu_j} \boldsymbol{G}_x \delta\breve{\boldsymbol{x}}$, is 0 since the average of δz_j over $\mathcal{N}(\delta z_j; 0, \sigma_{j(\boldsymbol{x})})$ is 0.

As a result, the average of $D(\boldsymbol{x}, \hat{\boldsymbol{x}})$ over $\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x})$ can be approximated as:

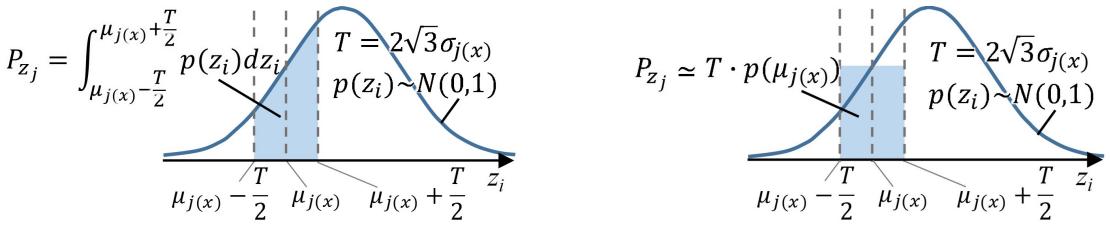
$$E_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[D(\boldsymbol{x}, \hat{\boldsymbol{x}})\right] \simeq D(\boldsymbol{x}, \breve{\boldsymbol{x}}) + \sum_{j=1}^{n} {\sigma_{j(\boldsymbol{x})}}^2 \, {}^t\boldsymbol{x}_{\mu_j} \boldsymbol{G}_x \boldsymbol{x}_{\mu_j}. \tag{29}$$

A.2. Proof of Lemma 2: More precise derivation of KL divergence approximation.

This appendix explains more precise derivation of KL divergence approximation. First, we show the approximation for the Gaussian prior. We also show at the end of this appendix that our approximation also holds for arbitrary prior.

In the case of Gaussian prior, we show that KL divergence can be interpreted as an amount of information in the transform coding (Goyal, 2001) allowing the distortion ${\sigma_{j(\boldsymbol{x})}}^2$. In the transform coding, input data is transformed by an orthonormal transform. Then, the transformed data is quantized, and an entropy code is assigned to the quantized symbol, such that the length of the entropy code is equivalent to the logarithm of the estimated symbol probability. Here, we assume ${\sigma_{j(\boldsymbol{x})}}^2 \ll 1$ will be observed in meaningful dimensions as shown later.

It is generally intractable to derive the rate and distortion of individual symbols in the ideal information coding. Thus, we first discuss the case of uniform quantization. Let P_{z_j} and R_{z_j} be the probability and amount of information in the



(a) Probability P_{z_j} (b) Approximation of P_{z_j}

Figure 8. Probability for a symbol with mean μ and noise σ^2

uniform quantization coding of $z_j \sim \mathcal{N}(z_j; 0, 1)$. Here, $\mu_{j(\boldsymbol{x})}$ and ${\sigma_{j(\boldsymbol{x})}}^2$ are regarded as a quantized value and a coding noise after the uniform quantization, respectively. Since we assume ${\sigma_{j(\boldsymbol{x})}}^2 \ll 1$, $\mu_{j(\boldsymbol{x})} \sim \mathcal{N}(\mu_{j(\boldsymbol{x})}; 0, 1)$ will also hold. Let T be a quantization step size. The coding noise after quantization is $T^2/12$ for the quantization step size T, as explained in Appendix G.1. Thus, T is derived as $T = 2\sqrt{3}\sigma_{j(\boldsymbol{x})}$ from ${\sigma_{j(\boldsymbol{x})}}^2 = T^2/12$. We also assume ${\sigma_{j(\boldsymbol{x})}}^2 \ll 1$. As shown in Fig.8a, P_{z_j} is denoted by $\int_{\mu_{j(\boldsymbol{x})}-T/2}^{\mu_{j(\boldsymbol{x})}+T/2} p(z_j)\mathrm{d}z_j$ where $p(z_j)$ is $\mathcal{N}(z_j; 0, 1)$. Using Simpson's numerical integration method and $e^x = 1 + x + O(x^2)$ expansion, P_{z_j} is approximated as:

$$\begin{aligned} P_{z_j} &\simeq \frac{T}{6}\left(p(\mu_{j(\boldsymbol{x})} - \tfrac{T}{2}) + 4p(\mu_{j(\boldsymbol{x})}) + p(\mu_{j(\boldsymbol{x})} + \tfrac{T}{2})\right) \\ &= \frac{Tp(\mu_{j(\boldsymbol{x})})}{6}\left(4 + e^{\frac{4\mu_{j(\boldsymbol{x})}T - T^2}{8}} + e^{\frac{-4\mu_{j(\boldsymbol{x})}T - T^2}{8}}\right) \\ &\simeq Tp\left(\mu_{j(\boldsymbol{x})}\right)\left(1 - T^2/24\right) \\ &= \sqrt{\frac{6}{\pi}}\sigma_{j(\boldsymbol{x})}\, e^{-({\mu_{j(\boldsymbol{x})}}^2)/2}\left(1 - \frac{{\sigma_{j(\boldsymbol{x})}}^2}{2}\right). \end{aligned} \tag{30}$$

Using $\log(1 + x) = x + O(x^2)$ expansion, $R_{\mu\sigma}$ is derived as:

$$R_{z_j} = -\log P_{z_j} \simeq \frac{1}{2}\left({\mu_{j(\boldsymbol{x})}}^2 + {\sigma_{j(\boldsymbol{x})}}^2 - \log {\sigma_{j(\boldsymbol{x})}}^2 - \log\frac{6}{\pi}\right) = D_{\mathrm{KL}j(\boldsymbol{x})}(\cdot) + \frac{1}{2}\log\frac{\pi e}{6}. \tag{31}$$

When R_{z_j} and $D_{\mathrm{KL}j(\boldsymbol{x})}(\cdot)$ in Eq. 2 are compared, both equations are equivalent except a small constant difference $\frac{1}{2}\log(\pi e/6) \simeq 0.176$ for each dimension. As a result, KL divergence for j-th dimension is equivalent to the rate for the uniform quantization coding, allowing a small constant difference.

To make theoretical analysis easier, we use the simpler approximation as $P_{z_j} = T\, p(\mu_{j(\boldsymbol{x})}) = 2\sqrt{3}\sigma_{j(\boldsymbol{x})}\, p(\mu_{j(\boldsymbol{x})})$ instead of Eq.30, as shown in Fig.8b. Then, R_{z_j} is derived as:

$$R_{z_j} = -\log(2\sqrt{3}\,\sigma_{j(\boldsymbol{x})}\, p(\mu_{j(\boldsymbol{x})})) = \frac{1}{2}\left({\mu_{j(\boldsymbol{x})}}^2 - \log {\sigma_{j(\boldsymbol{x})}}^2 - 1\right) + \frac{1}{2}\log\frac{\pi e}{6}. \tag{32}$$

Here, the first term of the right equation is equivalent to Eq. 11. This equation also means that the approximation of KL divergence in Eq. 11 is equivalent to the rate in the uniform quantization coding with $P_{z_j} = 2\sqrt{3}\sigma_{j(\boldsymbol{x})}\, p(\mu_{j(\boldsymbol{x})})$ approximation, allowing the same small constant difference as in Eq. 31. It is noted that the approximation $P_{z_j} = 2\sqrt{3}\sigma_{j(\boldsymbol{x})}\, p(\mu_{j(\boldsymbol{x})})$ in Figure 8b can be applied to any kinds of prior PDFs because there is no explicit assumption for the prior PDF. This implies that the theoretical discussion after Eq. 11 in the main text will hold in arbitrary prior PDFs.

The meaning of the small constant difference $\frac{1}{2}\log\frac{\pi e}{6}$ in Eqs. 31 and 32 can be explained as follows: Pearlman & Said (2011) show that the difference of the rate between the ideal information coding and uniform quantization is $\frac{1}{2}\log\frac{\pi e}{6}$. This is caused by the entropy difference of the noise distributions. In the ideal case, the noise distribution is known as a Gaussian. In the case the noise variance is σ^2, the entropy of the Gaussian noise is $\frac{1}{2}\log(\sigma^2 2\pi e)$. For the uniform quantization with a uniform noise distribution, the entropy is $\frac{1}{2}\log(\sigma^2 12)$. As a result, the difference is just $\frac{1}{2}\log\frac{\pi e}{6}$. Because the rate estimation in this appendix uses a uniform quantization, the small offset $\frac{1}{2}\log\frac{\pi e}{6}$ can be regarded as a difference between

the ideal information coding and the uniform quantization. As a result, KL divergence in Eq. 2 and Eq. 11 can be regarded as a rate in the ideal informaton coding for the symbol with the mean $\mu_{j(\boldsymbol{x})}$ and variance ${\sigma_{j(\boldsymbol{x})}}^2$.

Here, we validate the assumption that $\sigma_{j(\boldsymbol{x})} \ll 1$ will be observed in meaningful dimensions. From the discussion above, the information R_{z_j} in each dimension can be considered as KL divergence:

$$R_{z_j} = \frac{1}{2}\left({\mu_{j(\boldsymbol{x})}}^2 + {\sigma_{j(\boldsymbol{x})}}^2 - \log {\sigma_{j(\boldsymbol{x})}}^2 - 1\right). \tag{33}$$

For simple analysis, we assume that $\sigma_{j(\boldsymbol{x})}$ is constant in the j-th dimension. We further assume $\mu_{j(\boldsymbol{x})} \sim \mathcal{N}(\mu_{j(\boldsymbol{x})}; 0, 1)$. Then $E[R_{z_j}]$ which shows the information of the j-th dimensional component is derived as:

$$\begin{aligned} E[R_{z_j}] &\simeq E_{\mu_{j(\boldsymbol{x})}\sim\mathcal{N}(\mu_{j(\boldsymbol{x})};0,1)}[R_{z_j}] \\ &= \int \frac{1}{2}\left({\mu_{j(\boldsymbol{x})}}^2 + {\sigma_{j(\boldsymbol{x})}}^2 - \log {\sigma_{j(\boldsymbol{x})}}^2 - 1\right) \mathcal{N}(\mu_{j(\boldsymbol{x})}; 0, 1)\, \mathrm{d}\mu_{j(\boldsymbol{x})} \\ &= \frac{1}{2}\left({\sigma_{j(\boldsymbol{x})}}^2 - \log {\sigma_{j(\boldsymbol{x})}}^2\right). \end{aligned} \tag{34}$$

From this equation, we can estimate an amount of information in each dimension from the posterior variance. From this equation, it is derived that if the amount of information $E[R_{z_j}]$ is more than about 1.20 nat or 1.73 bit, ${\sigma_{j(\boldsymbol{x})}}^2 < 0.1$ holds. In addition, as the information $E[R_{z_j}]$ is increasing, ${\sigma_{j(\boldsymbol{x})}}^2$ becomes exponentially decreasing. As a result, the assumption that ${\sigma_{j(\boldsymbol{x})}}^2 \ll 1$ will be observed in meaningful dimensions is reasonable.

Finally, we show that the approximation of the KL divergence in the second line of Eq. 10 also holds for arbitrary priors. Let $p(\boldsymbol{z})$, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, and $D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}))$ be an arbitrary prior, a posterior $\prod_j \mathcal{N}(\mu_{j(\boldsymbol{x})}, \sigma_{j(\boldsymbol{x})})$, and KL divergence, respectively. First, the shape of $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ becomes close to a delta function $\delta(\boldsymbol{z} - \boldsymbol{\mu}_{(\boldsymbol{x})})$ when each $\sigma_{j(\boldsymbol{x})}$ is small. Thus $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ will act like a delta function $\delta(\boldsymbol{z} - \boldsymbol{\mu}_{(\boldsymbol{x})})$. Next the differential entropy of $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, i.e., $-\int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log q_\phi(\boldsymbol{z}|\boldsymbol{x}) \mathrm{d}\boldsymbol{z}$ is derived as $\sum_j^n \log \sigma_{j(\boldsymbol{x})}\sqrt{2\pi e}$. Using these equations, KL divergence for an arbitrary prior can be approximated by the second line of Eq. 10 as follows:

$$\begin{aligned} D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})) &= -\int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} + \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log q_\phi(\boldsymbol{z}|\boldsymbol{x}) \mathrm{d}\boldsymbol{z} \\ &\simeq -\int \delta(\boldsymbol{z} - \boldsymbol{\mu}_{(\boldsymbol{x})}) \log p(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} + \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log q_\phi(\boldsymbol{z}|\boldsymbol{x}) \mathrm{d}\boldsymbol{z} \\ &= -\log p(\boldsymbol{\mu}_{(\boldsymbol{x})}) - \sum_j^n \log \sigma_j \sqrt{2\pi e} \\ &= -\log\left(p(\boldsymbol{\mu}_{(\boldsymbol{x})}) \prod_{j=1}^n \sigma_{j(\boldsymbol{x})}\right) - \frac{n \log 2\pi e}{2} \end{aligned} \tag{35}$$

In the derivation of the third line in Eq. 10, we assume that $q_\phi(\boldsymbol{\mu}_{(\boldsymbol{x})})$ is close to $p(\boldsymbol{\mu}_{(\boldsymbol{x})})$ where $p(\cdot)$ is the prior distribution of $\boldsymbol{z}$. The reason of this assumption is as follows: ELBO can be also derived as $\log p(\boldsymbol{x}) - D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\boldsymbol{z}|\boldsymbol{x}))$ (Bishop, 2006). When ELBO is maximized at each $\boldsymbol{x}$, $q_\phi(\boldsymbol{z}|\boldsymbol{x}) \simeq p_\theta(\boldsymbol{z}|\boldsymbol{x})$ will hold to minimize KL divergence where $\log p(\boldsymbol{x})$ is a constant. Finally, we have $q_\phi(\boldsymbol{z}) \simeq p(\boldsymbol{z})$ by the marginalization of $\boldsymbol{x}$. Appendix A.3 also validates this assumption in the simple 1-dimensional VAE case where $q_\phi(\mu_{(x)}) = p(\mu_{(x)}) = \mathcal{N}(\mu_{(x)}; 0, 1)$ holds. Thus, we can derive the approximation $p(\boldsymbol{\mu}_{(\boldsymbol{x})}) \simeq q_\phi(\boldsymbol{\mu}_{(\boldsymbol{x})}) = p(\boldsymbol{x})\ |\det(\partial \boldsymbol{x}/\partial \boldsymbol{\mu}_{(\boldsymbol{x})})|$ in Eq. 10.

A.3. Proof of Lemma 3: Estimation of the coding loss and transform loss in 1-dimensional linear VAE

This appendix estimates the coding loss and transform loss in 1-dimensional linear β-VAE for the Gaussian data, and also shows that the result is consistent with the Wiener filter (Wiener, 1964). Let x be a one dimensional input data with the normal distribution:

$$x \in \mathbb{R}, \quad x \sim \mathcal{N}(x; 0, {\sigma_x}^2). \tag{36}$$

First, a simple VAE model in this analysis is explained. z denotes a one dimensional latent variable. Let the prior distribution $p(z)$ be $\mathcal{N}(z; 0, 1)$. Next, two linear parametric encoder and decoder are provided with constant parameters a, b, and σ_z to

optimize:

$$\begin{aligned} \mathrm{Enc}_\phi: \quad z &= \mu + \sigma_z \epsilon \ \text{ where } \ \mu = ax \ \text{ and } \ \epsilon \sim \mathcal{N}(\epsilon; 0, 1), \\ \mathrm{Dec}_\theta: \quad \hat{x} &= bz. \end{aligned} \tag{37}$$

Here, the encoding parameter ϕ consists of $\{a, \sigma_z\}$, and the decoding parameter θ consists of $\{b\}$. Then the square error is used as a reconstruction loss.

Next, the objective is derived. $D_{\mathrm{KL}x}$ and D_x denote the KL divergence and reconstruction loss at x, respectively. We further assume that D_x uses a square error. Then we define the loss objective at x as $L_x = D_x + \beta D_{\mathrm{KL}x}$. Using Eq. 11, $D_{\mathrm{KL}x}$ can be evaluated as:

$$\begin{aligned} D_{\mathrm{KL}x} &= -\log(\sigma_z \, p(\mu)) - \frac{1}{2}\log 2\pi e \\ &= -\log(\sigma_z \, \mathcal{N}(ax; 0, 1)) - \frac{1}{2}\log 2\pi e \\ &= -\log \sigma_x + \frac{a^2 x^2}{2} - \frac{1}{2}. \end{aligned} \tag{38}$$

Then, D_x is evaluated as:

$$\begin{aligned} D_x &= E_{\epsilon\sim\mathcal{N}(\epsilon;0,1)}\left[(x - \mathrm{Dec}_\theta(\mathrm{Enc}_\phi(x)))^2\right] \\ &= E_{\epsilon\sim\mathcal{N}(\epsilon;0,1)}\left[(x - (b(ax + \sigma_z\epsilon)))^2\right] \\ &= \int \left((ab-1)^2 x^2 + 2(ab-1)xb\sigma_z\epsilon + b^2 {\sigma_z}^2 \epsilon^2\right)\mathcal{N}(\epsilon; 0, 1)\, \mathrm{d}\epsilon \\ &= (ab-1)^2 x^2 + b^2 {\sigma_z}^2. \end{aligned} \tag{39}$$

By averaging L_x over $x \sim \mathcal{N}(x; 0, {\sigma_x}^2)$, the objective L to minimize is derived as:

$$\begin{aligned} L &= E_{x\sim\mathcal{N}(x;0,{\sigma_x}^2)}[L_x] \\ &= \int \left((ab-1)^2 x^2 + b^2{\sigma_z}^2 + \beta\left(-\log\sigma_z + \frac{a^2x^2}{2} - \frac{1}{2}\right)\right)\mathcal{N}(x; 0, {\sigma_x}^2)\, \mathrm{d}x. \\ &= (ab-1)^2{\sigma_x}^2 + b^2{\sigma_z}^2 + \beta\left(-\log\sigma_z + \frac{a^2{\sigma_x}^2}{2} - \frac{1}{2}\right). \end{aligned} \tag{40}$$

Here, the first term $(ab-1)^2{\sigma_x}^2$ and the second term $b^2{\sigma_z}^2$ in the last line are corresponding to the transform loss D_{T} and coding loss D_{C}, respectively.

