Learning from Nested Data with Ornstein Auto-Encoders

Youngwon Choi [1 2] **Sungdong Lee** [1] **Joong-Ho Won** [1]

Abstract

Many of real-world data, e.g., the VGGFace2 dataset, which is a collection of multiple portraits of individuals, come with nested structures due to grouped observation. The Ornstein auto-encoder (OAE) is an emerging framework for representation learning from nested data, based on an optimal transport distance between random processes. An attractive feature of OAE is its ability to generate new variations nested within an observational unit, whether or not the unit is known to the model. A previously proposed algorithm for OAE, termed the random-intercept OAE (RIOAE), showed an impressive performance in learning nested representations, yet lacks theoretical justification. In this work, we show that RIOAE minimizes a loose upper bound of the employed optimal transport distance. After identifying several issues with RIOAE, we present the product-space OAE (PSOAE) that minimizes a tighter upper bound of the distance and achieves orthogonality in the representation space. PSOAE alleviates the instability of RIOAE and provides more flexible representation of nested data. We demonstrate the high performance of PSOAE in the three key tasks of generative models: exemplar generation, style transfer, and new concept generation.

1. Introduction

Many real-world data are collected in grouped observation units. The resulting sample naturally possesses a nested structure. For example, in the VGGFace2 dataset (Cao et al., 2018), there are 3.31 million portraits of 9131 people, i.e., 362.6 images for each person on average. Certainly, portraits of the same person are highly correlated. Likewise, the images from MNIST dataset also retain correlated structure for each digit. In multicenter electronic health records, patients are nested within a hospital. For such nested data, representation learning aims to find a representation where within-unit variation and between-unit variation are well-separated in the representation space. It is also desirable that the model can deal with an unknown, possibly unbounded, total number of observational units. Similarly, it should be considered that an unbounded number of variations can be observed within a unit. In the VGGFace2 data, for example, the identity of a portrait can be considered as an observational unit. The number of these units is possibly infinite, and the available sample may not include all the units. All of the possible variations from a unit may not be observed either. In the MNIST data, on the other hand, the number of observational units (digits) is finite and known. In sum, we need a model that can adopt various nested structure.

As a concrete application of nested representation learning, consider face unlock systems for smartphones. The training of such a system is highly subject to data imbalance since in addition to the initial training database, the acquired face images are based on a few snapshots of the users. Further, the number of users in the database keeps increasing. The underlying face recognition algorithm will benefit if each user is well-separated from others in the representation space, and the images of these users can be augmented, including those of virtual users.

The structure acquired from a representation learning model can be demonstrated by sample generation. Three types of tasks have been advocated to assess the quality of a generative model (Zhu et al., 2017; Lake et al., 2019): 1) exemplar generation, which generates new variations of a given observational unit, 2) style transfer, which transfers the variations within a given observation unit to another one, and 3) new concept generation, which amounts to simulating new observational units; this can be combined with tasks 1 and 2. An adequate representation for such nested data should be a single representation that can address all three tasks for an unbounded number of observational units and variations.

Generative latent variable models (LVMs) such as the generative adversarial networks (GAN, Goodfellow et al., 2014), the variational auto-encoder (VAE, Kingma & Welling, 2014), and the Wasserstein auto-encoder (WAE, Tolstikhin

[1]Department of Statistics, Seoul National University. [2]Current affiliation: UCLA Center for Vision & Imaging Biomarkers. Correspondence to: Joong-Ho Won <wonj@stats.snu.ac.kr>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

et al., 2018) have delivered promising outcomes. Extensions of these approaches to structured data have mostly focused on obtaining disentangled representation in semi-supervised learning settings (Makhzani et al., 2016; Chen et al., 2016; Zhao et al., 2017; Lopez et al., 2018), but they are not suitable for the nested data structure we consider. For example, the conditional adversarial auto-encoder (CAAE, Makhzani et al., 2016) can be thought as learning an LVM for each conditional distribution. This can carry out exemplar generation and style transfer tasks by interpreting each observational unit as a class. However, assuming a fixed number of units (classes), these models cannot generate a sample of a new observational unit not present in the training data. Similar limitations are present in extensions of VAE (Kingma et al., 2014; Louizos et al., 2016; Lopez et al., 2018), GAN (Chen et al., 2016), and WAE (Patrini et al., 2020).

To obtain a single representation that addresses all the three tasks, a sensible approach is to model the nested structure in the latent space directly, and find appropriate mappings between them. For example, the random intercept model (Diggle et al., 2002; Fitzmaurice et al., 2012) is a common approach in statistics to model nested data:

$$Z_j^i = B^i + E_j^i, \qquad B^i \perp\!\!\!\perp E_j^i,$$
$$B^i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \tau_0^2 \mathbf{I}), \quad E_j^i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_0^2 \mathbf{I}), \tag{1}$$

where Z_j^i denotes the jth observation in unit i. Each unit i is represented by the random intercept B^i. Differing numbers of samples between units are also naturally handled. A noticeable feature of model (1) is that it defines an *exchangeable* sequence: within a unit, the order of observation should not matter. Thus the nested data can be considered as an independent, identically distributed (i.i.d.) copy of the exchangeable random process. Interpreting nested-structured data as i.i.d. observations of a random process, or, moreover, those of an exchangeable random process, provides a fruitful viewpoint on disentangling representations. For both VGGFace2 and MNIST data, permuting the order of portraits in each person or handwritings in each digit does not notably affect any learning tasks commonly undertaken, so we can see them as exchangeable sequences.

The Ornstein auto-encoder (OAE, Choi & Won, 2019) conducts nested data generation from this point of view. OAEs trained with the algorithm proposed by Choi & Won (2019) has shown impressive performance in discriminating individuals from the VGGFace2 data and digits from highly imbalanced MNIST data in the latent space. An interesting feature of OAE is that it can generate samples from an observational unit *whether or not that unit is present in the training dataset*, and can even generate a new unit from the latent space. *Nested within a given unit*, old or new, data with either new variations or those transferred from other known unit can be generated. For example, if an OAE is trained with the VGGFace2 dataset, then infinite variations portraits of a single person, whether the person is present in the dataset or not, can be generated. To our knowledge, this is the first framework that can perform all of these three tasks with the complex real-world data. This feature makes OAE attractive to many applications that suffer from data imbalance. Unfortunately, however, a theoretical claim on which the algorithm of Choi & Won (2019) is based on turns out to be incorrect, as we will see in the sequel. Their algorithm thus leaves an intriguing question on the gap between the practical performance and theoretical justification.

Contributions The goal of this paper is to fill in this gap and provide an improved learning algorithm with a better theoretical justification. We first show that the claim of Choi & Won (2019) that an optimal transport distance between exchangeable processes reduced to a simpler Wasserstein distance is incorrect. We then show that the algorithm of Choi & Won (2019) actually optimizes an *upper bound* of the optimal transport distance, namely Ornstein's d-bar distance. Based on this observation, we proceed with deriving a tighter upper bound and propose an algorithm that optimizes this improved upper bound. This novel bound also imposes an explicit constraint that the two latent variables encoding within-unit and between-unit variations should be independent, which is lacking in the previous approach. Thanks to this separation, the present algorithm for OAE shows much improved performance in all of the three learning tasks and remarkable improvement on the tasks in which other attempts fails.

After developing background material in Section 2, we review the existing work on OAE and derive an upper bound of the d-bar distance in Section 3. Section 4 examines the problems arising from the existing algorithm, and suggests an improved learning algorithm. In Section 5, we exhibit the performance of our algorithm using the VGGFace2 and MNIST datasets. Section 6 gives conclusion of this paper. Proofs of the proposition, details of the implementation, and additional examples are in the Supplement.

Notation The spaces of observable variables and latent variables are denoted as $\mathcal{X}$ and $\mathcal{Z}$, respectively. Both spaces are assumed to be complete, separable metric spaces in which (regular) conditional distributions are well-defined. The metric associated with $\mathcal{X}$ is denoted d. A Cartesian product space of $\mathcal{X}$ is denoted by $\mathcal{X}^n$ for $n = 1, 2, \ldots$, with $n = \infty$ permitted, where $\mathcal{X}^\infty = \{(\ldots, x_{-1}, x_0, x_1, \ldots) : x_j \in \mathcal{X}\}$. With the Borel σ-field and a probability measure on $\mathcal{X}$ (resp. $\mathcal{Z}$), event space and probability measure on any product space, including $\mathcal{X}^\infty \times \mathcal{Z}^\infty$, are well-defined. (Regular) conditional distributions related to random variables (processes) defined on these probability spaces are also well-defined. (Event spaces are omitted unless necessary.) Capital letters (e.g., X) indicate ran-

dom variables, and their realizations are noted in lower case letters (e.g., x). Doubly-infinite random processes and their realizations are denoted by boldfaces: e.g., $\mathbf{X}$ and $\mathbf{x}$. Superscripts, as in $\mathbf{X}^i$ (resp. X^i), are used to indicate an (ith) i.i.d. copy of $\mathbf{X}$ (resp. X). Subscripts are used to represent coordinates of a random process, e.g., $\mathbf{X} = (\cdots, X_{-1}, X_0, X_1, \cdots) \in \mathcal{X}^\infty$. A finite-length random sequence is denoted as $\mathbf{X}_{1:n} = (X_1, \cdots, X_n)$. The probability distribution of random process $\mathbf{X}$ is denoted by $P_\mathbf{X}$, etc.; we use Q in place of P if the distribution is subject to optimization. Given a measurable function $\mathbf{g} : \mathcal{X}^\infty \to \mathcal{Y}$, where $\mathcal{Y}$ is another metric space, the distribution of $\mathbf{g}(\mathbf{X})$ is denoted by the pushforward $\mathbf{g}_\sharp P_\mathbf{X}$.

2. Preliminaries

2.1. Generative latent variable models

Generative latent variable models (LVMs) refer to a family of parametric models that learn an unknown distribution P_X on a high-dimensional space $\mathcal{X}$ by using a latent variable in a low-dimensional space $\mathcal{Z}$. Learning is conducted by considering a "decoder" or conditional distribution $Q_{Y|Z}$ of a random variable Y on $\mathcal{X}$ given Z and guiding it so that the marginal distribution $P_Y = \int Q_{Y|Z} dP_Z$ is close to P_X in the sense that some divergence $\mathcal{D}$ between P_X and P_Y are minimized. Often the decoder is chosen to be deterministic, i.e., $Q_{Y|Z}$ is such that $Y = g(Z)$ a.s. for some $g : \mathcal{Z} \to \mathcal{X}$. If g belongs to a set $\mathcal{G}_{NN}$ that can be parameterized by a neural network, then a LVM seeks $\inf_{g \in \mathcal{G}_{NN}} \mathcal{D}(P_X, g_\sharp P_Z)$, where $g_\sharp P_Z$ is the marginal distribution of Y induced by g and P_Z.