By solving $\mathrm{d}L/\mathrm{d}a = 0$, $\mathrm{d}L/\mathrm{d}b = 0$, and $\mathrm{d}L/\mathrm{d}\sigma_z = 0$, a , b, and σ_z are derived as follows:

$$\begin{aligned} a &= 1/\sigma_x, \\ b &= \frac{\sigma_x\left(1 + \sqrt{1 - 2\beta/{\sigma_x}^2}\right)}{2}, \\ \sigma_z &= \frac{2\sqrt{\beta/2}}{\sigma_x\left(1 + \sqrt{1 - 2\beta/{\sigma_x}^2}\right)}. \end{aligned} \tag{41}$$

From Eq. 41, D_{T} and D_{C} are derived as:

$$\begin{aligned} D_{\mathrm{T}} &= \left(\frac{\sqrt{1 - 2\beta/{\sigma_x}^2} - 1}{2}\right)^2 {\sigma_x}^2, \\ D_{\mathrm{C}} &= \beta/2. \end{aligned} \tag{42}$$

As shown in section 4.1, the added noise, $\beta/2$, should be reasonably smaller than the data variance ${\sigma_x}^2$. If ${\sigma_x}^2 \gg \beta$, b and σ_z in Eq. 41 can be approximated as:

$$D_{\mathrm{T}} \simeq \frac{(\beta/2)^2}{{\sigma_x}^2} = \frac{\beta/2}{{\sigma_x}^2} D_{\mathrm{C}}. \tag{43}$$

As shown in this equation, $D_{\mathrm{T}}/D_{\mathrm{C}}$ is small in the VAE where the added noise is reasonably small, and D_{T} can be ignored.

Note that the distribution of $\mu = a\, x = x/\sigma_x$, i.e., $q_\phi(\mu)$, is derived as $\mathcal{N}(\mu; 0, 1)$ by scaling $p(x) = \mathcal{N}(x; 0, {\sigma_x}^2)$ with a factor of $a = 1/\sigma_x$. Thus $q_\phi(\mu)$ is equivalent to the prior of z, i.e., $\mathcal{N}(z; 0, 1)$ in this simple VAE case.

Next, the relation to the Wiener filter (Wiener, 1964) is discussed. The Wiener filter is one of the most basic, but most important theories for signal restoration. We consider an simple 1-dimensional Gaussian process. Let $x \sim \mathcal{N}(x; 0, \sigma_x^2)$ be input data. Then, x is scaled by s, and a Gaussian noise $n \sim \mathcal{N}(n; 0, \sigma_n^2)$ is added. Thus, $y = s\, x + n$ is observed. From the Wiener filter theory, the estimated value with minimum distortion, $\hat{x}$ can be formulated as:

$$\hat{x} = \frac{s{\sigma_x}^2}{s^2{\sigma_x}^2 + {\sigma_n}^2} y. \tag{44}$$

In this case, the estimation error is derived as:

$$E[(\hat{x} - x)^2] = \frac{{\sigma_n}^4}{(s^2{\sigma_x}^2 + {\sigma_n}^2)^2}{\sigma_x}^2 + \frac{s^2{\sigma_x}^4}{(s^2{\sigma_x}^2 + {\sigma_n}^2)^2}{\sigma_n}^2 = \frac{{\sigma_x}^2}{{\sigma_x}^2 + ({\sigma_n}^2/s^2)}({\sigma_n}^2/s^2). \tag{45}$$

In the second equation, the first term is corresponding to the transform loss, and the second term is corresponding to the coding loss. Here the ratio of the transform loss and coding loss is derived as ${\sigma_n}^2/(s^2{\sigma_x}^2)$. By appying $s = 1/\sigma_x$ and $\sigma_n = \sigma_z$ to ${\sigma_n}^2/(s^2{\sigma_x}^2)$ and assuming $\sigma_x^2 \gg \beta/2$, this ratio can be described as:

$$\frac{{\sigma_n}^2}{s^2{\sigma_x}^2} = {\sigma_z}^2 = \frac{\beta/2}{\sigma_x^2} \frac{4}{\left(1 + \sqrt{1 - 2\beta/{\sigma_x}^2}\right)^2} = \frac{\beta/2}{\sigma_x^2} + O\left(\left(\frac{\beta/2}{\sigma_x^2}\right)^2\right). \tag{46}$$

This result is consistent with Eq. 43, implying that optimized VAE and the Wiener filter show similar behaviours.

A.4. Proof of Lemma 4 : Derivation of the orthogonality

Lemma 4 is proved by examining the minimum condition of $L_{\boldsymbol{x}}$ at $\boldsymbol{x}$. The proof outline is similar to Kato et al. (2020) while $\sigma_{j(\boldsymbol{x})}$ should be also considered as a variable in our derivation.

We first show the following mathematical formula which is used our derivation. Let $\boldsymbol{A}$ be a regular matrix and $\boldsymbol{a}_i$ be its i-th column vector. $\tilde{\boldsymbol{a}}_i$ denotes the i-th column vector of a cofactor matrix for $\boldsymbol{A}$. Then the following equation holds mathematically.

$$\frac{\mathrm{d} \log|\det(\boldsymbol{A})|}{\mathrm{d}\boldsymbol{a}_i} = \frac{\mathrm{d} \log|\det(\boldsymbol{A})|}{\mathrm{d}\det(\boldsymbol{A})} \frac{\mathrm{d}\det(\boldsymbol{A})}{\mathrm{d}\boldsymbol{a}_i} = \frac{1}{\det(\boldsymbol{A})} \tilde{\boldsymbol{a}}_i. \tag{47}$$

Let $\tilde{\boldsymbol{x}}_{\mu_j}$ be the j-th column vector of a cofactor matrix for Jacobian matrix $\partial \boldsymbol{x}/\partial \boldsymbol{\mu}_{(\boldsymbol{x})}$. Using the formula in Eq. 47, the partial derivative of $L_{\boldsymbol{x}}$ by $\boldsymbol{x}_{\mu_j}$ is described by

$$\frac{\partial L_{\boldsymbol{x}}}{\partial \boldsymbol{x}_{\mu_j}} = 2{\sigma_{j(\boldsymbol{x})}}^2 \boldsymbol{G}_x \boldsymbol{x}_{\mu_j} - \frac{\beta}{\det\left(\partial \boldsymbol{x}/\partial \boldsymbol{\mu}_{(\boldsymbol{x})}\right)} \tilde{\boldsymbol{x}}_{\mu_j}. \tag{48}$$

Note that ${}^t\boldsymbol{x}_{\mu_k} \cdot \tilde{\boldsymbol{x}}_{\mu_j} = \det(\partial \boldsymbol{x}/\partial \boldsymbol{z})\, \delta_{jk}$ holds by the cofactor's property. Here, $\cdot$ denotes the dot product, and δ_{jk} denotes the Kronecker delta. By setting Eq. 48 to zero and multiplying ${}^t\boldsymbol{x}_{z_k}$ from the left, we have the next orthogonal form of $\boldsymbol{x}_{\mu_j}$:

$$(2{\sigma_{j(\boldsymbol{x})}}^2/\beta)\, {}^t\boldsymbol{x}_{\mu_k} \boldsymbol{G}_x \boldsymbol{x}_{\mu_j} = \delta_{jk}. \tag{49}$$

Next, the partial derivative of $L_{\boldsymbol{x}}$ by $\sigma_{j(\boldsymbol{x})}$ is derived as:

$$\frac{\partial L_{\boldsymbol{x}}}{\partial \sigma_{j(\boldsymbol{x})}} = 2\sigma_{j(\boldsymbol{x})}\, {\boldsymbol{x}_{\mu_j}}^t\, \boldsymbol{G}_x \boldsymbol{x}_{\mu_j} - \frac{\beta}{\sigma_{j(\boldsymbol{x})}}. \tag{50}$$

By setting Eq. 50 to zero, we have the next equation:

$$(2{\sigma_{j(\boldsymbol{x})}}^2/\beta)\, {}^t\boldsymbol{x}_{\mu_j} \boldsymbol{G}_x \boldsymbol{x}_{\mu_j} = 1. \tag{51}$$

Note that Eq. 51 is a part of Eq. 49 where $j = k$. As a result, the condition to minimize $L_{\boldsymbol{x}}$ is derived as Eq. 49.

A.5. Proof of Proposition 1: Estimation of input data distribution in the metric space

This equation explains the derivation of Eq. 19 in Proposition 1. Using Eq. 16, the third equation in Eq. 19 is derived as:

$$p(\boldsymbol{y}) = \prod_j^n p(y_j) = \prod_j^n (\mathrm{d}y_j/\mathrm{d}\mu_{j(\boldsymbol{x})})^{-1} p(\mu_j) = \prod_j^n p(\mu_j) \prod_j^n \frac{\sigma_{j(\boldsymbol{x})}}{\sqrt{\beta/2}} = (\beta/2)^{n/2} p(\boldsymbol{\mu}_{j(\boldsymbol{x})}) \prod_j^n \sigma_{j(\boldsymbol{x})}. \tag{52}$$

This shows that the posterior variance $\sigma_{j(\boldsymbol{x})}$ bridges between the distributions of data and prior. Thus the prior close to the data distribution will facilitate training, where $\sigma_{j(\boldsymbol{x})}$ is close to constant.

The fourth equation in Eq. 19 in Proposition 1 is derived by applying Eq. 16 to Eq. 13 and arranging the result. Let $L_{\min \boldsymbol{x}}$ be a minimum of $L_{\boldsymbol{x}}$ at $\boldsymbol{x}$. $D_{\min \boldsymbol{x}}$ and $R_{\min \boldsymbol{x}}$ denote a coding loss and KL divergence in $L_{\min \boldsymbol{x}}$, respectively.

First, $D_{\min \boldsymbol{x}}$ is derived. The next equation holds from Eq. 14.

$$\sigma_{j(\boldsymbol{x})}{}^2 \, {}^t\boldsymbol{x}_{\mu_j} \boldsymbol{G}_x \boldsymbol{x}_{\mu_j} = \beta/2. \tag{53}$$

By applying Eq. 53 to the first term of Eq. 13, $D_{\min \boldsymbol{x}}$ is derived as:

$$D_{\min \boldsymbol{x}} = \sum_j^n \sigma_{j(\boldsymbol{x})}{}^2 \, {}^t\boldsymbol{x}_{\mu_j} \boldsymbol{G}_x \boldsymbol{x}_{\mu_j} = n\beta/2. \tag{54}$$

This implies that the reconstruction loss is constant for all inputs at the minimum condition.

Second, $R_{\min \boldsymbol{x}}$ is derived. From Eq. 52, the next equation holds.

$$p(\boldsymbol{\mu}_{j(\boldsymbol{x})}) \prod_j^n \sigma_{j(\boldsymbol{x})} = (\beta/2)^{-n/2} \, p(\boldsymbol{y}). \tag{55}$$

By applying Eq. 55 to the second equation of Eq. 10, $R_{\min \boldsymbol{x}}$ is derived as:

$$R_{\min \boldsymbol{x}} = -\log p(\boldsymbol{y}) - \frac{n \log(\beta \pi e)}{2}. \tag{56}$$

As a result, the minimum value of the objective $L_{\min \boldsymbol{x}}$ is derived as:

$$L_{\min \boldsymbol{x}} = D_{\min \boldsymbol{x}} + \beta R_{\min \boldsymbol{x}} = -\beta \log p(\boldsymbol{y}) + \frac{n\beta}{2}(1 - \log(\beta \pi e)). \tag{57}$$

As a result, $p(\boldsymbol{x})$ can be evaluated as:

$$\exp(-L_{\min \boldsymbol{x}}/\beta) = p(\boldsymbol{y}) \exp(-\frac{n(1 - \log(\beta \pi e))}{2}) \propto p(\boldsymbol{y}) \simeq p_{\boldsymbol{G}_{\boldsymbol{x}}}(\boldsymbol{x}). \tag{58}$$

This result implies that the VAE objective converges to the log-likelihood of the input $\boldsymbol{x}$ at the optimized condition as expected.

A.6. Proof of Proposition 2: Estimation of data distribution in the input space

This appendix shows the derivation of variables in Eqs. 19 and 21. When we estimate a probability in real dataset, we use an approximation of $L_{\boldsymbol{x}}$. First, the derivation of $L_{\boldsymbol{x}}$ approximation for the input $\boldsymbol{x}$ is presented. Then, the PDF ratio between the input space and inner product space is explained for the cases $m = n$ and $m > n$.

Derivation of $L_{\boldsymbol{x}}$ approximation for the input x of real data:
As shown in in Eq. 1, $L_{\boldsymbol{x}}$ is denoted as $-E_{\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\,\cdot\,] + \beta D_{\mathrm{KL}}(\,\cdot\,)$. We approximate $E_{\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\,\cdot\,]$ as $\frac{1}{2}(D(\boldsymbol{x}, \mathrm{Dec}_\theta(\boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{\sigma}_{\boldsymbol{x}})) + D(\boldsymbol{x}, \mathrm{Dec}_\theta(\boldsymbol{\mu}_{\boldsymbol{x}} - \boldsymbol{\sigma}_{\boldsymbol{x}})))$, i.e., the average of two samples, instead of the average over $\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x})$. $D_{\mathrm{KL}}(\,\cdot\,)$ can be calculated from $\boldsymbol{\mu}_{\boldsymbol{x}}$ and $\boldsymbol{\sigma}_{\boldsymbol{x}}$ using Eq. 2.

The PDF ratio in the case $m = n$:
The PDF ratio for $m = n$ is a Jacobian determinant between two spaces. First, $(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}})^T \boldsymbol{G}_{\boldsymbol{x}} (\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}}) = \boldsymbol{I}_m$ holds from

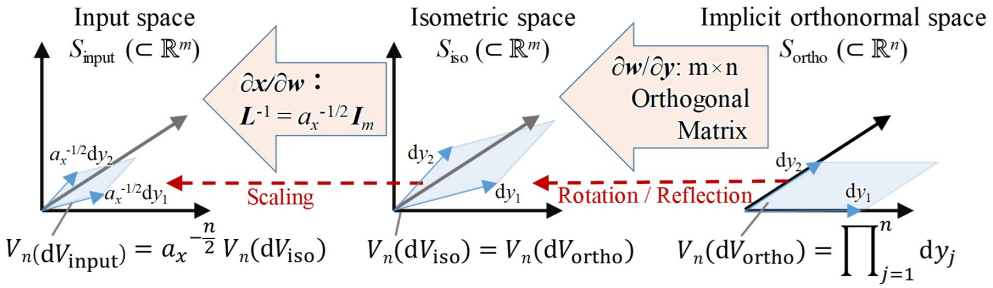


Figure 9. Projection of the volume element from the implicit orthonormal space to the isometric space and input space. $V_n(\cdot)$ denotes n-dimensional volume.

Eq. 17. $|\partial \boldsymbol{x}/\partial \boldsymbol{y}|^2 \ |\boldsymbol{G}_{\boldsymbol{x}}| = 1$ also holds by calculating the determinant. Finally, $|\partial \boldsymbol{x}/\partial \boldsymbol{y}|$ is derived as $|\boldsymbol{G}_x|^{1/2}$ using $|\partial \boldsymbol{y}/\partial \boldsymbol{x}| = |\partial \boldsymbol{x}/\partial \boldsymbol{y}|^{-1}$.

The PDF ratio in the case $m > n$ and $\boldsymbol{G}_{\boldsymbol{x}} = a_{\boldsymbol{x}} \boldsymbol{I}_m$:
Although the strict derivation needs the treatment of the Riemannian manifold, we provide a simple explanation in this appendix. Here, it is assumed that $D_{\mathrm{KL}(j)}(\cdot) > 0$ holds for all $j = [1, ..n]$. If $D_{\mathrm{KL}(j)}(\cdot) = 0$ for some j, n is replaced by the number of latent variables with $D_{\mathrm{KL}(j)}(\cdot) > 0$.

For the implicit isometric space $S_{\text{iso}}(\subset \mathbb{R}^m)$, there exists a matrix $\boldsymbol{L}_{\boldsymbol{x}}$ such that both $\boldsymbol{y} = \boldsymbol{L}_{\boldsymbol{x}} \boldsymbol{x}$ and $\boldsymbol{G}_{\boldsymbol{x}} = {}^t\boldsymbol{L}_{\boldsymbol{x}} \boldsymbol{L}_{\boldsymbol{x}}$ holds. $\boldsymbol{w}$ denotes a point in S_{iso}, i.e., $\boldsymbol{w} \in S_{\text{iso}}$. Because $\boldsymbol{G}_{\boldsymbol{x}}$ is assumed as $a_{\boldsymbol{x}} \boldsymbol{I}_m$ in Section 4.3, $\boldsymbol{L}_{\boldsymbol{x}} = a_{\boldsymbol{x}}{}^{1/2} \boldsymbol{I}_m$ holds. Then, the mapping function $\boldsymbol{w} = h(\boldsymbol{x})$ between S_{input} and S_{iso} is defined, such that:

$$\frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}} = \boldsymbol{L}_{\boldsymbol{x}}, \quad \text{and} \quad h(\boldsymbol{x}^{(0)}) = \boldsymbol{w}^{(0)} \quad \text{for} \quad \exists\, \boldsymbol{x}^{(0)} \in S_{\text{input}} \quad \text{and} \quad \exists\, \boldsymbol{w}^{(0)} \in S_{\text{iso}}. \tag{59}$$

Let $\delta \boldsymbol{x}$ and $\delta \boldsymbol{w}$ are infinitesimal displacements around $\boldsymbol{x}$ and $\boldsymbol{w} = h(\boldsymbol{x})$, such that $\boldsymbol{w} + \delta \boldsymbol{w} = h(\boldsymbol{x} + \delta \boldsymbol{x})$. Then the next equation holds from Eq. 59:

$$\delta \boldsymbol{w} = \boldsymbol{L}_{\boldsymbol{x}} \delta \boldsymbol{x}. \tag{60}$$

Let $\delta \boldsymbol{x}^{(1)}$, $\delta \boldsymbol{x}^{(2)}$, $\delta \boldsymbol{w}^{(1)}$, and $\delta \boldsymbol{w}^{(2)}$ be two arbitrary infinitesimal displacements around $\boldsymbol{x}$ and $\boldsymbol{w} = h(\boldsymbol{x})$, such that $\delta \boldsymbol{w}^{(1)} = \boldsymbol{L}_{\boldsymbol{x}} \delta \boldsymbol{x}^{(1)}$ and $\delta \boldsymbol{w}^{(2)} = \boldsymbol{L}_{\boldsymbol{x}} \delta \boldsymbol{x}^{(2)}$. Then the following equation holds, where $\cdot$ denotes the dot product.

$${}^t\delta \boldsymbol{x}^{(1)} \boldsymbol{G}_{\boldsymbol{x}} \delta \boldsymbol{x}^{(2)} = {}^t(\boldsymbol{L}_{\boldsymbol{x}} \delta \boldsymbol{x}^{(1)})(\boldsymbol{L}_{\boldsymbol{x}} \delta \boldsymbol{x}^{(2)}) = \delta \boldsymbol{w}^{(1)} \cdot \delta \boldsymbol{w}^{(2)}. \tag{61}$$

This equation shows the isometric mapping from the inner product space for $\boldsymbol{x} \in S_{\text{input}}$ with the metric tensor $\boldsymbol{G}_{\boldsymbol{x}}$ to the Euclidean space for $\boldsymbol{w} \in S_{\text{iso}}$.

Note that all of the column vectors in the Jacobian matrix $\partial \boldsymbol{x}/\partial \boldsymbol{y}$ also have a unit norm and are orthogonal to each other in the metric space for $\boldsymbol{x} \in S_{\text{input}}$ with the metric tensor $\boldsymbol{G}_{\boldsymbol{x}}$. Therefore, the $m \times n$ Jacobian matrix $\partial \boldsymbol{w}/\partial \boldsymbol{y}$ should have a property that all of the column vectors have a unit norm and are orthogonal to each other in the Euclidean space.

Then n-dimensional space which is composed of the meaningful dimensions from the implicit isometric space is named as the implicit orthonormal space S_{ortho}. Figure 9 shows the projection of the volume element from the implicit orthonormal space to the isometric space and input space. Let $\mathrm{d}V_{\text{ortho}}$ be an infinitesimal n-dimensional volume element in S_{ortho}. This volume element is a n-dimensional rectangular solid having each edge length $\mathrm{d}y_j$. Let $V_n(\mathrm{d}V_{\mathrm{X}})$ be the n-dimensional volume of a volume element $\mathrm{d}V_{\mathrm{X}}$. Then, $V_n(\mathrm{d}V_{\text{ortho}}) = \prod_j^n \mathrm{d}y_j$ holds. Next, $\mathrm{d}V_{\text{ortho}}$ is projected to n dimensional infinitesimal element $\mathrm{d}V_{\text{iso}}$ in S_{iso} by $\partial \boldsymbol{w}/\partial \boldsymbol{y}$. Because of the orthonormality, $\mathrm{d}V_{\text{iso}}$ is equivalent to the rotation / reflection of $\mathrm{d}V_{\text{ortho}}$, and $V_n(\mathrm{d}V_{\text{iso}})$ is the same as $V_n(\mathrm{d}V_{\text{ortho}})$, i.e., $\prod_j^n \mathrm{d}y_j$. Then, $\mathrm{d}V_{\text{iso}}$ is projected to n-dimensional element $\mathrm{d}V_{\text{input}}$ in S_{input} by $\partial \boldsymbol{x}/\partial \boldsymbol{w} = \boldsymbol{L}_{\boldsymbol{x}}^{-1} = a_{\boldsymbol{x}}{}^{-1/2} \boldsymbol{I}_m$. Because each dimension is scaled equally by the scale factor $a_{\boldsymbol{x}}{}^{-1/2}$, $V_n(\mathrm{d}V_{\text{input}}) = \prod_j^n a_{\boldsymbol{x}}{}^{-1/2} \mathrm{d}y_j = a_{\boldsymbol{x}}{}^{-n/2} V_n(\mathrm{d}V_{\text{ortho}})$ holds. Here, the ratio of the volume element between S_{input} and S_{ortho} is $V_n(\mathrm{d}V_{\text{input}})/V_n(\mathrm{d}V_{\text{ortho}}) = a_{\boldsymbol{x}}{}^{-n/2}$. Note that the PDF ratio is derived by the reciprocal of $V_n(\mathrm{d}V_{\text{input}})/V_n(\mathrm{d}V_{\text{ortho}})$. As a result, the PDF ratio is derived as $a_{\boldsymbol{x}}{}^{n/2}$.

A.7. Proof of proposition 3: Determination of the meaningful dimension for representation

This appendix explain the derivation of Proposition 3. Here, we estimate the KL divergence, i.e., a rate for the dimensions whose variance is less than β. As shown later, the discussion in this appendix is closely related with Rate-distortion theory (Berger, 1971; Pearlman & Said, 2011; Goyal, 2001).

Let L_{G}, D_{G}, and R_{G} be averages of $L_{\min \boldsymbol{x}}$, $D_{\min \boldsymbol{x}}$, and $R_{\min \boldsymbol{x}}$ in Appendix A.5 over $\boldsymbol{x} \sim p(\boldsymbol{x})$, respectively. Here, $L_{\mathrm{G}} = D_{\mathrm{G}} + \beta R_{\mathrm{G}}$ holds by definition. Since $D_{\min \boldsymbol{x}}$ is a constant $n\beta/2$ as in Eq. 54, D_{G} is derived as:

$$D_{\mathrm{G}} = n\beta/2. \tag{62}$$

As D_{G} is constant, the minimum condition of L_{G} is equivalent to that of R_{G}. Let $D_{\mathrm{KLmin}\ j(x)}$ be a KL divergence of the j-th dimensional component at the minimum condition. Here, $R_{\min \boldsymbol{x}} = \sum_j^n D_{\mathrm{KLmin}\ j(x)}$ holds by definition. Eq. 11 holds for for small $\beta/2$. Thus, we can approximate $D_{\mathrm{KLmin}\ j(x)}$ for small $\beta/2$ from Eqs. 16 and 11 as:

$$\begin{aligned}
D_{\mathrm{KLmin}\ j(x)} &\simeq -\log\left(\sigma_{j(\boldsymbol{x})}\, p(\mu_{j(\boldsymbol{x})})\right) - \frac{\log 2\pi e}{2} \\
&= -\log\left(\sqrt{\beta/2}\, p(y_j)\right) - \frac{\log 2\pi e}{2} \\
&= -\log\left(p(y_j)\right) - \frac{\log \beta\pi e}{2} \\
&= -\log\left(p(y_j)\right) - H(\mathcal{N}(y_j; 0, \beta/2)).
\end{aligned} \tag{63}$$

Here, $H(\mathcal{N}(y_j; 0, \beta/2))$ denotes a entropy of the Gaussian with variance $\beta/2$. Next, R_{G} is expressed as:

$$\begin{aligned}
R_{\mathrm{G}} &= E_{\boldsymbol{x}\sim p(\boldsymbol{x})}[R_{\min \boldsymbol{x}}] \\
&= E_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\sum_j^n D_{\mathrm{KLmin}\ j(x)}\right] \\
&\simeq -\int p(\boldsymbol{x}) \sum_j^n (-\log\left(p(y_j)\right) - H(\mathcal{N}(y_j; 0, \beta/2)), 0) \mathrm{d}\boldsymbol{x}. \\
&= -\int p(\boldsymbol{y}) \left|\det\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}}\right)\right|^{-1} \sum_j^n (-\log\left(p(y_j)\right) - H(\mathcal{N}(y_j; 0, \beta/2)), 0) \left|\det\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}}\right)\right| \mathrm{d}\boldsymbol{y} \\
&= -\int p(\boldsymbol{y}) \sum_j^n (-\log\left(p(y_j)\right) - H(\mathcal{N}(y_j; 0, \beta/2)), 0) \mathrm{d}\boldsymbol{y}. \\
&= \sum_j^n \left(-\int p(y_j) \log\left(p(y_j)\right) \mathrm{d}y_j - H(\mathcal{N}(y_j; 0, \beta/2))\right).
\end{aligned} \tag{64}$$

Note that the KL-divergence is always equal or greater than 0 by definition. By considering this, R_{G} is further approximated as:

$$R_{\mathrm{G}} \simeq \sum_j^n \max\left(-\int p(y_j) \log\left(p(y_j)\right) \mathrm{d}y_j - H(\mathcal{N}(y_j; 0, \beta/2)),\ \ 0\right). \tag{65}$$

Note that the approximation of Eq. 65 is reasonable from the Rate-distortion theory and optimal transform coding theory (Berger, 1971; Pearlman & Said, 2011; Goyal, 2001). The outline of Rate-distortion theory and optimal transform coding is explained in Appendix B.6. The term $-\int p(y_j) \log p(y_j) \mathrm{d}y_j$ is the entropy of y_j. Thus, the optimal implicit isometric space is derived such that the entropy of data representation is minimum. When the data manifold has a disentangled property in the given metric by nature, each y_j will capture a disentangled feature with minimum entropy such that the mutual information between implicit isometric components becomes minimized. This is analogous to PCA for Gaussian data, which gives the disentangled representation with minimum entropy in SSE. Considering the similarity to the PCA eigenvalues, the variance of y_j will indicate the importance of each dimension.

Thus, if the entropy of y_j is larger than $H(\mathcal{N}(0, \beta/2)$, then it is reasonable that $D_{\mathrm{KLmin}\ j(x)} > 0$ holds. By contrast, if the entropy of y_j is less than $H(\mathcal{N}(0, \beta/2)$, then $D_{\mathrm{KLmin}\ j(x)} = 0$ will hold. In such dimensions, $\sigma_{j(\boldsymbol{x})} = 1$, $\mu_{j(\boldsymbol{x})} = 0$, and $D_{\mathrm{KL}j(x)} = 0$ will hold. In addition, ${\sigma_{j(\boldsymbol{x})}}^2\ {}^t\boldsymbol{x}_{\mu_{j(\boldsymbol{x})}} \boldsymbol{G}_{\boldsymbol{x}} \boldsymbol{x}_{\mu_{j(\boldsymbol{x})}}$ will be close to 0 because this needs not to be balanced with $D_{\mathrm{KL}j(x)}$.

These properties of VAE can be clearly explained by rate-distortion theory (Berger, 1971), which has been successfully applied to transform coding such as image / audio compression. Appendix B.6 explains that VAE can be interpreted as an optimal transform coding with non-linear scaling of latent space.

Thus, latent variables with variances from the largest to the n-th with $D_{\mathrm{KL}j(x)} > 0$ are sufficient for the representation and the dimensions with $D_{\mathrm{KL}j(x)} = 0$ can be ignored, allowing the reduction of the dimension n for $\boldsymbol{z}$.

A.8. Proof of proposition 4: Derivation of the estimated variance

This appendix explains the derivation of quantitative importance for each dimension in Eq. 22 of Proposition 4.

First, we set y_j to 0 at $\mu_{j(\boldsymbol{x})} = 0$ to derive y_j value from $\mathrm{d}y_j/\mathrm{d}\mu_{j(\boldsymbol{x})}$ in Eq. 16. We also assume that the prior distribution is $\mathcal{N}(\boldsymbol{z}; 0, \boldsymbol{I}_n)$. The variance is derived by the subtraction of $E[y_j]^2$, i.e., the square of the mean, from $E[y_j^2]$, i.e., the square mean. Thus, the approximations of both $E[y_j]$ and $E[y_j^2]$ are needed.

First, the approximation of the mean $E[y_j]$ is explained. Because the cumulative distribution functions (CDFs) of y_j are the same as CDF of $\mu_{j(\boldsymbol{x})}$, the following equations hold:

$$\int_{-\infty}^{0} p(y_j)\mathrm{d}y_j = \int_{-\infty}^{0} p(\mu_{j(\boldsymbol{x})})\mathrm{d}\mu_{j(\boldsymbol{x})} = 0.5,\ \int_{0}^{\infty} p(y_j)\mathrm{d}y_j = \int_{0}^{\infty} p(\mu_{j(\boldsymbol{x})})\mathrm{d}\mu_{j(\boldsymbol{x})} = 0.5. \tag{66}$$

This equation means that the median of the y_j distribution is 0. Because the mean and median are close in most cases, the mean $E[y_j]$ can be approximated as 0. As a result, the variance of y_j can be approximated by the square mean $E[y_j^2]$.

Second, the approximation of the square mean $E[y_j^2]$ is explained. Since we assume the manifold has a disentangled property by nature, the standard deviation of the posterior $\sigma_{j(\boldsymbol{x})}$ is assumed as a function of $\mu_{j(\boldsymbol{x})}$, regardless of $\boldsymbol{x}$. This function is denoted as $\sigma_j(\mu_{j(\boldsymbol{x})})$. For ≥ 0, y_j is approximated as follows, using Eq. 16 and replacing the average of $1/\sigma_j(\acute{\mu}_{j(\boldsymbol{x})})$ over $\acute{\mu}_{j(\boldsymbol{x})} = [0, \mu_{j(\boldsymbol{x})}]$ by $1/\sigma_j(\mu_{j(\boldsymbol{x})})$:

$$y_j = \int_0^{\mu_{j(\boldsymbol{x})}} \frac{\mathrm{d}y_j}{\mathrm{d}\acute{z}_j}\mathrm{d}\acute{z}_j = \sqrt{\frac{\beta}{2}} \int_0^{z_i} \frac{1}{\sigma_j(\acute{\mu}_{j(\boldsymbol{x})})}\mathrm{d}\acute{z}_i \simeq \sqrt{\frac{\beta}{2}} \frac{1}{\sigma_j(\mu_{j(\boldsymbol{x})})} \int_0^{\mu_{j(\boldsymbol{x})}} \mathrm{d}\acute{z}_j = \sqrt{\frac{\beta}{2}} \frac{\mu_{j(\boldsymbol{x})}}{\sigma_j(\mu_{j(\boldsymbol{x})})}. \tag{67}$$

The same approximation is applied to $z_i < 0$. Then the square mean of y_i is approximated as follows, assuming that the correlation between $\sigma(\mu_{j(\boldsymbol{x})})^{-2}$ and ${\mu_{j(\boldsymbol{x})}}^2$ is low:

$$\int {y_j}^2 p(y_j)\mathrm{d}y_j \simeq \frac{\beta}{2} \int \left(\frac{\mu_{j(\boldsymbol{x})}}{\sigma_j(\mu_{j(\boldsymbol{x})})}\right)^2 p(\mu_{j(\boldsymbol{x})})\mathrm{d}\mu_{j(\boldsymbol{x})} \simeq \frac{\beta}{2} \int \sigma_j(\mu_{j(\boldsymbol{x})})^{-2} p(\mu_{j(\boldsymbol{x})})\mathrm{d}\mu_{j(\boldsymbol{x})} \int {\mu_{j(\boldsymbol{x})}}^2 p(\mu_{j(\boldsymbol{x})})\mathrm{d}\mu_{j(\boldsymbol{x})}. \tag{68}$$

Finally, the square mean of y_i is approximated as the following equation, using $\int {\mu_{j(\boldsymbol{x})}}^2 p(\mu_{j(\boldsymbol{x})})\mathrm{d}\mu_{j(\boldsymbol{x})} = 1$ and replacing $\sigma_j(\mu_{j(\boldsymbol{x})})^2$ by ${\sigma_{j(\boldsymbol{x})}}^2$, i.e., the posterior variance derived from the input data:

$$\int {y_j}^2 p(y_j)\mathrm{d}y_j \simeq \frac{\beta}{2} \int \sigma_j(\mu_{j(\boldsymbol{x})})^{-2} p(\mu_{j(\boldsymbol{x})})\mathrm{d}\mu_{j(\boldsymbol{x})} \simeq \frac{\beta}{2} \underset{\boldsymbol{\mu}_{j(\boldsymbol{x})} \sim p(\mu_{j(\boldsymbol{x})})}{E} [\sigma_j(\mu_{j(\boldsymbol{x})})^{-2}] \simeq \frac{\beta}{2} \underset{\boldsymbol{x} \sim p(\boldsymbol{x})}{E} [{\sigma_{j(\boldsymbol{x})}}^{-2}]. \tag{69}$$

Although some rough approximations are used in the expansion, the estimated variance in the last equation seems still reasonable, because $\sigma_{j(\boldsymbol{x})}$ shows a scale factor between y_j and $\mu_{j(\boldsymbol{x})}$ while the variance of $\mu_{j(\boldsymbol{x})}$ is always 1 for the prior $\mathcal{N}(\mu_{j(\boldsymbol{x})}; 0, 1)$. Considering the variance of the prior $\int {\mu_{j(\boldsymbol{x})}}^2 p(\mu_{j(\boldsymbol{x})})\mathrm{d}\mu_{j(\boldsymbol{x})}$ in the expansion, this estimation method can be applied to any prior distribution.

B. Detailed relation to prior works

This section first describes the clear formulation of ELBO in VAE by utilizing isometric embedding. Then the detailed relationship, including correction, with previous works are explained.

B.1. Derivation of ELBO with clear and quantitative form

This section clarifies that the ELBO value after optimization becomes close to the log-likelihood of input data in the metric space (not input space), by the theoretical derivation of the reconstruction loss and KL divergence via isometric embedding.

We derive the ELBO (without β) at $\boldsymbol{x}$ in Eq. 1, i.e., $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}))$ when the objective of β-VAE $L_{\boldsymbol{x}}$ (with β) in Eq. 57, i.e., $L_{\boldsymbol{x}} = D(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \beta D_{\mathrm{KL}}(\cdot)$ is optimised.

First, the reconstruction loss can be rewritten as:

$$E_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] = \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = \int q_\phi(\boldsymbol{y}|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}|\boldsymbol{y}) \mathrm{d}\boldsymbol{y}. \tag{70}$$

Let $\boldsymbol{\mu}_{\boldsymbol{y}(\boldsymbol{x})}$ be a implicit isometric variable corresponding to $\mu_{(\boldsymbol{x})}$. Because the posterior variance in each isometric latent variable is a constant $\beta/2$, $q_\phi(\boldsymbol{y}|\boldsymbol{x}) \simeq \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_{\boldsymbol{y}(\boldsymbol{x})}, (\beta/2)\boldsymbol{I}_n)$ will hold. If $\beta/2$ is small, $p(\hat{\boldsymbol{x}}) \simeq p(\boldsymbol{x})$ will hold. Then, the next equation will hold also using isometricity;

$$p_\theta(\boldsymbol{x}|\boldsymbol{z}) = p_\theta(\boldsymbol{x}|\boldsymbol{y}) = p_\theta(\boldsymbol{x}|\hat{\boldsymbol{x}}) = p(\hat{\boldsymbol{x}}|\boldsymbol{x})p(\boldsymbol{x})/p(\hat{\boldsymbol{x}}) \simeq p(\hat{\boldsymbol{x}}|\boldsymbol{x}) \simeq q_\phi(\boldsymbol{y}|\boldsymbol{x}) \simeq \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_{\boldsymbol{y}(\boldsymbol{x})}, (\beta/2)\boldsymbol{I}_n). \tag{71}$$

Thus the reconstruction loss is estimated as:

$$\begin{aligned} E_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] &\simeq \int \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_{\boldsymbol{y}(\boldsymbol{x})}, (\beta/2)\boldsymbol{I}_n) \log \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_{\boldsymbol{y}(\boldsymbol{x})}, (\beta/2)\boldsymbol{I}_n)\, \mathrm{d}\boldsymbol{y} \\ &= -(n/2)\log(\beta\pi e). \end{aligned} \tag{72}$$

Next, KL divergence is derived from Eq. 56 as:

$$D_{\mathrm{KL}}(\cdot) = R_{\min \boldsymbol{x}} = -\log p(\boldsymbol{y}) - (n/2)\log(\beta\pi e). \tag{73}$$

By summing both terms, ELBO at $\boldsymbol{x}$ can be estimated as

$$\begin{aligned} ELBO &= E_{\boldsymbol{x}\sim p(\boldsymbol{x})}[E_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{\mathrm{KL}}(\cdot)] \\ &\simeq E_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\log p(\boldsymbol{y})] \\ &\simeq E_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\log p(\boldsymbol{x})]. \end{aligned} \tag{74}$$

As a result, ELBO (Eq. 1) in the original form (Kingma & Welling, 2014) is close to the log-likelihood of $\boldsymbol{x}$, regardless $\beta = 1$ or not, when the objective of β-VAE (Higgins et al., 2017) is optimised. Note that $\log p(\boldsymbol{x})$ in Eq.73 is defined in the metric space. This also implies that the representation $\boldsymbol{y}$ depends on the metrics.