Popular choices for the divergence $\mathcal{D}$ includes that of GAN:

$$\mathcal{D}_{\text{GAN}}(P_X, g_\sharp P_Z) = \sup_{f \in \mathcal{F}_{NN}} \{\mathbb{E}_{P_X} \log f(X) + \mathbb{E}_{P_Z} \log[1 - f(g(Z))]\}$$

for $\mathcal{F}_{NN}$ being a set of functions from $\mathcal{X}$ to $(0, 1)$ parameterized by a neural network, and that of WAE:

$$\mathcal{D}_{\text{WAE}}(P_X, g_\sharp P_Z) = \inf_{Q_{Z|X} \in \mathcal{Q}_{Z|X}} \mathbb{E}_{P_X} \mathbb{E}_{Q_{Z|X}} d^p(X, g(Z))$$

for some $p \geq 0$; the $\mathcal{Q}_{Z|X}$ is the set of all conditional distributions $Q_{Z|X}$ such that $\int Q_{Z|X} dP_X = P_Z$. Additionally, the maximum mean discrepancy divergence (MMD, Gretton et al., 2012) is defined as

$$\mathcal{D}_{\text{MMD},\kappa}(P_X, P_Y) = \|\mathbb{E}_{P_X} \kappa(\cdot, X) - \mathbb{E}_{P_Y} \kappa(\cdot, Y)\|^2_\mathcal{H}$$

for a bounded reproducing kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ inducing a Hilbert space $\mathcal{H}$ with inner product such that $\langle \kappa(\cdot, x), f \rangle_\mathcal{H} = f(x)$ and distributions P_X, P_Y on $\mathcal{X}$.

2.2. Ornstein's d-bar distance and OAE

Suppose that $\mathbf{X}$ and $\mathbf{Y}$ are two stationary processes in $\mathcal{X}^\infty$. Let $\bar{\rho}_n(P_{\mathbf{X}_{1:n}}, P_{\mathbf{Y}_{1:n}}) \triangleq \inf_{\pi \in \mathcal{P}(P_{\mathbf{X}_{1:n}}, P_{\mathbf{Y}_{1:n}})} E_\pi \rho_n(\mathbf{X}_{1:n}, \mathbf{Y}_{1:n})$ where $\mathcal{P}(P_{\mathbf{X}_{1:n}}, P_{\mathbf{Y}_{1:n}})$ is the set of joint distributions of sequences $(\mathbf{X}_{1:n}, \mathbf{Y}_{1:n}) \in \mathcal{X}^n \times \mathcal{X}^n$ having $P_{\mathbf{X}_{1:n}}$ and $P_{\mathbf{Y}_{1:n}}$ as marginals, and $\rho_n(\mathbf{X}_{1:n}, \mathbf{Y}_{1:n}) \triangleq n^{-1} \sum_{j=1}^n d^p(X_j, Y_j)$ for $p \geq 0$; d^0 represents the 0-1 loss. Ornstein's d-bar distance between the two random processes is defined as $\bar{d}_p(P_\mathbf{X}, P_\mathbf{Y}) \triangleq \bar{\rho}^{\min(1,1/p)}(P_\mathbf{X}, P_\mathbf{Y})$, where $\bar{\rho}(P_\mathbf{X}, P_\mathbf{Y}) \triangleq \sup_n \bar{\rho}_n(P_{\mathbf{X}_{1:n}}, P_{\mathbf{Y}_{1:n}})$ (Ornstein, 1973; Gray et al., 1975). Note that this is a random process version of the p-Wasserstein distance (see, e.g., Bousquet et al., 2017). Gray et al. (1975) show that the d-bar distance is a true distance for all possible stationary processes in $\mathcal{X}^\infty$, and furthermore, the equality

$$\bar{\rho}(P_\mathbf{X}, P_\mathbf{Y}) = \inf_{\pi \in \mathcal{P}_s(P_\mathbf{X}, P_\mathbf{Y})} \mathrm{E}_\pi d^p(X_0, Y_0), \tag{2}$$

where $\mathcal{P}_s(P_\mathbf{X}, P_\mathbf{Y})$ is the set of distributions of jointly stationary processes $(\mathbf{X}, \mathbf{Y}) \in \mathcal{X}^\infty \times \mathcal{X}^\infty$ having $P_\mathbf{X}$ and $P_\mathbf{Y}$ as marginals holds; $X_0 \overset{d}{=} X_1$ and $Y_0 \overset{d}{=} Y_1$.

The Ornstein auto-encoder (OAE) is a LVM that employs $\bar{\rho}(P_\mathbf{X}, P_\mathbf{Y})$ for $\mathcal{D}(P_X, P_Y)$. It defines a latent random process $\mathbf{Z} \in \mathcal{Z}^\infty$ with prior distribution $P_\mathbf{Z}$, and learns a deterministic decoder $\mathbf{g} : \mathcal{Z}^\infty \to \mathcal{X}^\infty$ that maps a stationary sequence to a stationary sequence, and $\mathbf{Y} = \mathbf{g}(\mathbf{Z})$ a.s. Clearly, WAE is a special case of OAE for an i.i.d. sequence. In this setting, similar to WAE, a reparameterization of (2):

$$\bar{\rho}(P_\mathbf{X}, \mathbf{g}_\sharp P_\mathbf{Z}) = \inf_{Q_{\mathbf{Z}|\mathbf{X}} \in \mathcal{Q}_{\mathbf{Z}|\mathbf{X}}} \mathbb{E}_{P_\mathbf{X}} \mathbb{E}_{Q_{\mathbf{Z}|\mathbf{X}}} d^p(X_0, [\mathbf{g}(\mathbf{Z})]_0), \tag{3}$$

can be made (Choi & Won, 2019, Theorem 1). Here, $\mathcal{Q}_{\mathbf{Z}|\mathbf{X}}$ is the set of conditional distributions (probabilistic encoders) $Q_{\mathbf{Z}|\mathbf{X}}$ such that $Q_{\mathbf{Z}|\mathbf{X}} P_\mathbf{X}$ is jointly stationary in $(\mathbf{X}, \mathbf{Z})$ and the aggregate posterior $Q_\mathbf{Z} \triangleq \int Q_{\mathbf{Z}|\mathbf{X}} dP_\mathbf{X}$ is equal to $P_\mathbf{Z}$. By minimizing (3) over $\mathbf{g} \in \mathcal{G}_{NN}$, we obtain an OAE model.