Next, the predetermined conditional distribution $p_{\mathbb{R}p}(\boldsymbol{x}|\hat{\boldsymbol{x}})$ used for training and the true conditional distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z}) = p_\theta(\boldsymbol{x}|\hat{\boldsymbol{x}})$ after optimization are examined. Although $p_{\mathbb{R}p}(\boldsymbol{x}|\hat{\boldsymbol{x}})$ and $p_\theta(\boldsymbol{x}|\hat{\boldsymbol{x}})$ are expected to be equivalent after optimization, the theoretical relationship between both is not well discussed. Assume $p_{\mathbb{R}p}(\boldsymbol{x}|\hat{\boldsymbol{x}}) = \mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}, \sigma^2\boldsymbol{I})$. In this case, the metric $D(\boldsymbol{x}, \hat{\boldsymbol{x}})$ is derived as $-\log p_{\mathbb{R}p}(\boldsymbol{x}|\hat{\boldsymbol{x}}) = (1/2\sigma^2)|\boldsymbol{x} - \hat{\boldsymbol{x}}|_2^2 + \mathrm{Const}$. Using Eq. 53, the following equations are derived:

$$E_{p(\boldsymbol{x})}[D(\boldsymbol{x}, \hat{\boldsymbol{x}})] = E_{p(\boldsymbol{x})}\left[(1/2\sigma^2)|\boldsymbol{x} - \hat{\boldsymbol{x}}|_2^2\right] = E_{p(\boldsymbol{x})}\left[(1/2\sigma^2)\sum_i (x_i - \hat{x}_i)^2\right] \simeq n\beta/2. \tag{75}$$

Assume that $\sum_i (x_i - \hat{x}_i)^2$ for all i are equivalent. Then the next equation is derived:

$$E_{p(\boldsymbol{x})}\left[(x_i - \hat{x}_i)^2\right] \simeq \beta\sigma^2. \tag{76}$$

Because the variance of each dimension is $\beta\sigma^2$, the conditional distribution after optimization is estimated as $p_\theta(\boldsymbol{x}|\hat{\boldsymbol{x}}) = \mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}, \beta\sigma^2\boldsymbol{I})$.

If $\beta = 1$, i.e., the original VAE objective, both $p_{\mathbb{R}p}(\boldsymbol{x}|\hat{\boldsymbol{x}})$ and $p_\theta(\boldsymbol{x}|\hat{\boldsymbol{x}})$ are equivalent. This result is consistent with what is expected.

If $\beta \neq 1$, however, $p_{\mathbb{R}p}(\boldsymbol{x}|\hat{\boldsymbol{x}})$ and $p_\theta(\boldsymbol{x}|\hat{\boldsymbol{x}})$ are different. In other words, what β-VAE really does is to scale a variance of the pre-determined conditional distribution in the original VAE by a factor of β as Eq. 76. The detail is explained in Appendix B.3.

If $D(\boldsymbol{x}, \boldsymbol{x} + \delta\boldsymbol{x}) = {}^t\delta\boldsymbol{x}\boldsymbol{G}_{\boldsymbol{x}}\delta\boldsymbol{x} + O(||\delta\boldsymbol{x}||^3)$ is not SSE, by introducing a variable $\acute{\boldsymbol{x}} = {\boldsymbol{L}_{\boldsymbol{x}}}^{-1}\boldsymbol{x}$ where $\boldsymbol{L}_{\boldsymbol{x}}$ satisfies ${}^t\boldsymbol{L}_{\boldsymbol{x}}\boldsymbol{L}_{\boldsymbol{x}} = \boldsymbol{G}_{\boldsymbol{x}}$, the metric $D(\cdot, \cdot)$ can be replaced by SSE in the Euclidean space of $\acute{\boldsymbol{x}}$.

B.2. Relation to Tishby et al. (1999)

The theory described in Tishby et al. (1999), which first proposes the concept of information bottleneck (IB), is consistent with our analysis. Tishby et al. (1999) clarified the behaviour of the compressed representation when the rate-distortion trade-off is optimized. $\boldsymbol{x} \in X$ denotes the signal space with a fixed probability $p(\boldsymbol{x})$ and $\hat{\boldsymbol{x}} \in \hat{X}$ denotes its compressed representation. Let $D(\boldsymbol{x}, \hat{\boldsymbol{x}})$ be a loss metric. Then the rate-distortion trade-off can be described as:

$$L = I(X; \hat{X}) + \beta' \underset{p(\boldsymbol{x},\hat{\boldsymbol{x}})}{E}[D(\boldsymbol{x}, \hat{\boldsymbol{x}})]. \tag{77}$$

By solving this condition, they derive the following equation:

$$p(\hat{\boldsymbol{x}}|\boldsymbol{x}) \propto \exp(-\beta' D(\boldsymbol{x}, \hat{\boldsymbol{x}})). \tag{78}$$

As shown in our discussion above, $p(\hat{\boldsymbol{x}}|\boldsymbol{x}) \simeq \mathcal{N}(\hat{\boldsymbol{x}}; \boldsymbol{x}, (\beta/2)\boldsymbol{I}_m)$ will hold in the metric defined space from our VAE analysis. This result is equivalent to Eq. 78 in their work if $D(\boldsymbol{x}, \hat{\boldsymbol{x}})$ is SSE and β' is set to β^{-1}, as follows:

$$p(\hat{\boldsymbol{x}}|\boldsymbol{x}) \propto \exp(-\beta' D(\boldsymbol{x}, \hat{\boldsymbol{x}})) = \exp\left(-\frac{||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2^2}{2(\beta/2)}\right) \propto \mathcal{N}(\hat{\boldsymbol{x}}; \boldsymbol{x}, (\beta/2)\boldsymbol{I}_m). \tag{79}$$

If $D(\boldsymbol{x}, \hat{\boldsymbol{x}})$ is not SSE, the use of the space transformation explained in appendix B.1 will lead to the same result.

B.3. Relation to β-VAE (Higgins et al., 2017)

This section explains the clear understanding of β-VAE (Higgins et al., 2017), and also corrects some of their theory.

In Higgins et al. (2017), ELBO equation is modified as:

$$E_{p(\boldsymbol{x})}[\, E_{\hat{\boldsymbol{x}} \sim p_\phi(\hat{\boldsymbol{x}}|\boldsymbol{x})}[q_\theta(\boldsymbol{x}|\hat{\boldsymbol{x}})] - \beta D_{\mathrm{KL}}(\cdot)\,]. \tag{80}$$

However, they use the predetermined probabilities of $p_\theta(\hat{\boldsymbol{x}}|\boldsymbol{x})$ such as the Bernoulli and Gaussian distributions in training (described in table 1 in Higgins et al. (2017)). As shown in our appendix G.2, the log-likelihoods of the Bernoulli and Gaussian distributions can be regarded as BCE and SSE metrics, respectively. As a result, the actual objective for training in Higgins et al. (2017) is not Eq. 80, but the objective $L_{\boldsymbol{x}} = D(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \beta D_{\mathrm{KL}}(\cdot)$ in Eq. 3 using BCE and SSE metrics with varying β. Thus ELBO as Eq. 1 form will become $\log p(\boldsymbol{x})$ in the BCE / SSE metric defined space regardless $\beta = 1$ or not, as shown in appendix B.1.

Actually, the equation 80 dose not show the log-likelihood of $\boldsymbol{x}$ after optimization. When $D_{\mathrm{KL}}(\cdot) \simeq -\log p(\boldsymbol{x}) - (n/2)\log(\beta\pi e)$ and $E_{\hat{\boldsymbol{x}} \sim p(\hat{\boldsymbol{x}}|\boldsymbol{x})}[p(\boldsymbol{x}|\hat{\boldsymbol{x}})] \simeq -(n/2)\log(\beta\pi e)$ are applied, the value of Eq. 80 is derived as $\beta \log p(\boldsymbol{x}) + (\beta - 1)(n/2)\log(\beta\pi e)$, which is different from the log-likelihood of $\boldsymbol{x}$ in Eq. 73 if $\beta \neq 1$.

Correctly, what β-VAE really does is only to scale the variance of the pre-determined conditional distribution in the original VAE by a factor of β. In the case the pre-determined conditional distribution is Gaussian $\mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}, \sigma^2\boldsymbol{I})$, the objective of β-VAE can be can be rewritten as a linearly scaled original VAE objective with a Gaussian $\mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}, \beta\sigma^2\boldsymbol{I})$ where the variance is $\beta\sigma^2$ instead of σ^2:

$$\begin{aligned}
E_{q_\phi(\cdot)}[\log \mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}, \sigma^2\boldsymbol{I})] - \beta D_{\mathrm{KL}}(\cdot) &= E_{q_\phi(\cdot)}\left[-\frac{1}{2}\log 2\pi\sigma^2 - \frac{|\boldsymbol{x} - \hat{\boldsymbol{x}}|_2^2}{2\sigma^2}\right] - \beta D_{\mathrm{KL}}(\cdot) \\
&= \beta\left(E_{q_\phi(\cdot)}\left[-\frac{1}{2}\log 2\pi\beta\sigma^2 - \frac{|\boldsymbol{x} - \hat{\boldsymbol{x}}|_2^2}{2\beta\sigma^2}\right] - D_{\mathrm{KL}}(\cdot)\right) \\
&\quad + \frac{\beta}{2}\log 2\pi\beta\sigma^2 - \frac{1}{2}\log 2\pi\sigma^2 \\
&= \beta\left(\underline{E_{q_\phi(\cdot)}[\log \mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}, \beta\sigma^2\boldsymbol{I})] - D_{\mathrm{KL}}(\cdot)}\right) + \mathrm{const.}
\end{aligned} \tag{81}$$

Here, the underlined terms in the last equation is just the ELBO with the predetermined conditional distribution $\mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}, \beta\sigma^2\boldsymbol{I})$. So the optimization of β-VAE objective with the predetermined conditional distribution $\mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}, \sigma^2\boldsymbol{I})$ is just the same as the optimization of the original VAE objective (β=1) with with the predetermined conditional distribution $\mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}, \beta\sigma^2\boldsymbol{I})$.

B.4. Relation to Alemi et al. (2018)

Alemi et al. (2018) discuss the rate-distortion trade-off by the theoretical entropy analysis. Their work is also presumed that the objective $L_{\boldsymbol{x}}$ was not mistakenly distinguished from ELBO, which leads to the incorrect discussion. In their work, the differential entropy for the input H, distortion D, and rate R are derived carefully. They suggest that VAE with $\beta = 1$ is sensitive (unstable) because D and R can be arbitrary value on the line $R = H - \beta D = H - D$. Furthermore, they also suggest that $R \geq H,\ D = 0$ at $\beta \to 0$ and $R = 0,\ D \geq H$ at $\beta \to \infty$ will hold as shown the figure 1 of their work.

In this appendix, we will show that β determines the value of R and D specifically. We also show that $R \simeq H - D$ will hold regardless $\beta = 1$ or not.

In their work, these values of H, D, and D are mathematically defined as:

$$H \equiv -\int \mathrm{d}\boldsymbol{x}\, p^*(\boldsymbol{x}) \log p^*(\boldsymbol{x}), \tag{82}$$

$$D \equiv -\int \mathrm{d}\boldsymbol{x}\, p^*(\boldsymbol{x}) \int \mathrm{d}\boldsymbol{z}\, e(\boldsymbol{z}|\boldsymbol{x}) \log d(\boldsymbol{x}|\boldsymbol{z}), \tag{83}$$

$$R \equiv \int \mathrm{d}\boldsymbol{x}\, p^*(\boldsymbol{x}) \int \mathrm{d}\boldsymbol{z}\, e(\boldsymbol{z}|\boldsymbol{x}) \log \frac{e(\boldsymbol{z}|\boldsymbol{x})}{m(\boldsymbol{z})}. \tag{84}$$

Here, $p^*(\boldsymbol{x})$ is a true PDF of $\boldsymbol{x}$, $e(\boldsymbol{z}|\boldsymbol{x})$ is a stochastic encoder, $e(\boldsymbol{z}|\boldsymbol{x})$ is a decoder, and $m(\boldsymbol{z})$ is a marginal probability of $\boldsymbol{z}$.

Our work allows a rough estimation of Eqs. 82-84 with β by introducing the implicit isometric variable $\boldsymbol{y}$ as explained in our work.

Using isometric variable $\boldsymbol{y}$ and the relation $\mathrm{d}\boldsymbol{z}\, e(\boldsymbol{z}|\boldsymbol{x}) = \mathrm{d}\boldsymbol{y}\, e(\boldsymbol{y}|\boldsymbol{x})$, Eq. 83 can be rewritten as:

$$D = -\int \mathrm{d}\boldsymbol{x}\, p^*(\boldsymbol{x}) \int \mathrm{d}\boldsymbol{y}\, e(\boldsymbol{y}|\boldsymbol{x}) \log d(\boldsymbol{x}|\boldsymbol{y}). \tag{85}$$

Let $\boldsymbol{\mu}_y$ be the implicit isometric latent variable corresponding to the mean of encoder output $\boldsymbol{\mu}_{(\boldsymbol{x})}$. As discussed in section 4.1, $e(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_{\boldsymbol{y}}, (\beta/2)\boldsymbol{I}_n)$ will hold. Because of isometricity, the value of $d(\boldsymbol{x}|\boldsymbol{y})$ will be also close to $e(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_{\boldsymbol{y}}, (\beta/2)\boldsymbol{I}_n)$. Though $d(\boldsymbol{x}|\boldsymbol{z})$ must depend on $e(\boldsymbol{z}|\boldsymbol{x})$, this important point has not been discussed well in this work. By using the implicit isometric variable, we can connect both theoretically. Thus, D can be estimated as:

$$\begin{aligned} D &\simeq \int \mathrm{d}\boldsymbol{x}\, p^*(\boldsymbol{x}) \int \mathrm{d}\boldsymbol{y}\, \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_y, (\beta/2)\boldsymbol{I}_n) \log \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_{\boldsymbol{y}}, (\beta/2)\boldsymbol{I}_n) \\ &\simeq \int \mathrm{d}\boldsymbol{x}\, p^*(\boldsymbol{x}) \left(\frac{n}{2} \log(\beta\pi e)\right) \\ &= \frac{n}{2} \log(\beta\pi e). \end{aligned} \tag{86}$$

Second, R is examined. $m(\boldsymbol{y})$ is a marginal probability of $\boldsymbol{y}$. Using the relation $\mathrm{d}\boldsymbol{z}\, e(\boldsymbol{z}|\boldsymbol{x}) = \mathrm{d}\boldsymbol{y}\, e(\boldsymbol{y}|\boldsymbol{x})$ and $e(\boldsymbol{z}|\boldsymbol{x})/m(\boldsymbol{z}) = (e(\boldsymbol{y}|\boldsymbol{x})(\mathrm{d}\boldsymbol{y}/\mathrm{d}\boldsymbol{z}))/(m(\boldsymbol{y})(\mathrm{d}\boldsymbol{y}/\mathrm{d}\boldsymbol{z})) = e(\boldsymbol{y}|\boldsymbol{x})/m(\boldsymbol{y})$, Eq. 84 can be rewritten as:

$$R \simeq \int \mathrm{d}\boldsymbol{x}\, p^*(\boldsymbol{x}) \int \mathrm{d}\boldsymbol{y}\, e(\boldsymbol{y}|\boldsymbol{x}) \log \frac{e(\boldsymbol{y}|\boldsymbol{x})}{m(\boldsymbol{y})}. \tag{87}$$

Because of isometricity, $e(\boldsymbol{y}|\boldsymbol{x}) \simeq p(\hat{\boldsymbol{x}}|\boldsymbol{x}) \simeq \mathcal{N}(\hat{\boldsymbol{x}}; \boldsymbol{x}, (\beta/2)\boldsymbol{I}_m)$ will approximately hold where $\hat{\boldsymbol{x}}$ denotes a decoder output. Thus $m(\boldsymbol{y})$ can be approximated by:

$$m(\boldsymbol{y}) \simeq \int \mathrm{d}\boldsymbol{x}\, p^*(\boldsymbol{x}) e(\boldsymbol{y}|\boldsymbol{x}) \simeq \int \mathrm{d}\boldsymbol{x}\, p^*(\boldsymbol{x})\, \mathcal{N}(\hat{\boldsymbol{x}}; \boldsymbol{x}, (\beta/2)\boldsymbol{I}_m) \tag{88}$$

Here, if $\beta/2$, i.e., added noise, is small enough compared to the variance of $\boldsymbol{x}$, a normal distribution function term in this equation will act like a delta function. Thus $m(\boldsymbol{y})$ can be approximated as:

$$m(\boldsymbol{y}) \simeq \int \mathrm{d}\acute{\boldsymbol{x}}\, p^*(\acute{\boldsymbol{x}})\, \delta(\acute{\boldsymbol{x}} - \boldsymbol{x}) \simeq p^*(\boldsymbol{x}). \tag{89}$$

In the similar way, the following approximation will also hold.

$$\int \mathrm{d}\boldsymbol{y}\; e(\boldsymbol{y}|\boldsymbol{x}) \log m(\boldsymbol{y}) \simeq \int \mathrm{d}\boldsymbol{y}\; e(\boldsymbol{y}|\boldsymbol{x}) \log p^*(\boldsymbol{x}) \simeq \int \mathrm{d}\acute{\boldsymbol{x}}\; \delta(\acute{\boldsymbol{x}} - \boldsymbol{x})\;\; \log\;\; p^*(\acute{\boldsymbol{x}}) \simeq \log p^*(\boldsymbol{x}). \tag{90}$$

By using these approximation and applying Eqs. 85-86, R in Eq. 84 can be approximated as:

$$\begin{aligned} R &\simeq \int \mathrm{d}\boldsymbol{x}\; p^*(\boldsymbol{x}) \int \mathrm{d}\boldsymbol{y}\; e(\boldsymbol{y}|\boldsymbol{x}) \log \frac{e(\boldsymbol{y}|\boldsymbol{x})}{p^*(\boldsymbol{x})} \\ &\simeq -\int \mathrm{d}\boldsymbol{x}\; p^*(\boldsymbol{x})\;\; \log p^*(\boldsymbol{x}) - \left(-\int \mathrm{d}\boldsymbol{x}\; p^*(\boldsymbol{x}) \int \mathrm{d}\boldsymbol{y}\; e(\boldsymbol{y}|\boldsymbol{x}) \log e(\boldsymbol{y}|\boldsymbol{x}) \right) \\ &\simeq H - \frac{n}{2} \log(\beta \pi e) \\ &\simeq H - D. \end{aligned} \tag{91}$$

As discussed above, R and D can be specifically derived from β. In addition, Shannon lower bound discussed in Alemi et al. (2018) can be roughly verified in the optimized VAE with clearer notations using β.

From the discussion above, we presume Alemi et al. (2018) might wrongly treat D in their work. They suggest that VAE with $\beta = 1$ is sensitive (unstable) because D and R can be arbitrary value on the line $R = H - \beta D = H - D$; however, our work as well as Tishby et al. (1999) (appendix B.2) and Dai & Wipf (2019)(appendix B.5) show that the differential entropy of the distortion and rate, i.e., D and R, are specifically determined by β after optimization, and $R = H - D$ will hold for any β regardless $\beta = 1$ or not. Alemi et al. (2018) also suggest D should satisfy $D \geq 0$ because D is a distortion; however, we suggest D should be treated as a differential entropy and can be less than 0 because $\boldsymbol{x}$ is once handled as a continuous signal with a stochastic process in Eqs. 82-84. Here, $D \simeq (n/2)\log(\beta\pi e)$ can be $-\infty$ if $\beta \to 0$, as also shown in Dai & Wipf (2019). Thus, upper bound of R at $\beta \to 0$ is not H, but $R = H - (-\infty) = \infty$, as shown in RD theory for a continuous signal. Huang et al. (2020) show this property experimentally in their figures 4-8 such that R seems to diverge if MSE is close to 0.

B.5. Relation to Dai et al. (2018) and Dai & Wipf (2019)

Our work is consistent with Dai et al. (2018) and Dai & Wipf (2019).

Dai et al. (2018) analyses VAE by assuming a linear model. As a result, the estimated posterior is constant. If the distribution of the manifold is the Gaussian, our work and Dai et al. (2018) give a similar result with constant posterior variances. For non-Gaussian data, however, the quantitative analysis such as probability estimation is intractable using their linear model. Our work reveals that the posterior variance gives a scaling factor between $\boldsymbol{z}$ in VAE and $\boldsymbol{y}$ in the isometric space when VAE is ideally trained with rich parameters. This is validated by Figures 3c and 3d, where the estimation of the posterior variance at each data point is a key.

Next, the relation to Dai & Wipf (2019) is discussed. They analyse a behavior of VAE when ideally trained. For example, the theorem 5 in their work shows that $D \to (d/2)\log\gamma + O(1)$ and $R \to -(\hat{\gamma}/2)\log\gamma + O(1)$ hold if $\gamma \to +0$, where γ, d, and $\hat{\gamma}$ denote a variance of $d(\boldsymbol{x}|\boldsymbol{z})$, data dimension, and latent dimension, respectively. By setting $\gamma = \beta/2$ and $d = \hat{\gamma} = n$, this theorem is consistent with R and D derived in Eq. 86 and Eq. 91.

B.6. Relation to Rate-distortion theory (Berger, 1971) and transform coding (Goyal, 2001; Pearlman & Said, 2011)

RD theory (Berger, 1971) formulated the optimal transform coding (Goyal, 2001; Pearlman & Said, 2011) for the Gaussian source with square error metric as follows. Let $\boldsymbol{x} \in \mathbb{R}^m$ be a point in a dataset. First, the data are transformed deterministically with the orthonormal transform (orthogonal and unit norm) such as Karhunen-Loève transform (KLT) (Rao & Yip, 2000). Note that the basis of KLT is equivalent to a PCA basis. Let $\boldsymbol{z} \in \mathbb{R}^m$ be a point transformed from $\boldsymbol{x}$. Then, $\boldsymbol{z}$ is entropy-coded by allowing equivalent stochastic distortion (or posterior with constant variance) in each dimension. A lower bound of a rate R at a distortion D is denoted by $R(D)$. The derivation of $R(D)$ is as follows. Let z_j be the j-th dimensional component of $\boldsymbol{z}$ and ${\sigma_{zj}}^2$ be the variance of z_j in a dataset. It is noted that ${\sigma_{zj}}^2$ is the equivalent to eigenvalues of PCA for the dataset. Let d be a distortion equally allowed in each dimensional channel. At the optimal condition, the

distortion $D_{\rm opt}$ and rate $R_{\rm opt}$ on the curve $R(D)$ is calculated as a function of d:

$$\begin{aligned} R_{\rm opt} &= \frac{1}{2}\sum_{j=1}^{m}\max(\log({\sigma_{zj}}^2/d), 0), \\ D_{\rm opt} &= \sum_{j=1}^{m}\min(d,\ {\sigma_{zj}}^2). \end{aligned} \tag{92}$$

The simplest way to allow equivalent distortion is to use a uniform quantization (Goyal, 2001). Let T be a quantization step, and $\mathrm{round}(\cdot)$ be a round function. Quantized value $\hat{z}_j$ is derived as kT, where $k = \mathrm{round}(z_j/T)$. Then, d is approximated by $T^2/12$ as explained in Appendix G.1.

To practically achieve the best RD trade-off in image compression, rate-distortion optimization (RDO) has also been widely used (Sullivan & Wiegand, 1998). In RDO, the best trade-off is achieved by finding a encoding parameter that minimizes the cost L at given Lagrange parameter λ as:

$$L = D + \lambda R. \tag{93}$$

This equation is equivalent to VAE when $\lambda = \beta^{-1}$.

We show the optimum condition of VAE shown in Eq. 62 and 65 can be mapped to the optimum condition of transform coding (Goyal, 2001) as shown in Eq. 92. First, the derivation of Eq. 92 is explained by solving the optimal distortion assignment to each dimension. In the transform coding for m dimensional the Gaussian data, an input data $\boldsymbol{x}$ is transformed to $\boldsymbol{z}$ using an orthonormal transform such as KLT/DCT. Then each dimensional component z_j is encoded with allowing distortion d_j. Let D be a target distortion satisfying $D = \sum_{j=1}^{m} d_j$. Next, σ_{zj}^2 denotes a variance of each dimensional component z_j for the input dataset. Then, a rate R can be derived as $\sum_{j=1}^{m}\frac{1}{2}\log(\sigma_{zj}^2/d_j)$. By introducing a Lagrange parameter λ and minimizing a rate-distortion optimization cost $L = D + \lambda R$, the optimum condition is derived as:

$$\lambda_{\rm opt} = 2D/m, \quad d_j = D/m = \lambda_{\rm opt}/2. \tag{94}$$

This result is consistent with Eq. 62 and 65 by setting $\beta = \lambda_{\rm opt} = 2D/m$. This implies that $L_{\rm G} = D_{\rm G} + \beta R_{\rm G}$ is a rate-distortion optimization (RDO) cost of transform coding when $\boldsymbol{x}$ is deterministically transformed to $\boldsymbol{y}$ in the implicit isometric space and stochastically encoded with a distortion $\beta/2$.

C. Details of the networks and training conditions in the experiments

This appendix explains the networks and training conditions in Section 5.

C.1. Toy data set

This appendix explains the details of the networks and training conditions in the experiment of the toy data set in Section 5.1.

Network configurations:
FC(i, o, f) denotes a FC layer with input dimension i, output dimension o, and activate function f.

The encoder network is composed of FC(16, 128, tanh)-FC(128, 64, tahh)-FC(64, 3, linear)×2 (for μ and σ). The decoder network is composed of FC(3, 64, tanh)-FC(64, 128, tahh)-FC(128, 16, linear).

Training conditions:
The reconstruction loss $D(\cdot,\cdot)$ is derived such that the loss per input dimension is calculated and all of the losses are averaged by the input dimension $m = 16$. The KL divergence is derived as a summation of $D_{{\rm KL}(j)}(\cdot)$ as explained in Eq. 2.

In our code, we use essentially the same, but a constant factor scaled loss objective from the original β-VAE form $L_{\boldsymbol{x}} = D(\cdot,\cdot) + \beta D_{{\rm KL}(j)}(\cdot)$ in Eq. 1, such as:

$$L_{\boldsymbol{x}} = \lambda\, D(\cdot,\cdot) + D_{{\rm KL}(j)}(\cdot). \tag{95}$$

Equation 95 is essentially equivalent to $L = D(\cdot,\cdot) + \beta D_{{\rm KL}(j)}(\cdot)$, multiplying a constant $\lambda = \beta^{-1}$ to the original form. The reason why we use this form is as follows. Let $\mathrm{ELBO}_{\rm true}$ be the true ELBO in the sense of log-likelihood, such as $E[\log p(\boldsymbol{x})]$. As shown in Eq. 57, the minimum of the loss objective in the original β-VAE form is likely to be a

$-\beta\mathrm{ELBO}_{\mathrm{true}} + \mathrm{Constant}$. If we use Eq. 95, the minimum of the loss objective will be $-\mathrm{ELBO}_{\mathrm{true}} + \mathrm{Constant}$, which seems more natural form of ELBO. Thus, Eq. 95 allows estimating a data probability from $L_{\boldsymbol{x}}$ in Eqs. 19 and 21, without scaling $L_{\boldsymbol{x}}$ by $1/\beta$.

Then the network is trained with $\lambda = \beta^{-1} = 100$ using 500 epochs with a batch size of 128. Here, Adam optimizer is used with the learning rate of 1e-3. We use a PC with CPU Inter(R) Xeon(R) CPU E3-1280v5@3.70GHz, 32GB memory equipped with NVIDIA GeForce GTX 1080. The simulation time for each trial is about 20 minutes, including the statistics evaluation codes.

C.2. CelebA data set

This appendix explains the details of the networks and training conditions in the experiment of the toy data set in Section 5.2.

Network configurations:
CNN(w, h, s, c, f) denotes a CNN layer with kernel size (w, h), stride size s, dimension c, and activate function f. GDN and IGDN [†] are activation functions designed for image compression (Ballé et al., 2016). This activation function is effective and popular in deep image compression studies.

The encoder network is composed of CNN(9, 9, 2, 64, GDN) - CNN(5, 5, 2, 64, GDN) - CNN(5, 5, 2, 64, GDN) - CNN(5, 5, 2, 64, GDN) - FC(1024, 1024, softplus) - FC(1024, 32, None)$\times$2 (for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$) in encoder.

The decoder network is composed of FC(32, 1024, softplus) - FC(1024, 1024, softplus) - CNN(5, 5, 2, 64, IGDN) - CNN(5, 5, 2, 64, IGDN) - CNN(5, 5, 2, 64, IGDN)-CNN(9, 9, 2, 3, IGDN).

Training conditions:
In this experiment, SSIM explained in Appendix G.2 is used as a reconstruction loss. The reconstruction loss $D(\cdot, \cdot)$ is derived as follows. Let SSIM be a SSIM calculated from two input images. As explained in Appendix G.2, SSIM is measured for a whole image, and its range is between 0 and 1. If the quality is high, SSIM value becomes close to 1. Then $1 - \mathrm{SSIM}$ is set to $D(\cdot, \cdot)$.

We also use the loss form as in Equation 95 in our code. In the case of the decomposed loss, the loss function $L_{\boldsymbol{x}}$ is set to $\lambda(D(\boldsymbol{x}, \breve{\boldsymbol{x}}) + D(\breve{\boldsymbol{x}}, \hat{\boldsymbol{x}})) + D_{\mathrm{KL}}(\cdot)$ in our code. Then, the network is trained with $\lambda = \beta^{-1} = 1,000$ using a batch size of 64 for 300,000 iterations. Here, Adam optimizer is used with the learning rate of 1e-3.

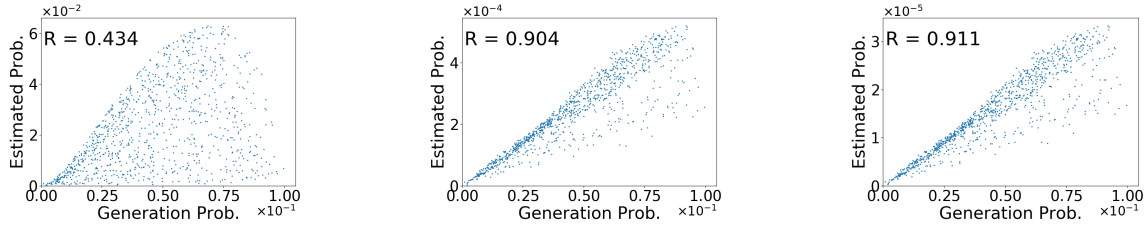
We use a PC with CPU Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz, 12GB memory equipped with NVIDIA GeForce GTX 1080. The simulation time for each trial is about 180 minutes, including the statistics evaluation codes.

[†]Google provides a code in the official Tensorflow library (https://github.com/tensorflow/compression)

D. Additional results in the toy datasets

D.1. Scattering plots for the square error loss in Section

Figure 10a shows the plots of $p(\boldsymbol{x})$ and estimated probabilities for the square error coding loss in Section 5.1, where the scale factor $a_{\boldsymbol{x}}$ in Eq. 21 is 1. Thus, both $\exp(-L_x/\beta)$ and $p(\boldsymbol{\mu}_{(\boldsymbol{x})})\prod_j \sigma_{j(\boldsymbol{x})}$ show a high correlation, allowing easy estimation of the data probability in the input space. In contrast, $p(\boldsymbol{\mu}_{(\boldsymbol{x})})$ still shows a low correlation. These results are consistent with our theory.



(a) $p(\boldsymbol{\mu}_{(\boldsymbol{x})})$ (b) $\exp(-L_x/\beta)$ (c) $p(\boldsymbol{\mu}_{(\boldsymbol{x})})\prod_j \sigma_{j(\boldsymbol{x})}$

Figure 10. Plots of the data generation probability (x-axis) versus estimated probabilities (y-axes) for the square error loss. y-axes are (a) $p(\boldsymbol{\mu}_{(\boldsymbol{x})})$, (b) $\exp(-L_x/\beta)$, and (c) $p(\boldsymbol{\mu}_{(\boldsymbol{x})})\prod_j \sigma_{j(\boldsymbol{x})}$.

D.2. Ablation study using 3 toy datasets, 3 coding losses, and 10 β parameters.

In this appendix, we explain the ablation study for the toy datasets. We introduce three toy datasets and three coding losses including those used in Section 5.1. We also change $\beta^{-1} = \lambda$ from 1 to $1,000$ in training. The details of the experimental conditions are shown as follows.

Datasets: First, we call the toy dataset used in Section 5.1 the Mix dataset in order to distinguish three datasets. The second dataset is generated such that three dimensional variables s_1, s_2, and s_3 are sampled in accordance with the distributions $p(s_1)$, $p(s_2)$, and $p(s_3)$ in Figure 11. The variances of the variables are the same as those of the Mix dataset, i.e., 1/6, 2/3, and 8/3, respectively. We call this the Ramp dataset. Because the PDF shape of this dataset is quite different from the prior $\mathcal{N}(\boldsymbol{z}; 0, I_3)$, the fitting will be the most difficult among the three. The third dataset is generated such that three dimensional variables s_1, s_2, and s_3 are sampled in accordance with the normal distributions $\mathcal{N}(s_1; 0, 1/6)$, $\mathcal{N}(s_2; 0, 2/3)$, and $\mathcal{N}(s_3; 0, 8/3)$, respectively. We call this the Norm dataset. The fitting will be the easiest, because both the prior and input have the normal distributions, and the posterior standard deviation, given by the PDF ratio at the same CDF, can be a constant.

Coding losses: Two of the three coding losses is the square error loss and the downward-convex loss described in Section 5.1. The third coding loss is a upward-convex loss which we design as Eq. 96 such that the scale factor $a_{\boldsymbol{x}}$ becomes the reciprocal of the scale factor in Eq. 24:

$$D(\boldsymbol{x}, \hat{\boldsymbol{x}}) = a_{\boldsymbol{x}} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2, \quad \text{where } a_{\boldsymbol{x}} = (2/3 + 2\,\|\boldsymbol{x}\|_2^2/21)^{-1} \text{ and } \boldsymbol{G}_x = a_{\boldsymbol{x}} \boldsymbol{I}_m. \tag{96}$$

Figure 12 shows the scale factors $a_{\boldsymbol{x}}$ in Eqs. 24 and 96, where s_1 in $\boldsymbol{x} = (s_1, 0, 0)$ moves within ± 5.

Parameters: As explained in Appendix C.1, $\lambda = 1/\beta$ is used as a hyper parameter. Specifically, $\lambda = 1$, 2, 5, 10, 20, 50, 100, 200, 500, and $1,000$ are used.

Figures 13 - 21 show the property measurements for all combinations of the datasets and coding losses, with changing λ. In each Figure, the estimated norms of the implicit transform are shown in the figure (a), the ratios of the estimated variances are shown in the figure (b), and the correlation coefficients between $p(\boldsymbol{x})$ and estimated data probabilities are shown in the figure (c), respectively.

First, the estimated norm of the implicit transform in the figures (a) is discussed. In all conditions, the norms are close to 1 as described in Eq. 23 in the λ range 50 to 1000. These results show consistency with our theoretical analysis, supporting the existence of the implicit orthonormal transform. The values in the Norm dataset are the closest to 1, and those in the Ramp dataset are the most different, which seems consistent with the difficulty of the fitting.

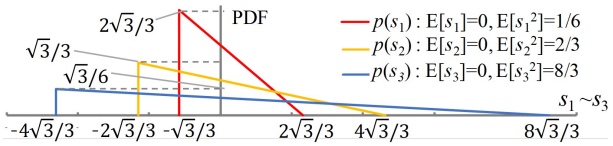


Figure 11. PDFs of three variables to generate a Ramp dataset.

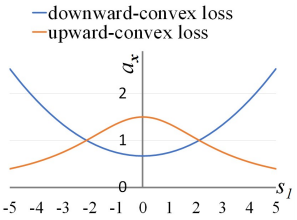


Figure 12. Scale factor $a_{\boldsymbol{x}}$ for the downward-convex loss and upward-convex loss.