The use of a process optimal transport distance enables the flexible nested sample generation feature of OAE.Yet, the infinite dimensional nature of the encoder-decoder pair prevents OAE from realization.

3. Random-intercept OAE

3.1. OAE for exchangeable data

The simplest case that an OAE offers a tractable algorithm is when both processes $\mathbf{X}$ and $\mathbf{Z}$ are i.i.d., in which equation (3) reduces to the pth power of the p-Wasserstein distance between single coordinates, i.e., $\bar{\rho}_1(P_{X_0}, g_\sharp P_{Z_0})$ and

$\mathbf{g}(\mathbf{Z}) = (\dots, g(Z_{-1}), g(Z_0), g(Z_1), \dots)$. Hence an algorithm for WAEs (Tolstikhin et al., 2018) can be employed.

Motivated by this simplification, Choi & Won (2019) claimed that the same reduction is possible if the pair process $\{(X_j, Z_j)\}$ is exchangeable. This claim is plausible since De Finetti's theorem states that an exchangeable sequence is conditionally i.i.d. However, their claim leans on the assumption that the marginal distribution of (X_0, Y_0) in $\mathcal{P}(P_{\mathbf{X}_0}, P_{\mathbf{Y}_0})$ has the representation $\int F^1 dP_F$ for a random distribution F^1 on $\mathcal{X} \times \mathcal{X}$ and its distribution P_F, which is not true in general except for genuinely i.i.d. sequences.

Despite the absence of a theoretical support, the algorithm of Choi & Won (2019), reproduced in the Supplement (Algorithm i) for subsequent references, shows successful results. The following result sheds light on theoretical justification of their algorithm, and is important in its own right since it provides an upper bound of $\bar{\rho}(P_{\mathbf{X}}, \mathbf{g}_\sharp P_{\mathbf{Z}})$ in terms of single coordinates of the processes. Recall that a version of De Finetti's theorem (Olshen, 1974) ensures the existence of a real-valued random variable conditioned on which the coordinates of $\mathbf{X}$ are i.i.d. when the sequence $\mathbf{X}$ is exchangeable.

Theorem 3.1. *Assume process distributions $P_{\mathbf{X}}$ on $\mathcal{X}^\infty$ and $P_{\mathbf{Z}}$ on $\mathcal{Z}^\infty$ are both exchangeable. Also assume that there exists a distribution P_B on another complete, separable metric space $\mathcal{B}$ (e.g., $\mathcal{B} = \mathbb{R}$) such that its joint distributions $P_{\mathbf{X},B}$ and $P_{\mathbf{Z},B}$ satisfy $P_{\mathbf{X_{1:n}},B} = \big[\prod_{j=1}^n P_{X_0|B}\big] P_B$ and $P_{\mathbf{Z_{1:n}},B} = \big[\prod_{j=1}^n P_{Z_0|B}\big] P_B$ for any n, respectively. Then, for any measurable $\mathbf{g} : \mathcal{Z}^\infty \to \mathcal{X}^\infty$ that maps an exchangeable sequence to an exchangeable sequence, we have*

$$\bar{\rho}(P_{\mathbf{X}}, \mathbf{g}_\sharp P_{\mathbf{Z}}) \leq \inf_{Q_{\mathbf{Z}|\mathbf{X},B} \in \mathcal{Q}} \mathbb{E}_{P_{\mathbf{X},B}} \mathbb{E}_{Q_{\mathbf{Z}|\mathbf{X},B}} d^p(X_0, [\mathbf{g}(\mathbf{Z})]_0). \quad (4)$$

Here, $\mathcal{Q}$ is the set of all conditional distributions $Q_{\mathbf{Z}|\mathbf{X},B}$ such that the joint distribution $Q_{\mathbf{Z}|\mathbf{X},B} P_{\mathbf{X},B}$ of $(\mathbf{X}, \mathbf{Z}, B)$ has marginals $\big[\prod_{j=1}^n Q_{Z_0|X_0,B}\big] P_{\mathbf{X}_{1:n},B}$ and $P_{\mathbf{Z_{1:n}}}$ on $(\mathbf{X}_{1:n}, \mathbf{Z}_{1:n}, B)$ and $\mathbf{Z}_{1:n}$ for any n, respectively.

$\mathcal{D}_{OAE}(P_{\mathbf{X}}, \mathbf{g}_\sharp P_{\mathbf{Z}})$. For this upper bound to be tractable, 1) the random variable B that deconvolves both $\mathbf{X}$ and $\mathbf{Z}$ needs to be known; and 2) the complexity of the decoder $\mathbf{g}$ needs to be addressed. The latter issue can be resolved by requiring $\mathbf{g}(\mathbf{Z}) = (\dots, g(Z_{-1}), g(Z_0), g(Z_1), \dots)$ for some measurable function $g : \mathcal{Z} \to \mathcal{X}$. In this case, with a slight abuse of notation, we write $\mathcal{D}_{OAE}(P_{\mathbf{X}}, \mathbf{g}_\sharp P_{\mathbf{Z}})$ as

$$\mathcal{D}_{\text{OAE}}(P_{\mathbf{X}}, g_\sharp P_{\mathbf{Z}}) = \inf_{Q_{\mathbf{Z}|\mathbf{X},B} \in \mathcal{Q}} \mathrm{E}_{P_{\mathbf{X},B}} \mathrm{E}_{Q_{\mathbf{Z}|\mathbf{X},B}} d^p(X_0, g(Z_0)). \quad (5)$$

Note that divergence (5) coincides with divergence (3) when both $\mathbf{X}$ and $\mathbf{Z}$ are i.i.d., that is, the OAE reduces to the WAE.