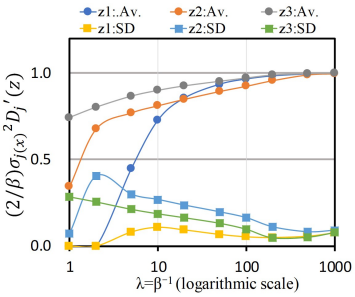
Second, the ratio of the estimated variances is discussed. In the figures (b), $\mathrm{Var}(z_j)$ denotes the estimated variance, given by the average of $\sigma_{j(\boldsymbol{x})}^{-2}$. Then, $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$ and $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$ are plotted. In all conditions, the ratios of $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$ and $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$ are close to the variance ratios of the input variables, i.e., 4 and 16, in the λ range 5 to 500. Figure 22 shows the detailed comparison of the ratio for the three datasets and three coding losses at $\lambda = 100$. In most cases, the estimated variances in the downward-convex loss are the smallest, and those in the upward-convex loss are the largest, which is more distinct for $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$. This can be explained as follows. When using the downward-convex loss, the space region with a large norm is thought of as shrinking in the inner product space, as described in Section 5.1. This will make the variance smaller. In contrast, when using the upward-convex loss, the space region with a large norm is thought of as expanding in the inner product space, making the variance larger. Here, the dependency of the losses on the ratio changes is less in the Norm dataset. The possible reason is that data in the normal distribution concentrate around the center, having less effect on the loss scale factor in the downward-convex loss and upward-convex loss.

Third, the correlation coefficients between $p(\boldsymbol{x})$ and the estimated data probabilities in the figures (c) are discussed. In the Mix dataset and Ramp dataset, the correlation coefficients are around 0.9 in the λ range from 20 to 200 when the estimated probabilities ${a_{\boldsymbol{x}}}^{n/2} p(\boldsymbol{\mu}_{(\boldsymbol{x})}) \prod_{j=1}^{n} \sigma_{j(\boldsymbol{x})}$ and ${a_{\boldsymbol{x}}}^{n/2} \exp(-(1/\beta)L_{\boldsymbol{x}})$ in Eq. 21 are used. When using $p(\boldsymbol{\mu}_{(\boldsymbol{x})}) \prod_{j=1}^{n} \sigma_{j(\boldsymbol{x})}$ and $\exp(-(1/\beta)L_{\boldsymbol{x}})$ in the downward-convex loss and upward-convex loss, the correlation coefficients become worse. In addition, when using the prior probability $p(\boldsymbol{\mu}_{(\boldsymbol{x})})$, the correlation coefficients always show the worst. In the Norm dataset, the correlation coefficients are close to 1.0 in the wider range of λ when using the estimated distribution in Eq. 21. When using $p(\boldsymbol{\mu}_{(\boldsymbol{x})}) \prod_{j=1}^{n} \sigma_{j(\boldsymbol{x})}$ and $\exp(-(1/\beta)L_{\boldsymbol{x}})$ in the downward-convex loss and upward-convex loss, the correlation coefficients also become worse. When using the prior probability $p(\boldsymbol{\mu}_{(\boldsymbol{x})})$, however, the correlation coefficients are close to 1 in contrast to the other two datasets. This can be explained because both the input distribution and the prior distribution are the same normal distribution, allowing the posterior variances almost constant. These results also show consistency with our theoretical analysis.

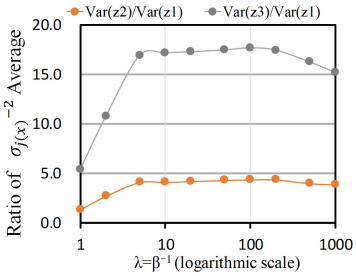
Figure 23 shows the dependency of the coding loss on β for the Mix, Ramp, and Norm dataset using square the error loss. From D_{G} in Eq. 64 and $n = 3$, the theoretical value of coding loss is $\frac{3\beta}{2}$, as also shown in the figure. Unlike Figs. 13-21, x-axis is $\beta = \lambda^{-1}$ to evaluate the linearity. As expected in Theorem 3, the coding losses are close to the theoretical value where $\beta < 0.1$, i.e., $\lambda > 10$.

Figure 24 shows the dependency of the ratio of transform loss to coding loss on β for the Mix, Ramp, and Norm dataset using square the error loss. From Eq. 43, the estimated transform loss is $\sum_{i=1}^{3} (\beta/2)^2 / \mathrm{Var}(s_i) = \frac{63\beta^2}{32}$. Thus the theoretical value is $(\frac{63\beta^2}{32})/(\frac{3\beta}{2}) = \frac{21\beta}{16}$, as is also shown in the figure. x-axis is also $\beta = \lambda^{-1}$ like Figure 23. Considering the correlation coefficient discussed above, the useful range of β seems between 0.005-0.05 (20-200 for λ). In this range, the ratio is less than 0.1, implying the transform loss is almost negligible. As expected in Lemma 3 and appendix A.3, the ratio is close to the theoretical value where $\beta > 0.01$, i.e., $\lambda < 100$. For $\beta < 0.01$, the transform loss is still negligibly small, but the ratio is somewhat off the theoretical value. The reason is presumably that the transform loss is too small to fit the network.

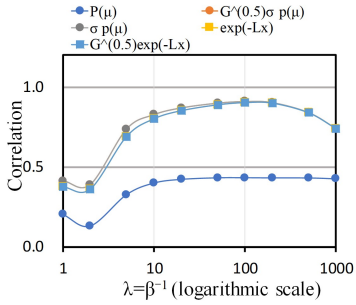
As shown above, this ablation study strongly supports our theoretical analysis in sections 4.



(a) Estimated norm $\frac{2}{\beta}\sigma_{j(\boldsymbol{x})}{}^2 D'_j(\boldsymbol{z})$.

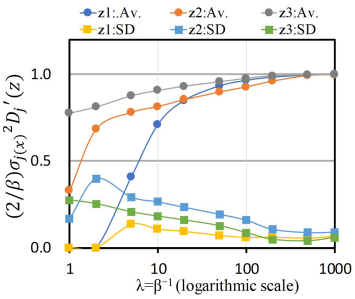


(b) Ratio of the estimated variances $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$ and $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$

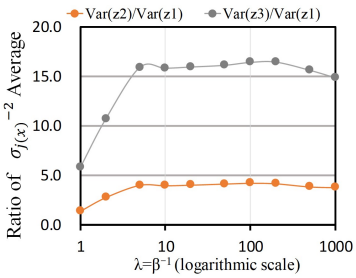


(c) Correlation coefficient of the estimated data probability

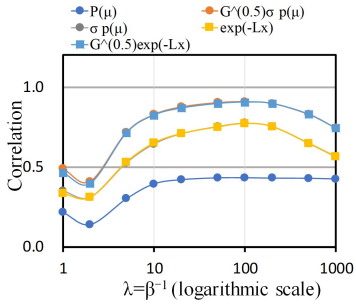
Figure 13. Property measurements of the Mix dataset using the square error loss. λ is changed from 1 to $1,000$. $\mathrm{Var}(z_j)$ denotes the estimated variance, given by the average of $\sigma_{j(\boldsymbol{x})}^{-2}$.



(a) Estimated norm $\frac{2}{\beta}\sigma_{j(\boldsymbol{x})}{}^2 D'_j(\boldsymbol{z})$.

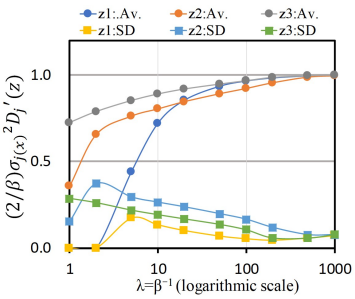


(b) Ratio of the estimated variances $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$ and $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$

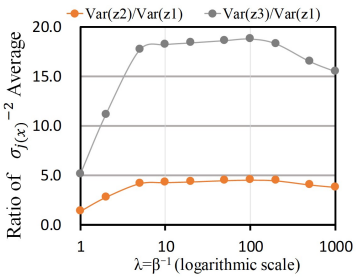


(c) Correlation coefficient of the estimated data probability

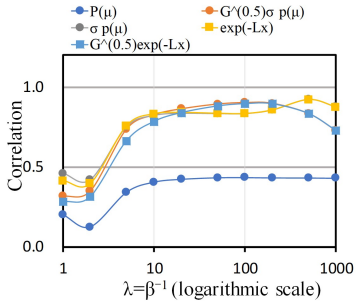
Figure 14. Property measurements of the Mix dataset using the downward-convex loss. λ is changed from 1 to $1,000$. $\mathrm{Var}(z_j)$ denotes the estimated variance, given by the average of $\sigma_{j(\boldsymbol{x})}^{-2}$.



(a) Estimated norm $\frac{2}{\beta}\sigma_{j(\boldsymbol{x})}{}^2 D'_j(\boldsymbol{z})$.

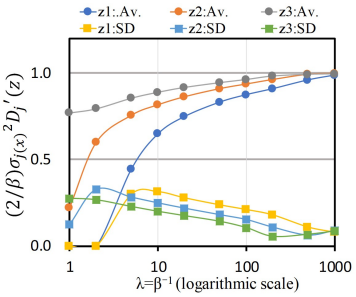


(b) Ratio of the estimated variances $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$ and $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$

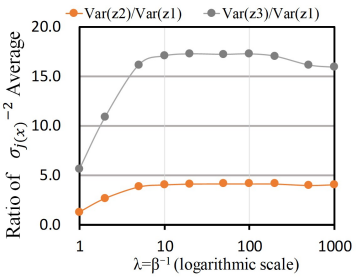


(c) Correlation coefficient of the estimated data probability

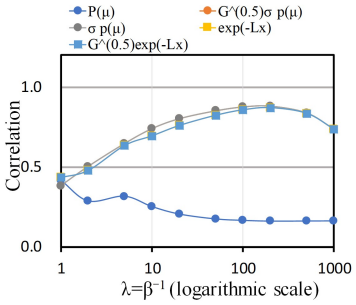
Figure 15. Property measurements of the Mix dataset using the upward-convex loss. λ is changed from 1 to $1,000$. $\mathrm{Var}(z_j)$ denotes the estimated variance, given by the average of $\sigma_{j(\boldsymbol{x})}^{-2}$.



(a) Estimated norm $\frac{2}{\beta}\sigma_{j(\boldsymbol{x})}{}^2 D'_j(\boldsymbol{z})$.

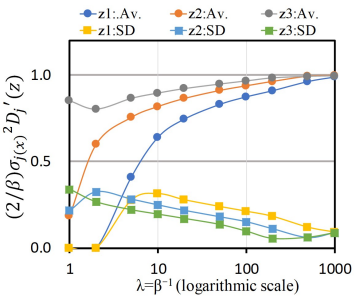


(b) Ratio of the estimated variances $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$ and $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$

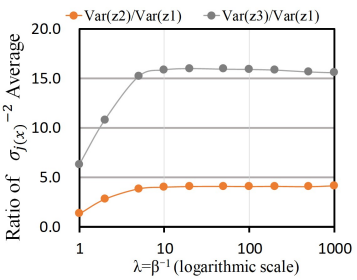


(c) Correlation coefficient of the estimated data probability

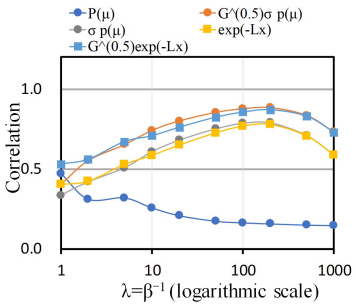
Figure 16. Property measurements of the Ramp dataset using the square error loss. λ is changed from 1 to $1,000$. $\mathrm{Var}(z_j)$ denotes the estimated variance, given by the average of $\sigma_{j(\boldsymbol{x})}^{-2}$.



(a) Estimated norm $\frac{2}{\beta}\sigma_{j(\boldsymbol{x})}{}^2 D'_j(\boldsymbol{z})$.

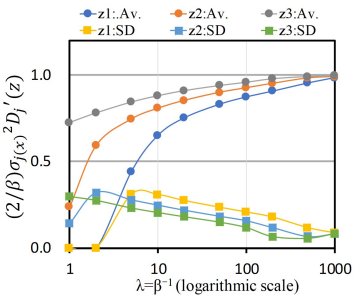


(b) Ratio of the estimated variances $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$ and $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$

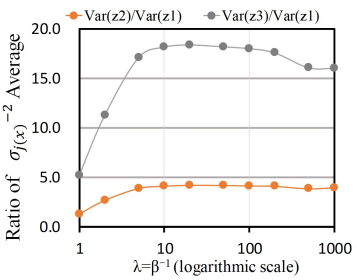


(c) Correlation coefficient of the estimated data probability

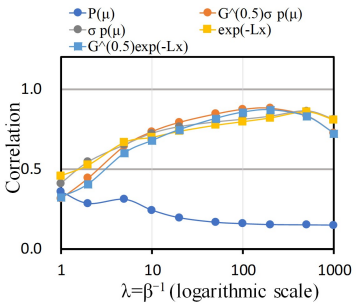
Figure 17. Property measurements of the Ramp dataset using the downward-convex loss. λ is changed from 1 to $1,000$. $\mathrm{Var}(z_j)$ denotes the estimated variance, given by the average of $\sigma_{j(\boldsymbol{x})}^{-2}$.



(a) Estimated norm $\frac{2}{\beta}\sigma_{j(\boldsymbol{x})}{}^2 D'_j(\boldsymbol{z})$.

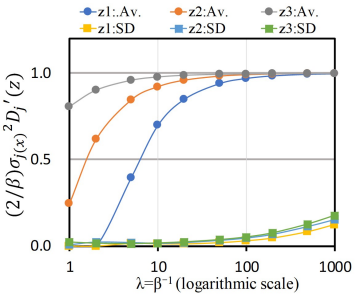


(b) Ratio of the estimated variances $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$ and $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$

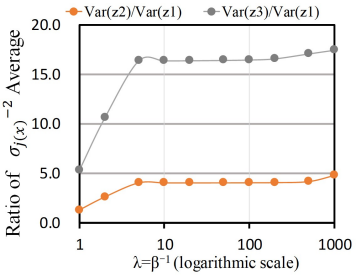


(c) Correlation coefficient of the estimated data probability

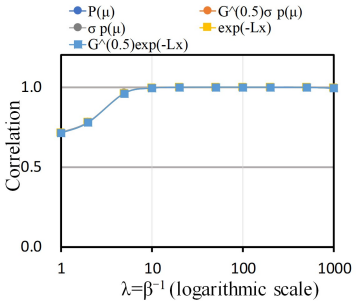
Figure 18. Property measurements of the Ramp dataset using the upward-convex loss. λ is changed from 1 to $1,000$. $\mathrm{Var}(z_j)$ denotes the estimated variance, given by the average of $\sigma_{j(\boldsymbol{x})}^{-2}$.



(a) Estimated norm $\frac{2}{\beta}{\sigma_{j(\boldsymbol{x})}}^2 D'_j(\boldsymbol{z})$.

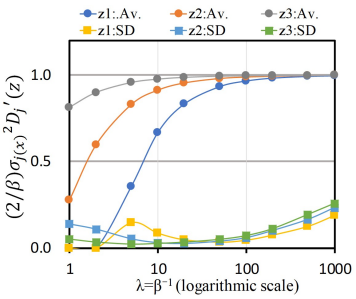


(b) Ratio of the estimated variances $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$ and $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$

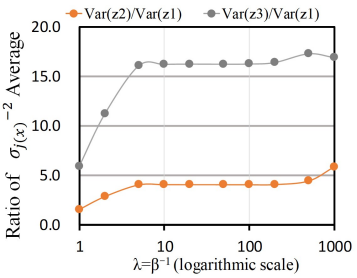


(c) Correlation coefficient of the estimated data probability

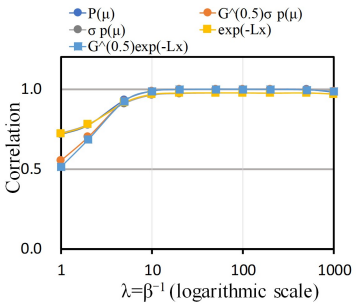
Figure 19. Property measurements of the Norm dataset using the square error loss. λ is changed from 1 to $1,000$. $\mathrm{Var}(z_j)$ denotes the estimated variance, given by the average of $\sigma_{j(\boldsymbol{x})}^{-2}$.



(a) Estimated norm $\frac{2}{\beta}{\sigma_{j(\boldsymbol{x})}}^2 D'_j(\boldsymbol{z})$.

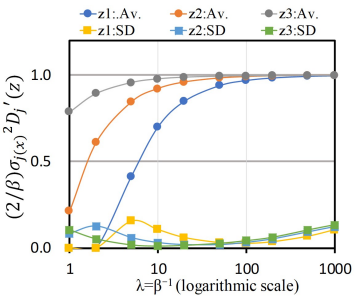


(b) Ratio of the estimated variances $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$ and $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$

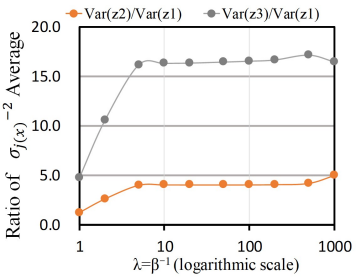


(c) Correlation coefficient of the estimated data probability

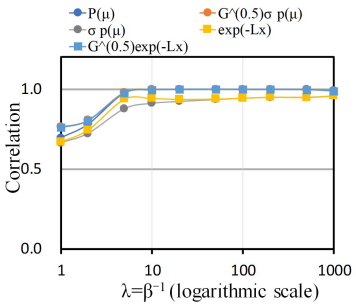
Figure 20. Property measurements of the Norm dataset using the downward-convex loss. λ is changed from 1 to $1,000$. $\mathrm{Var}(z_j)$ denotes the estimated variance, given by the average of $\sigma_{j(\boldsymbol{x})}^{-2}$.



(a) Estimated norm $\frac{2}{\beta}{\sigma_{j(\boldsymbol{x})}}^2 D'_j(\boldsymbol{z})$.



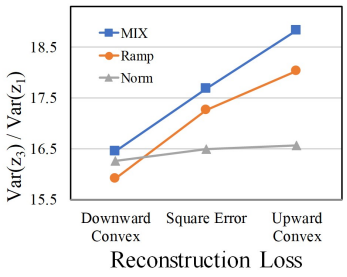
(b) Ratio of the estimated variances $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$ and $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$



(c) Correlation coefficient of the estimated data probability

Figure 21. Property measurements of the Mix dataset using the upward-convex loss. λ is changed from 1 to $1,000$. $\mathrm{Var}(z_j)$ denotes the estimated variance, given by the average of $\sigma_{j(\boldsymbol{x})}^{-2}$.

(a) $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$.



(b) $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$.

Figure 22. Ratio of the estimated variances $\mathrm{Var}(z_3)/\mathrm{Var}(z_1)$ and $\mathrm{Var}(z_2)/\mathrm{Var}(z_1)$ for the three datasets and three coding losses at $\lambda = 100$. $\mathrm{Var}(z_j)$ denotes the estimated variance, given by the average of $\sigma^{-2}_{j(\boldsymbol{x})}$.

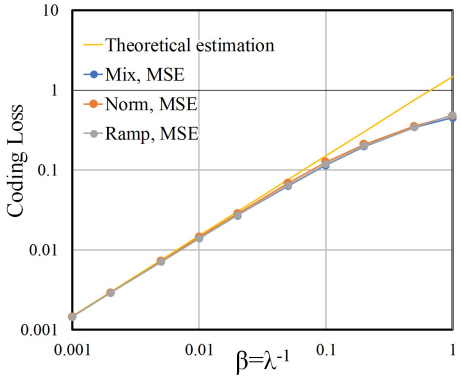


Figure 23. Dependency of Coding Loss on β for Mix, Norm, and Ramp dataset using square loss.

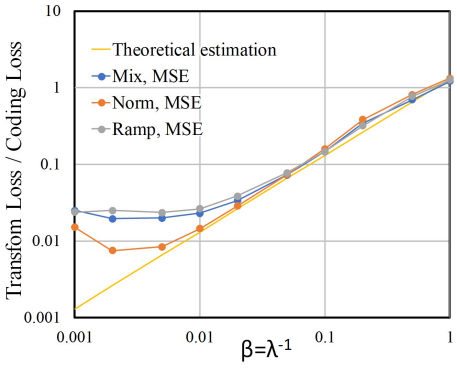


Figure 24. Dependency of Transform loss / Coding Loss Ratio on β for Mix, Norm, and Ramp dataset using square loss.

D.3. Increase of latent dimension

The Table 4 and Figure 25 show the results using Table 1 condition except the latent dimension is increased to 5. For z_1 to z_3, each value is close to Table 1. For z_4 and z_5, $(2/\beta){\sigma_j}^2 {D'}_j$ are almost 0 and the averages of ${\sigma_{j(\boldsymbol{x})}}^{-2}$ are close to 1. In such dimensions, $\mathrm{Var}(y_j) < \beta/2$ and $D_{\mathrm{KL}j}(\cdot) = 0$ will hold as explained in Appendix A.7, E.2, and B.6 (RD theory). Figure 25 describes the plot for ${a_x}^{n/2} \exp(-L_{\boldsymbol{x}}/\beta)$ corresponding to Fig. 3d, also showing almost proportionality.

variable		z_1	z_2	z_3	z_4	z_5
$\frac{2}{\beta}{\sigma_j}^2 D'_j$	Av.	0.963	0.918	0.964	0.000	0.000
	SD	0.053	0.169	0.103	0.000	0.000
${\sigma_{j(\boldsymbol{x})}}^{-2}$	Av.	3.34e1	1.46e2	5.88e2	1.00e0	1.00e0
(Ratio)	Av.	1.0	4.39	17.69	0.03	0.03

Table 4. Property measurements of the toy dataset with 5-dimensional latents trained using the square error loss.

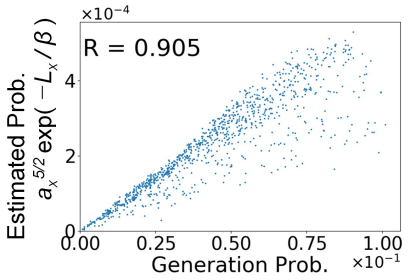


Figure 25. Plots of the data generation probability (x-axis) versus estimated probabilities (y-axes) for the square error loss. The dimension of latents is set to 5.

E. Additional results in CelebA dataset

E.1. Traversed outputs for all the component in the experimental section 5.2

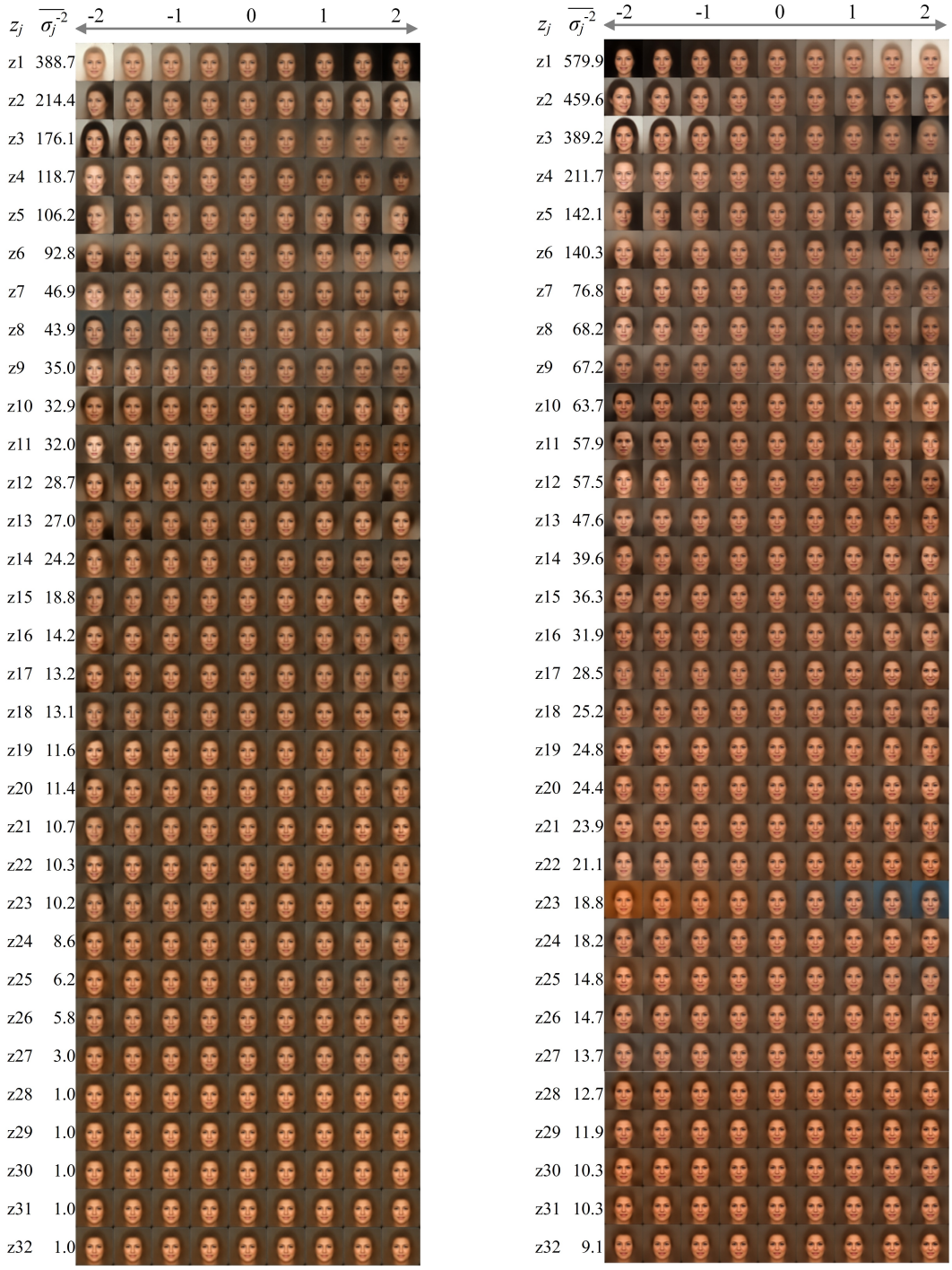
Figure 26 shows decoder outputs for all the components, where each latent variable is traversed from -2 to 2. The estimated variance of each y_j, i.e., $\overline{\sigma_j^{-2}}$, is also shown in these figures. The latent variables z_i are numbered in descending order by the estimated variances. Figure 26a is a result using the conventional loss form, i.e., $L_{\boldsymbol{x}} = D(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \beta D_{\mathrm{KL}}(\cdot)$. The degrees of change seem to descend in accordance with the estimated variances. In the range where j is 1 from 10, the degrees of changes are large. In the range $j > 10$, the degrees of changes becomes gradually smaller. Furthermore, almost no change is observed in the range $j > 27$. As shown in Figure 4, $D_{\mathrm{KL(j)}}(\cdot)$ is close to zero for $j > 27$, meaning no information. Note that the behavior of dimensional components where $D_{\mathrm{KL(j)}}(\cdot) = 0$ is explained in section E.2. Thus, this result is clearly consistent with our theoretical analysis in section 4.3.

Figure 26b is a result using the decomposed loss form, i.e., $L_{\boldsymbol{x}} = D(\boldsymbol{x}, \breve{\boldsymbol{x}}) + D(\breve{\boldsymbol{x}}, \hat{\boldsymbol{x}}) + \beta D_{\mathrm{KL}}(\cdot)$. The degrees of change also seem to descend in accordance with the estimated variances. When looking at the detail, there are still minor changes even $j = 32$. As shown in Figure 5, KL divergences $D_{\mathrm{KL(j)}}(\cdot)$ for all the components are larger than zero. This implies all of the dimensional components have meaningful information. Therefore, we can see a minor change even $j = 32$. Thus, this result is also consistent with our theoretical analysis in Section 4.3.

Another minor difference is sharpness. Although the quantitative comparison is difficult, the decoded images in Figure 26b seems somewhat sharper than those in Figure 26a. A possible reason for this minor difference is as follows. The transform loss $D(\boldsymbol{x}, \breve{\boldsymbol{x}})$ serves to bring the decoded image of $\boldsymbol{\mu}_{(\boldsymbol{x})}$ closer to the input. In the conventional image coding, the orthonormal transform and its inverse transform are used for encoding and decoding, respectively. Therefore, the input and the decoded output are equivalent when not using quantization. If not so, the quality of the decoded image will suffer from the degradation. Considering this analogy, the use of decomposed loss might improve the decoded images for $\boldsymbol{\mu}_{(\boldsymbol{x})}$, encouraging the improvement of the orthonormality of the encoder/decoder in VAE.

E.2. The understanding of latent components where $D_{\mathrm{KL}j}(\cdot) = 0$ in Figure 4

This section explains the behaviors of latent components where $D_{\mathrm{KL}j}(\cdot) = 0$, especially in Fig. 4. First, we explain why the norm becomes 0 when $D_{\mathrm{KL}j}(\cdot) = 0$. The loss in Eq. 13 consists of a norm (multiplied by $\beta/2$) and $\beta D_{\mathrm{KL}j}(\cdot)$ to find the best balance (trade-off) between them. If $D_{\mathrm{KL}j}(\cdot) = 0$, the norm also becomes zero because balancing them is no more needed. Second, we explain the condition where $D_{\mathrm{KL}j}(\cdot) = 0$. Let $\mathrm{Var}(y_j)$ and ${\sigma_{y_j(x)}}^2$ be the variance and posterior variance of j-th implicit isometric component y_j, respectively. Here, ${\sigma_{y_j(x)}}^2$ is $\beta/2$ in our theory. Then the condition where $D_{\mathrm{KL}j}(\cdot) = 0$ is derived as $\mathrm{Var}(y_j) \leq {\sigma_{y_j(x)}}^2$, as shown in Appendix A.7 and B.6 (RD theory). In RD theory, this is corresponding to the case where the signal magnitude is always less than the quantizer size ($\sqrt{\beta/2}$ in β-VAE case) and no information is needed to be encoded. Finally, we explain the reason why the behaviors in Figs. 4 and 5 are different. In Fig. 5 with the decomposed loss, ${\sigma_{y_j(x)}}^2$ is almost $\beta/2$ as the theory expects. In this case, all of $\mathrm{Var}(y_j)$ happen to be greater than ${\sigma_{y_j(x)}}^2$. In Fig. 4 with the conventional loss, however, ${\sigma_{y_j(x)}}^2$ is about 1.83 times greater than $\beta/2$. Note that in both Figs. 4 and 5, $\mathrm{Var}(y_j)$ will be almost the same because of the isometric embedding. Since ${\sigma_{y_j(x)}}^2$ becomes larger, the number of dimensions where $\mathrm{Var}(y_j) \leq {\sigma_{y_j(x)}}^2$ will increases. Accordingly, the dimensions where the norms are zero also increase. Figure 27 shows the CelebA results with smaller β, resulting smaller ${\sigma_{y_j(x)}}^2$. Here, all dimensions have nonzero norms because $\mathrm{Var}(y_j) > {\sigma_{y_j(x)}}^2$ will hold.



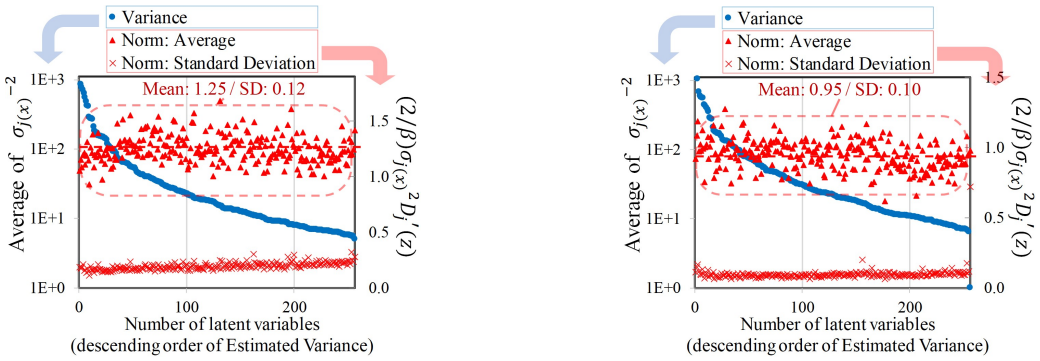
(a) Trained using the conventional loss form.

(b) Trained using the decomposed loss form.

Figure 26. Traversed outputs for all the component, changing z_j from -2 to 2. The latent variables z_j are numbered in descending order by the estimated variance $\overline{\sigma_j^{-2}}$ shown in Figures 4 and 5.

E.3. Additional experimental result with other condition

In this Section, we provide the experimental results with other condition. We use essentially the same condition as described in Appendix C.2, except for the following conditions. The bottleneck size and λ are set to 256 and 10000, respectively. The



(a) Conventional loss form

(b) Decomposed loss form

Figure 27. Graph of ${\sigma_{j(\boldsymbol{x})}}^{-2}$ average and $\frac{2}{\beta}{\sigma_{j(\boldsymbol{x})}}^{2} D_j'(\boldsymbol{z})$ in CelebA dataset. The bottleneck size and λ are set to 256 and 10000, respectively.

encoder network is composed of CNN(9, 9, 2, 64, GDN) - CNN(5, 5, 2, 64, GDN) - CNN(5, 5, 2, 64, GDN) - CNN(5, 5, 2, 64, GDN) - FC(1024, 2048, softplus) - FC(2048, 256, None)$\times$2 (for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$) in encoder. The decoder network is composed of FC(256, 2048, softplus) - FC(2048, 1024, softplus) - CNN(5, 5, 2, 64, IGDN) - CNN(5, 5, 2, 64, IGDN) - CNN(5, 5, 2, 64, IGDN)-CNN(9, 9, 2, 3, IGDN).

Figures 27a and 27b show the averages of ${\sigma_{j(\boldsymbol{x})}}^{-2}$ as well as the average and the standard deviation of $\frac{2}{\beta}{\sigma_{j(\boldsymbol{x})}}^{2} D_j'(\boldsymbol{z})$ in the conventional loss form and the decomposed loss form, respectively. When using the conventional loss form, the mean of $\frac{2}{\beta}{\sigma_{j(\boldsymbol{x})}}^{2} D_j'(\boldsymbol{z})$ is 1.25, which is closer to 1 than the mean 1.83 in Section 5.2. This suggests that the implicit transform is closer to the orthonormal. The possible reason is that a bigger reconstruction error is likely to cause the interference to RD-trade off and a slight violation of the theory, and it might be compensated with a larger lambda. When using the decomposed loss form, the mean of $\frac{2}{\beta}{\sigma_{j(\boldsymbol{x})}}^{2} D_j'(\boldsymbol{z})$ is 0.95, meaning almost unit norm. These results also support that VAE provides the implicit orthonormal transform even if the lambda or bottleneck size is varied.

F. Additional Experimental Result with MNIST dataset

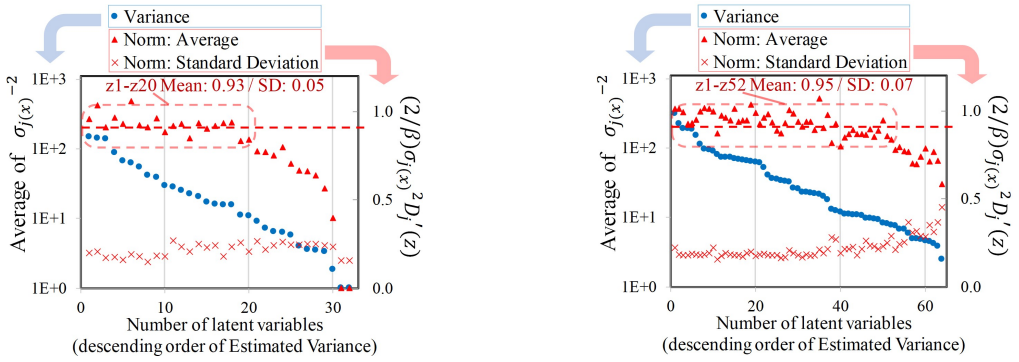
In this Appendix, we provide the experimental result of Section 5.2 with MNIST dataset[‡] consists of binary hand-written digits with a dimension of 768(=28 $\times$ 28). We use standard training split which includes 50,000 data points. For the reconstruction loss, we use the binary cross entropy loss (BCE) for the Bernoulli distribution. We averaged BCE by the number of pixels.

The encoder network is composed of FC(768, 1024, relu) - FC(1024, 1024, relu) - FC(1024, bottleneck size) in encoder. The decoder network is composed of FC(bottleneck size, 1024, relu) - FC(1024, 1024, relu) - FC(1024, 768, sigmoid). The batch size is 256 and the training iteration number is 50,000. In this section, results with two parameters, (bottleneck size=32, λ=2000) and (bottleneck size=64, λ=10000) are provided. Note that since we averaged BCE loss by the number of pixels, β in the conventional β VAE is derived by $768/\lambda$. Then, the model is optimized by Adam optimizer with the learning rate of 1e-3, using the conventional (not decomposed) loss form.

We use a PC with CPU Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz, 12GB memory equipped with NVIDIA GeForce GTX 1080. The simulation time for each trial is about 10 minutes, including the statistics evaluation codes.

Figure 28 shows the averages of ${\sigma_{j(\boldsymbol{x})}}^{-2}$ as well as the average and the standard deviation of $\frac{2}{\beta}{\sigma_{j(\boldsymbol{x})}}^{2} D_j'(\boldsymbol{z})$. In both conditions, the means of $\frac{2}{\beta}{\sigma_{j(\boldsymbol{x})}}^{2} D_j'(\boldsymbol{z})$ averages are also close to 1 except in the dimensions where ${\sigma_{j(\boldsymbol{x})}}^{-2}$ is less than 10. These results suggest the theoretical property still holds when using the BCE loss. In the dimensions where ${\sigma_{j(\boldsymbol{x})}}^{-2}$ is less than 10, the $\frac{2}{\beta}{\sigma_{j(\boldsymbol{x})}}^{2} D_j'(\boldsymbol{z})$ is somewhat lower than 1. The possible reason is that $D_{\mathrm{KL(j)}}(\cdot)$ in such dimension is 0 for some inputs and is larger than 0 in other inputs. The understanding of the transition region needs further study.