3.2. Random-intercept OAE

To address the first issue of the last paragraph, Algorithm i enforces exchangeablility to $P_{\mathbf{Z}}$ using the random intercept model similar to (1): for given distributions P_B and P_{E_0}, the joint distribution $P_{\mathbf{Z},B}$ of the random process $\mathbf{Z}^i$ and random variable B^i for unit i is specified by

$$Z_j^i = B^i + E_j^i,\ B^i \overset{\text{i.i.d.}}{\sim} P_B,\ E_j^i \overset{\text{i.i.d.}}{\sim} P_{E_0},\ B^i \perp\!\!\!\perp E_j^i. \quad (6)$$

By De Fenetti's theorem, there is a random variable $\hat{B}$ that deconvolves $\mathbf{X} \overset{d}{=} \mathbf{X}^i$, i.e., coordinates of $\mathbf{X} = (\dots, X_{-1}, X_0, X_1, \dots)$ are conditionally i.i.d. given $\hat{B}$. Then the joint distribution $Q_{\mathbf{X},\hat{B}}$ of $\mathbf{X}$ and $\hat{B}$ has $\mathbf{X}$-marginal $P_{\mathbf{X}}$. Hence the conditional distribution $Q_{\hat{B}|\mathbf{X}}$ is well-defined. Ideally and in order to satisfy the assumption of Theorem 3.1 as well, the $\hat{B}$-marginal of $Q_{\mathbf{X},\hat{B}}$ should equal to P_B. Hence we require

$$\int_{\mathcal{X}^\infty} Q_{\hat{B}|\mathbf{X}} dP_{\mathbf{X}} = P_B \quad (7)$$

and treat $Q_{\hat{B}|\mathbf{X}}$ as an "optimization variable" to learn.

To this end, Algorithm i can be understood as solving the following optimization problem

$$\inf_{g \in \mathcal{G}_{NN}} \inf_{Q_{\hat{B}|\mathbf{X}}} \inf_{Q_{Z_0|X_0,B}} \big[\mathbb{E}_{P_{\mathbf{X}}} \mathbb{E}_{Q_{\hat{B}|\mathbf{X}}} \mathbb{E}_{Q_{Z_0|B,X_0}} d^p(X_0, g(Z_0)) + \lambda_1 \mathbb{E}_{P_B} \mathcal{D}_{\text{GAN}}(P_{Z_0|B}, \textstyle\int_{\mathcal{X}} Q_{Z_0|B,X_0} dP_{X_0|B}) + \lambda_2 \mathcal{D}_{\text{MMD},\kappa}(P_B, \textstyle\int_{\mathcal{X}^\infty} Q_{\hat{B}|\mathbf{X}} dP_{\mathbf{X}}) \big] \quad (8)$$

stochastically (Lines 8-9). The third term of the objective functional is a penalized form of the constraint (7). The second term is a penalized form of an additional constraint

$$\int_{\mathcal{X}} Q_{Z_0|B,X_0} dP_{X_0|B} = P_{Z_0|B} \quad (9)$$

that promotes, along with (7), for the pair $(Q_{Z_0|X_0,B}, Q_{\hat{B}|\mathbf{X}})$ to populate a set $\tilde{\mathcal{Q}}$ of conditional distributions $\tilde{Q}_{\mathbf{Z}|\mathbf{X},B} \in \mathcal{Q}$, where the latter set is defined in Theorem 3.1. Under these constraints, the first term coincides with the infimand of equation (5). The following result shows that $\tilde{\mathcal{Q}}$ is a proper subset of $\mathcal{Q}$.

Proposition 3.1. *Assume a process distribution $P_{\mathbf{X}}$ on $\mathcal{X}^\infty$ is exchangeable and the random variable $B \in \mathcal{Z}$ deconvolves $\mathbf{X}$ so that $P_{\mathbf{X_{1:n}},B} = \big[\prod_{j=1}^n P_{X_0|B}\big] P_B$ for any n. Given a distribution P_{E_0} of a random variable E_0 on $\mathcal{Z}$, assume the random intercept model (6) for $P_{\mathbf{Z}}$. If we define*

$$\mathcal{D}_{\text{RIOAE}}(P_{\mathbf{X}}, g_\sharp P_{\mathbf{Z}}) = \inf_{Q_{\mathbf{Z}|\mathbf{X},B} \in \mathcal{Q}^{RI}} \mathrm{E}_{P_{\mathbf{X},B}} \mathrm{E}_{Q_{\mathbf{Z}|\mathbf{X},B}} d^p(X_0, g(Z_0)). \quad (10)$$

where $\mathcal{Q}^{RI}$ is the set of conditional distributions $Q_{\mathbf{Z}|\mathbf{X},B}$ such that the joint distribution $P_{\mathbf{X},B} Q_{\mathbf{Z}|\mathbf{X},B}$ of

$(\mathbf{X}, \mathbf{Z}, B)$ has the marginal $\left[\prod_{j=1}^{n} Q_{Z_0|X_0,B}\right] P_{\mathbf{X}_{1:n},B}$ on $(\mathbf{X}_{1:n}, \mathbf{Z}_{1:n}, B)$ for any n, and constraint (9) holds. Then,

$$\mathcal{D}_{\mathrm{OAE}}(P_{\mathbf{X}}, g_\sharp P_{\mathbf{Z}}) \leq \mathcal{D}_{\mathrm{RIOAE}}(P_{\mathbf{X}}, g_\sharp P_{\mathbf{Z}}), \tag{11}$$

for any measurable $g : \mathcal{Z} \to \mathcal{X}$.

Thus Algorithm i minimizes an upper bound of the right-hand side of (4), which, in turn, is an upper bound of the left-hand side. For these reasons, the LVM implied by Algorithm i can be called a *random-intercept OAE*.