[‡]http://yann.lecun.com/exdb/mnist/



(a) bottle neck=32, λ=2000 (b) bottle neck=64, λ=10000

Figure 28. Graph of ${\sigma_{j(\boldsymbol{x})}}^{-2}$ average and $\frac{2}{\beta}{\sigma_{j(\boldsymbol{x})}}^{2}D_j'(\boldsymbol{z})$ in MNIST dataset.

G. Derivation/Explanation in RDO-related equation expansions

G.1. Approximation of distortion in uniform quantization

Let T be a quantization step. Quantized values $\hat{z_j}$ is derived as $k\,T$, where $k = \mathrm{round}(z_j/T)$. Then d, the distortion per channel, is approximated by

$$\begin{aligned} d &= \sum_k \int_{(k-1/2)T}^{(k+1/2)T} p(z_j)(z_j - k\,T)^2\,\mathrm{d}z_j \;\simeq\; \sum_k T\,p(k\,T)\int_{(k-1/2)T}^{(k+1/2)T}\frac{1}{T}(z_j - k\,T)^2\,\mathrm{d}z_j \\ &= \frac{T^2}{12}\sum_k T\,p(k\,T) \;\simeq\; \frac{T^2}{12}. \end{aligned} \tag{97}$$

Here, $\sum_k T\,p(k\,T) \simeq \int_{-\infty}^{\infty} p(z_j)\mathrm{d}z_j = 1$ is used. The distortion for the given quantized value is also estimated as $T^2/12$, because this value is approximated by $\int_{(k-1/2)T}^{(k+1/2)T}\frac{1}{T}(z_j - k\,T)^2\,\mathrm{d}z_j$.

G.2. Approximation of reconstruction loss as a quadratic form.

In this appendix, the approximations of the reconstruction losses as a quadratic form ${}^t\delta\boldsymbol{x}\,\boldsymbol{G}_{\boldsymbol{x}}\delta\boldsymbol{x} + C_{\boldsymbol{x}}$ are explained for the sum of square error (SSE), binary cross entropy (BCE) and Structural Similarity (SSIM). Here, we have borrowed the derivation of BCE and SSIM from Kato et al. (2020), and add some explanation and clarification to them for convenience. We also describe the log-likelihood of the Gaussian distribution.

Let $\hat{\boldsymbol{x}}$ and $\hat{x_i}$ be decoded sample $\mathrm{Dec}_\theta(\boldsymbol{z})$ and its i-th dimensional component respectively. $\delta\boldsymbol{x}$ and δx_i denote $\boldsymbol{x} - \hat{\boldsymbol{x}}$ and $x_i - \hat{x_i}$, respectively. It is also assumed that $\delta\boldsymbol{x}$ and δx_i are infinitesimal. The details of the approximations are described as follows.

Sum square error:
In the case of sum square error, $\boldsymbol{G}_{\boldsymbol{x}}$ is equal to $\boldsymbol{I}_m$. This can be derived as:

$$\sum_{i=1}^{m}(x_i - \hat{x_i})^2 = \sum_{i=1}^{m}\delta x_i^2 = {}^t\delta\boldsymbol{x}\boldsymbol{I}_m\delta\boldsymbol{x}. \tag{98}$$

Binary cross entropy:
Binary cross entropy is a log likelihood of the Bernoulli distribution. The Bernoulli distribution is described as:

$$p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \prod_{i=1}^{m}\hat{x_i}^{x_i}\,(1-\hat{x_i})^{(1-x_i)}. \tag{99}$$

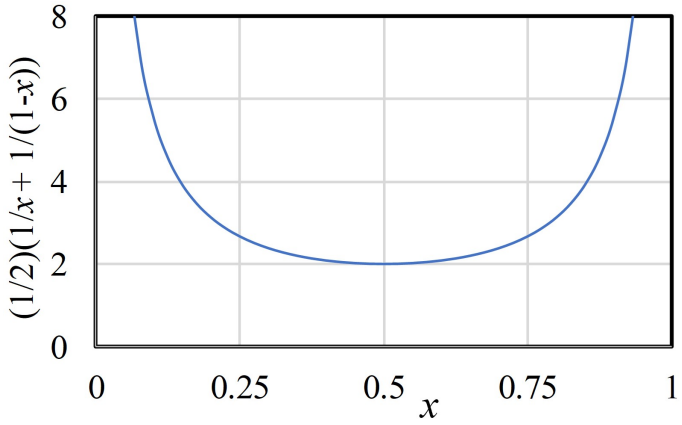


Figure 29. Graph of $\frac{1}{2}\left(\frac{1}{x}+\frac{1}{1-x}\right)$ in the BCE approximation.

Then, the binary cross-entropy (BCE) can be expanded as:

$$\begin{aligned}
-\log p_\theta(\boldsymbol{x}|\boldsymbol{z}) &= -\log \prod_{i=1}^{m} \hat{x_i}^{x_i}\,(1-\hat{x_i})^{(1-x_i)} \\
&= \sum_{i=1}^{m}(-x_i \log \hat{x_i} - (1-x_i)\log(1-\hat{x_i})) \\
&= \sum_i \left(-x_i \log\left(1+\frac{\delta x_i}{x_i}\right) - (1-x_i)\log\left(1-\frac{\delta x_i}{1-x_i}\right)\right) \\
&\quad + \sum_i (-x_i \log(x_i) - (1-x_i)\log(1-x_i)). \qquad (100)
\end{aligned}$$

Here, the second term of the last equation is a constant $C_{\boldsymbol{x}}$ depending on $\boldsymbol{x}$. Using $\log(1+x) = x - x^2/2 + O(x^3)$, the first term of the last equation is further expanded as follows:

$$\begin{aligned}
&\sum_i \left(-x_i\left(\frac{\delta x_i}{x_i} - \frac{{\delta x_i}^2}{2{x_i}^2}\right) - (1-x_i)\left(-\frac{\delta x_i}{1-x_i} - \frac{{\delta x_i}^2}{2\,(1-x_i)^2}\right) + O\left({\delta x_i}^3\right)\right) \\
&\qquad = \sum_i \left(\frac{1}{2}\left(\frac{1}{x_i}+\frac{1}{1-x_i}\right){\delta x_i}^2 + O\left({\delta x_i}^3\right)\right). \qquad (101)
\end{aligned}$$

As a result, a metric tensor $\boldsymbol{G_x}$ can be approximated as the following positive definite Hermitian matrix:

$$\boldsymbol{G_x} = \begin{pmatrix} \frac{1}{2}\left(\frac{1}{x_1}+\frac{1}{1-x_1}\right) & 0 & \cdots \\ 0 & \frac{1}{2}\left(\frac{1}{x_2}+\frac{1}{1-x_2}\right) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}. \qquad (102)$$

Here, the loss function in each dimension $\frac{1}{2}\left(\frac{1}{x_1}+\frac{1}{1-x_1}\right)$ is a downward-convex function as shown in Figure 29.

Structural similarity (SSIM):
Structural similarity (SSIM) (Wang et al., 2001) is widely used for picture quality metric, which is close to subjective quality. Let SSIM be a SSIM value between two pictures. The range of the SSIM is between 0 and 1. The higher the value, the better the quality. In this appendix, we also show that $(1-\text{SSIM})$ can be approximated to a quadratic form such as ${}^t\delta\boldsymbol{x}\,\boldsymbol{G_x}\delta\boldsymbol{x}$.

$\text{SSIM}_{N\times N(h,v)}(\boldsymbol{x},\boldsymbol{y})$ denotes a SSIM value between $N\times N$ windows in pictures X and Y, where $\boldsymbol{x}\in\mathbb{R}^{N^2}$ and $\boldsymbol{y}\in\mathbb{R}^{N^2}$ denote $N\times N$ pixels cropped from the top-left coordinate (h,v) in the images X and Y, respectively. Let $\mu_{\boldsymbol{x}}$, $\mu_{\boldsymbol{y}}$ be the

averages of all dimensional components in $\boldsymbol{x}$, $\boldsymbol{y}$, and $\sigma_{\boldsymbol{x}}$, $\sigma_{\boldsymbol{y}}$ be the variances of all dimensional components in $\boldsymbol{x}$, $\boldsymbol{y}$ in the $N \times N$ windows, respectively. Then, $\mathrm{SSIM}_{N\times N(h,v)}(\boldsymbol{x},\boldsymbol{y})$ is derived as

$$\mathrm{SSIM}_{N\times N(h,v)}(\boldsymbol{x},\boldsymbol{y}) = \frac{2\mu_{\boldsymbol{x}}\mu_{\boldsymbol{y}}}{{\mu_{\boldsymbol{x}}}^2+{\mu_{\boldsymbol{y}}}^2}\cdot\frac{2\sigma_{\boldsymbol{xy}}}{{\sigma_{\boldsymbol{x}}}^2+{\sigma_{\boldsymbol{y}}}^2}. \tag{103}$$

In order to calculate a SSIM value for a picture, the window is shifted in a whole picture and all of SSIM values are averaged. Therefore, if $\left(1-\mathrm{SSIM}_{N\times N(h,v)}(\boldsymbol{x},\boldsymbol{y})\right)$ is expressed as a quadratic form ${}^t\delta\boldsymbol{x}\,\boldsymbol{G}_{(h,v)\boldsymbol{x}}\,\delta\boldsymbol{x}$, $(1-\mathrm{SSIM})$ can be also expressed in quadratic form ${}^t\delta\boldsymbol{x}\,\boldsymbol{G}_{\boldsymbol{x}}\delta\boldsymbol{x}$.

Let $\delta\boldsymbol{x}$ be a minute displacement of $\boldsymbol{x}$. $\mu_{\delta\boldsymbol{x}}$ and ${\sigma_{\delta\boldsymbol{x}}}^2$ denote an average and variance of all dimensional components in $\delta\boldsymbol{x}$, respectively. Then, SSIM between $\boldsymbol{x}$ and $\boldsymbol{x}+\delta\boldsymbol{x}$ can be approximated as:

$$\mathrm{SSIM}_{N\times N(h,v)}(\boldsymbol{x},\boldsymbol{x}+\delta\boldsymbol{x}) \simeq 1-\frac{{\mu_{\delta\boldsymbol{x}}}^2}{2{\mu_x}^2}-\frac{{\sigma_{\delta\boldsymbol{x}}}^2}{2{\sigma_x}^2}+O\left(\left(|\delta\boldsymbol{x}|/|\boldsymbol{x}|\right)^3\right). \tag{104}$$

Then ${\mu_{\delta\boldsymbol{x}}}^2$ and ${\sigma_{\delta\boldsymbol{x}}}^2$ can be expressed as

$${\mu_{\delta\boldsymbol{x}}}^2 = {}^t\delta\boldsymbol{x}\,\boldsymbol{M}\delta\boldsymbol{x},\quad \text{where}\quad \boldsymbol{M}=\frac{1}{N^2}\begin{pmatrix}1 & 1 & \dots & 1\\ 1 & 1 & \dots & 1\\ \vdots & \vdots & \ddots & \vdots\\ 1 & 1 & \dots & 1\end{pmatrix}, \tag{105}$$

and

$${\sigma_{\delta\boldsymbol{x}}}^2 = {}^t\delta\boldsymbol{x}\,\boldsymbol{V}\delta\boldsymbol{x},\quad \text{where}\quad \boldsymbol{V}=\frac{1}{N}\boldsymbol{I}_N-M, \tag{106}$$

respectively. As a result, $\left(1-\mathrm{SSIM}_{N\times N(h,v)}(\boldsymbol{x},\boldsymbol{x}+\delta\boldsymbol{x})\right)$ can be expressed in the following quadratic form as:

$$1-\mathrm{SSIM}_{N\times N(h,v)}(\boldsymbol{x},\boldsymbol{x}+\delta\boldsymbol{x}) \simeq {}^t\delta\boldsymbol{x}\,\boldsymbol{G}_{(h,v)\boldsymbol{x}}\delta\boldsymbol{x},\quad \text{where } \boldsymbol{G}_{(h,v)\boldsymbol{x}}=\left(\frac{1}{2{\mu_x}^2}\boldsymbol{M}+\frac{1}{2{\sigma_x}^2}\boldsymbol{V}\right). \tag{107}$$

It is noted that $\boldsymbol{M}$ is a positive definite Hermitian matrix and $\boldsymbol{V}$ is a positive semidefinite Hermitian matrix. Therefore, $\boldsymbol{G}_{(h,v)\boldsymbol{x}}$ is a positive definite Hermitian matrix. As a result, $(1-\mathrm{SSIM})$ can be also expressed in quadratic form ${}^t\delta\boldsymbol{x}\,\boldsymbol{G}_{\boldsymbol{x}}\delta\boldsymbol{x}$, where $\boldsymbol{G}_{\boldsymbol{x}}$ is a positive definite Hermitian matrix.

Log-likelihood of Gaussian distribution:
Gaussian distribution is described as:

$$p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \prod_{i=1}^{m}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x_i-\hat{x_i})^2/2\sigma^2} = \prod_{i=1}^{m}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-{\delta x_i}^2/2\sigma^2}, \tag{108}$$

where σ^2 is a variance as a hyper parameter. Then, the log-likelihood of the Gaussian distribution is denoted as:

$$-\log p_\theta(\boldsymbol{x}|\boldsymbol{z}) = -\log\prod_{i=1}^{m}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\delta x_i^2/2\sigma^2} = \frac{1}{2\sigma^2}\sum_{i=1}^{m}\delta x_i^2+\frac{m}{2}\log(2\pi\sigma^2). \tag{109}$$

Since he first term is $(1/2\sigma^2)\,{}^t\delta\boldsymbol{x}\boldsymbol{I}_m\delta\boldsymbol{x}$, $\boldsymbol{G}_{\boldsymbol{x}}=(1/2\sigma^2)\,\boldsymbol{I}_m$ holds. $C_{\boldsymbol{x}}$ is the second term of the last equation in Eq.109.

H. Detail of the Experiment in Section 5.3

In this section, we provide further detail of experiment in Section 5.3.

H.1. Datasets

We describe the detail of following four public datasets:

Table 5. Hyper parameter for RaDOGAGA

Dataset	Autoencoder	Transform loss	λ_1	λ_2	
KDDCup99	200, 100, 10, 100, 200	L2	30000	6000	
Thyroid	60, 30, 6, 30, 60	L2	6000	18000	
Arrhythmia	200, 100, 50, 100, 200	L2	6000	24000	
KDDCup-rev	200, 50, 20, 50, 200	SSE	30000	6000	

KDDCUP99 (Dua & Graff, 2019) The KDDCUP99 10 percent dataset from the UCI repository is a dataset for cyber-attack detection. This dataset includes 494,021 instances. Each instance contains 34 continuous and 7 categorical features. We use one hot representation to encode the categorical features, and finally obtain a dataset with features of 121 dimensions. Only 20% of instances labeled -normal- and the rest labeled as -attacks-. Therefore, -normal- instances are used as anomalies, because they are in a minority group.

Thyroid (Dua & Graff, 2019) This dataset consists of 3,772 data sample with 6-dimensional feature from patients. Each instance can be divided in three classes: normal (not hypothyroid), hyperfunction, and subnormal functioning. We regard the hyperfunction class (2.5%) as an anomaly and rest two classes as normal.

Arrhythmia (Dua & Graff, 2019) This is dataset to detect cardiac arrhythmia which containing 452 instances with 274-dimensional feature. We treat minor classes (3, 4, 5, 7, 8, 9, 14, and 15, accounting for 15% of the total) as anomalies. The rest of classes are treated as normal.

KDDCUP-Rev (Dua & Graff, 2019) This is a revised version of KDDCUP99. To treat "normal" instances as the majority in the KDDCUP dataset, we keep all -normal- instances and randomly pick up -attack- instances so that the ratio of -normal- and -attack- to be 8:2. The number of instances is 121,597 in the end.

Data is max-min normalized toward dimension through the entire dataset, which is the same setting as previous studies.

H.2. Network architecture, Hyperparameter, and Training Detail

The VAE in this experiment consists of FC layers. Expect for the last layer of the encoder, Leaky ReLU (for KDDCup99, Thyroid, and Arrhythmia) or tanh (for KDDCup-rev) is attached as the activation function.

In this experiment, VAE is constructed by the form of decomposed loss to promote isometricity as explained in Remark 1. Here, the deconposed loss function $L_{\boldsymbol{x}}$ is set to $\lambda_1 D(\boldsymbol{x}, \breve{\boldsymbol{x}}) + \lambda_2 D(\breve{\boldsymbol{x}}, \hat{\boldsymbol{x}}) + D_{\mathrm{KL}}(\cdot)$, meaning, $\lambda_2 = \beta^{-1}$, are adjusted independently for reconstruction loss and transform loss. For the transform loss, we tested both L2 norm and SSE loss and choose the better one for each dataset. The reason for introducing L2 loss to the transform loss is as follows. The reduction of the transform loss promotes the isometricity, as explained in Remark 1 of Section 4.2. Since the derivative of L2 norm is steeper than SSE used in coding loss, the use of L2 norm for transform loss will reduce the value of transform loss explicitly and promote the isometricity.

Hyperparameter is described in Table 5. The first column is the number of neurons. For Thyroid, we also tested the size of (30, 24, 6, 24, 30). For other datasets, we tested the size of (200, 100 or 50, 10 or 20 or 50, 100 or 50, 200). The second column is the type of reconstruction loss. (λ_1, λ_2) is determined experimentally. Both of them varied from 6000 to 30000 by every 6000 intervals. For all datasets, optimization is done by Adam optimizer with a learning rate of 1×10^{-4} with batch size of 1024. The epoch numbers for each dataset are 600, 40000, 30000, and 600 respectively. Test models are saved by every 1/10 epochs and early stop is applied. For this experiment, we use GeForce GTX 1080.

H.3. Precision, Recall, and F1

Due to the page limitation, we reported only F1 score in main paper. Now we provide Precision and Recall Score as well in Table 6 .

Table 6. Average and standard deviations (in brackets) of Precision, Recall and F1

Dataset	Methods	Precision	Recall	F1
KDDCup	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)
	VAE	**0.9568(0.0007)**	**0.9718 (0.0007)**	**0.9642(0.0007)**
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)**
	VAE	0.6458 (0.04270)	0.6739 (0.04455)	0.6596 (0.0436)
Arrythmia	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)
	VAE	0.4912(0.0406)	0.5061 (0.0419)	0.4985 (0.0413)
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)
	VAE	**0.9880 (0.0008)**	**0.9881 (0.0008)**	**0.9880 (0.0008)**

†Scores are cited from Liao et al. (2018) (GMVAE) and Kato et al. (2020)(DAGMM, RaDOGAGA)