In the stochastic approximation to the third term of (8), sampling from $Q_{\hat{B}|\mathbf{X}}$ requires infinite-length data, hence it is infeasible. Line 6 of Algorithm i employs $Q_{\hat{B}|X_0}$, an encoder that takes only a single coordinate of data as input, instead of $Q_{\hat{B}|\mathbf{X}}$. For each unit i with m_i repeated measurements, it samples $\hat{b}^i_j \sim Q_{\hat{B}|X_0}(\cdot|x^i_j)$ for each $j = 1, \ldots, m_i$ and then aggregate $\hat{b}^i_j$'s to obtain $\hat{b}^i = \frac{1}{m_i}\sum_{j=1}^{m_i} \hat{b}^i_j$ to approximate a sample from $Q_{B|\mathbf{X}}$. This practice can be justified as follows. The version of De Fenetti's theorem employed here relies on the fact that there is a 1-to-1 mapping between the range of the limit of the empirical distribution of the coordinates X_j of $\mathbf{X}$ and the real line to identify the $\hat{B}$ (Olshen, 1974, p. 319). Thus the $\hat{b}^i$ can be roughly understood as the empirical distribution smoothed by this mapping; if $Q_{\hat{B}|X_0}$ is well-chosen, then we can expect $\hat{b}^i \overset{d}{=} \hat{B}$.

4. Product-space OAE

4.1. Issues with the random-intercept OAE

The present analysis of the random-intercept OAE (RIOAE) reveals at least three problems. First, the objective $\mathcal{D}_{\mathrm{RIOAE}}(P_{\mathbf{X}}, g_\sharp P_{\mathbf{Z}})$ is merely an upper bound of $\mathcal{D}_{\mathrm{OAE}}(P_{\mathbf{X}}, g_\sharp P_{\mathbf{Z}})$, which is already an upper bound of $\bar{\rho}(P_{\mathbf{X}}, g_\sharp P_{\mathbf{Z}})$ (Proposition 3.1). Furthermore, constraints (7) and (9) are not strong enough to impose the independence and additivity requirements of the random intercept model (6), causing an instability in training. Finally, constraint (9) is imposed *for all possible observational units*, i.e., for all possible values of B. Because of this excessive number of constraints, there has to be a sufficient number of observations with diverse variation for every unit to ensure successful training, which would be an exceptional feature for real-world data.

4.2. Product-space model for latent space

The random intercept model (6) is not the only way of imposing exchangeability to the sequence $\mathbf{Z} = (\cdots, Z_{-1}, Z_0, Z_1, \cdots)$ in $\mathcal{Z}^\infty$. More importantly, the additive structure of this model may limit the expressiveness of the entire generative model. A more flexible approach is to decompose $\mathcal{Z}$ into $\mathcal{I} \times \mathcal{V}$. For given distributions P_B on $\mathcal{I}$ and P_{E_0} on $\mathcal{V}$, we specify $P_{\mathbf{Z}}$ on $\mathcal{Z}^\infty$ by

$$Z^i_j = (B^i, E^i_j), B^i \overset{\text{i.i.d.}}{\sim} P_B,\ E^i_j \overset{\text{i.i.d.}}{\sim} P_{E_0}, B^i \perp\!\!\!\perp E^i_j. \tag{12}$$

Ideally, B^i encodes the "identity" of the observational unit shared among the coordinates of $\mathbf{Z}^i$ that is the ith independent copy of $\mathbf{Z}$, and E^i_j encodes the "within-unit variation" shared among all observational units. Clearly $\mathbf{Z}^i$ is exchangeable. Furthermore, the sequence $(\ldots, g(B^i, E^i_{-1}), g(B^i, E^i_0), g(B^i, E^i_1), \ldots)$ is exchangeable for any function $g : \mathcal{I} \times \mathcal{V} \to \mathcal{X}$. The additivity constraint $Z^i_j = B^i + E^i_j$ in (6) is absorbed into the decoder g and can be learned from data if appropriate. We call this more flexible approach to OAE the *product-space OAE*.

The key advantage of the product-space OAE (PSOAE) over RIOAE is that it directly optimizes the upper bound (5):

Theorem 4.1. *Assume the process distribution $P_{\mathbf{X}}$ on $\mathcal{X}^\infty$ is exchangeable, and the random variable B on $\mathcal{I}$ deconvolves $\mathbf{X}$ so that $P_{\mathbf{X_{1:n}},B} = \left[\prod_{j=1}^{n} P_{X_0|B}\right] P_B$ for any n. Suppose the process distribution $P_{\mathbf{Z}}$ on $\mathbf{Z}^\infty$ follows the product-space model (12). Then, for decoder $\mathbf{g}(\mathbf{Z}) = (\ldots, g(B, E_{-1}), g(B, E_0), g(B, E_1), \ldots)$, we have*

$$\mathcal{D}_{\mathrm{OAE}}(P_{\mathbf{X}}, g_\sharp P_{\mathbf{Z}}) = \inf_{Q_{E_0|X_0,B} \in \mathcal{Q}_{E_0}} \mathbb{E}_{P_{\mathbf{X}}} \mathbb{E}_{P_{B|\mathbf{X}}} \mathbb{E}_{Q_{E_0|X_0,B}} d^p(X_0, g(B, E_0)),$$

where $\mathcal{Q}_{E_0} = \{Q_{E_0|X_0,B} : \int_{\mathcal{X}} Q_{E_0|X_0,B} dP_{X_0|B} = P_{E_0}\}$.

If we let a joint distribution of (X_0, B, E_0) be $Q_{X_0,B,E_0} = Q_{E_0|X_0,B} P_{X_0,B}$, then the induced conditional distribution of E_0 given B is $Q_{E_0|B} = \int_{\mathcal{X}} Q_{E_0|X_0,B} dP_{X_0|B}$. Hence the constraint defining the set $\mathcal{Q}_{E_0}$ is written as

$$Q_{E_0|B} = P_{E_0}, \tag{13}$$

which implies that 1) E_0 and B are independent with respect to Q_{X_0,B,E_0}, and 2) the induced marginal Q_{E_0} is equal to the given marginal P_{E_0}. The latter condition can be written

$$P_{E_0} = \int_{\mathcal{X}\times\mathcal{I}} Q_{E_0|X_0,B} dP_{X_0,B}. \tag{14}$$

In the reformulation of $\mathcal{D}_{\mathrm{OAE}}(P_{\mathbf{X}}, g_\sharp P_{\mathbf{Z}})$ in Theorem 4.1, the conditional distribution $P_{B|\mathbf{X}}$ is unknown. Like RIOAE, we can introduce a new "optimization variable" $Q_{\hat{B}|\mathbf{X}}$ and impose constraint (7).

Then the resulting optimization problem for the PSOAE is

$$\begin{aligned}\inf_g \inf_{Q_{B|\mathbf{X}}} \inf_{Q_{E_0|B,X_0}} & \mathbb{E}_{P_{\mathbf{X}}} \mathbb{E}_{Q_{B|\mathbf{X}}} \mathbb{E}_{Q_{E_0|B,X_0}} d^p(X_0, g(B, E_0)) \\ & + \lambda_1 \mathcal{D}_B(P_B, \textstyle\int_{\mathcal{X}^\infty} Q_{B|\mathbf{X}} dP_{\mathbf{X}}) \\ & + \lambda_2 \mathcal{D}_{E_0}(P_{E_0}, \textstyle\int_{\mathcal{X}\times\mathcal{I}} Q_{E_0|X_0,B} dP_{X_0,B}) \\ & + \lambda_3 \mathcal{I}\mathcal{D}(Q_{B,E_0}),\end{aligned} \tag{15}$$

for appropriate choices of divergences $\mathcal{D}_B$ and $\mathcal{D}_{E_0}$; the final penalty is a measure of deviation from independence for the joint distribution of B and E_0 induced by Q_{X_0,B,E_0}.

Remark 4.1. *The characterization of $\mathcal{D}_{\mathrm{OAE}}$ (5) in Theorem 4.1 is significant, since equations (13) through (15) show that the contributions of the "identity" latent variable B and "variation" latent variable E are made explicitly independent. This contrasts to the RIOAE based on $\mathcal{D}_{\mathrm{RIOAE}}$ (10) in which these two variables are not completely orthogonal. The implication is that during the RIOAE training the B may absorb some of the variations in the E, causing an instability (see Section 4.1). Thus the value of the novel bound $\mathcal{D}_{\mathrm{OAE}}$ is not only limited to the tightness over $\mathcal{D}_{\mathrm{RIOAE}}$ (10). It is not clear that such a characterization is also possible with $\mathcal{D}_{\mathrm{RIOAE}}$.*

Algorithm 1 Product-space OAE training

Input: Exchangeable sequences $(x_1^i, ..., x_{n_i}^i)$ for $i = 1, ..., L$
Output: Encoder pair $(Q_{B|X_0}, Q_{E_0|B,X_0})$ and decoder g
Require: P_B, P_{E_0}, regularization coefficients λ_1, λ_2 and λ_3, positive definite kernels ι, ϑ

while $Q_{B|X_0}$, $Q_{E_0|B,X_0}$, f, g not converge **do**
  while $Q_{B|X_0}$, g not converged **do**
    Sample units $i = 1, ..., n$ and sequence $(x_1^i, ..., x_m^i)$ for each unit i
    Sample b^i from P_B and $(e_1^i, ..., e_m^i)$ from P_{E_0} for all $i = 1, ..., n$
    Sample $\hat{b}_j^i \sim Q_{B|X_0}(\cdot|x_j^i)$ for each $j = 1, ..., m$ and aggregate $\hat{b}^i = \frac{1}{m}\sum_{j=1}^m \hat{b}_j^i$ for $i = 1, ..., n$.
    Sample $(\hat{e}_1^i, ..., \hat{e}_m^i)$ from $Q_{E_0|B,X_0}$ given $\hat{b}^i$ and $(x_1^i, ..., x_m^i)$ for all $i = 1, ..., n$
    Update $Q_{B|X_0}$ and g by descending:

$$\frac{1}{nm}\sum_{i=1}^n\sum_{j=1}^m d^p(x_j^i, g(\hat{b}^i, \hat{e}_j^i)) + \frac{\lambda_1}{n(n-1)}\sum_{i\neq l}\kappa(b^i, b^l) + \frac{\lambda_1}{n(n-1)}\sum_{i\neq l}\kappa(\hat{b}^i, \hat{b}^l) - \frac{2\lambda_1}{n^2}\sum_{i,l}\kappa(b^i, \hat{b}^l) + \frac{\lambda_3}{(nm)^2}\sum_{i,j}^n\sum_{q,r}^m \iota(\hat{b}_q^i, \hat{b}_r^j)\vartheta(\hat{e}_q^i, \hat{e}_r^j) + \frac{\lambda_3}{(nm)^4}\sum_{i,j,k,l}^n\sum_{q,r,v,w}^m \iota(\hat{b}_q^i, \hat{b}_r^j)\vartheta(\hat{e}_v^k, \hat{e}_w^l) - \frac{\lambda_3}{(nm)^3}\sum_{i,j,k}^n\sum_{q,r,v}^m \iota(\hat{b}_q^i, \hat{b}_r^j)\vartheta(\hat{e}_q^i, \hat{e}_v^k)$$

  end while
  while $Q_{E_0|B,X_0}$, f, g not converged **do**
    Repeat 3 - 6
    Update $Q_{E_0|B,X_0}$ and g by descending:

$$\frac{1}{nm}\sum_{i=1}^n\sum_{j=1}^m d^p(x_j^i, g(\hat{b}^i, \hat{e}_j^i)) - \frac{\lambda_2}{nm}\sum_{i=1}^n\sum_{j=1}^m \log f(\hat{e}_j^i) + \frac{\lambda_3}{(nm)^2}\sum_{i,j}^n\sum_{q,r}^m \iota(\hat{b}_q^i, \hat{b}_r^j)\vartheta(\hat{e}_q^i, \hat{e}_r^j) + \frac{\lambda_3}{(nm)^4}\sum_{i,j,k,l}^n\sum_{q,r,v,w}^m \iota(\hat{b}_q^i, \hat{b}_r^j)\vartheta(\hat{e}_v^k, \hat{e}_w^l) - \frac{\lambda_3}{(nm)^3}\sum_{i,j,k}^n\sum_{q,r,v}^m \iota(\hat{b}_q^i, \hat{b}_r^j)\vartheta(\hat{e}_q^i, \hat{e}_v^k)$$

    Update f by ascending: $\sum_{i,j}\log f(e_j^i) + \log(1 - f(\tilde{e}_j^i))$
  end while
end while

4.3. Training the product-space OAE

Alternating optimization. Empirically, the following coordinate-descent type alternating optimization scheme is effective in training the PSOAE: 1) Fix the parameters of the "within-unit variation encoder" $Q_{E_0|B,X_0}$, and update the parameters of the "identity encoder" $Q_{B|\mathbf{X}}$ and decoder g until the infimand of problem (15) no longer changes. 2) Fix the parameters of the identity encoder and update the parameters of $Q_{E_0|B,X_0}$ and decoder g until the infimand of problem (15) no longer changes. 3) Repeat steps 1 and 2 until the parameters of the encoder pair $(Q_{B|\mathbf{X}}, Q_{E_0|B,X_0})$ and decoder g converge. Similar to the RIOAE, we use $\mathcal{D}_{\mathrm{GAN}}$ for $\mathcal{D}_{E_0}$ and $\mathcal{D}_{\mathrm{MMD},\kappa}$ for $\mathcal{D}_B$. This choice of divergences reflects our experience with the WAE-GAN for VGGFace2 and that the number of units is usually smaller than non-nested cases.Sampling from $Q_{\hat{B}|\mathbf{X}}$ follows the approach of RIOAE (Algorithm i, Line 6). We also sample $\hat{e}_j^i$ independently from $Q_{E_0|B,X_0}(\cdot|\hat{b}^i, x_j^i)$ given the $\hat{b}^i$ and the observations x_j^i of unit i, for $j = 1, \ldots, m_i$. For the penalty $\mathcal{ID}$, we employ the Hilbert-Schmidt Independence Criterion (Gretton et al., 2005):

$$\mathrm{HSIC}_{\iota,\vartheta}(Q_{B,E_0}) = \left\|\mathrm{E}_{Q_{B,E_0}}\left(\left[\iota(\cdot, B) - \mathrm{E}_{P_B}\iota(\cdot, B)\right] \otimes \left[\vartheta(\cdot, E_0) - \mathrm{E}_{P_{E_0}}\vartheta(\cdot, E_0)\right]\right)\right\|_{\mathrm{HS}}^2,$$

for reproducing kernels $\iota : \mathcal{I} \times \mathcal{I} \to \mathbb{R}$ and $\vartheta : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ that respectively induce Hilbert spaces $\mathcal{H}_\iota$ and $\mathcal{H}_\vartheta$; $\otimes$ denotes the tensor product and $\|C\|_{\mathrm{HS}}^2$ is the squared Hilbert-Schmidt norm of the cross-covariance operator C. This independence criterion is effective especially when the dataset has a small number of samples per each unit. The resulting training procedure is summarized as Algorithm 1.

Initialization. In practice, random variable B only deconvolves the output process $\mathbf{Y}$ but not necessarily the input $\mathbf{X}$. Ideally, the latent variable B or its copy $\hat{B}$ should encode the "identity" of the realizations of an observational unit and make the coordinates of input $\mathbf{X}$ conditionally i.i.d. Thus, encoder $Q_{\hat{B}|\mathbf{X}}$ should be some smoothed version of a classifier. Any sensible classifier of the training data can be fit and used as the "initial value" of $Q_{\hat{B}|\mathbf{X}}$. For example, features from the last hidden layer of the pre-trained ResNet classifier (Cao et al., 2018) can be employed for VGGFace2.

5. Empirical results

The performance of the PSOAE with the proposed training method was assessed for the following three tasks. **Exemplar generation**: Given a few observations $(x_j^i)_{j=1}^m$ from a new unit i, sample the "identity variables" $(\hat{b}_j^i)_{j=1}^m$ from $Q_{B|X_0}(\cdot|x_j^i)$ to take an average $\hat{b}^i = \frac{1}{m}\sum_{j=1}^m \hat{b}_j^i$. Draw a "within-unit variation" e_j from P_{E_0}. An exemplar is the reconstruction $g(\hat{b}^i, e_j)$. If $m = 1$, this task is called one-shot exemplar generation. **Style transfer**: From observations $(x_l^k)_{l=1}^L$ of another unit $k \neq i$, sample $\hat{b}^k$ as in exemplar generation. Draw "within-unit variation" $\hat{e}_l^k$ from $Q_{E_0|B,X_0}(\cdot|\hat{b}^k, x_l^k)$. Then, the sequence $(g(\hat{b}^i, \hat{e}_l^k))_{l=1}^L$ transfers the style of unit k to i. If $m = 1$,

this task is called a one-shot style transfer. **New concept generation**: In order to generate a new unit not in the data, sample b^{new} from P_B and sequence (e_j^{new}) from P_{E_0} i.i.d. Then pass $((b^{\text{new}}, e_j^{\text{new}}))$ to the decoder g. In addition, the representation power of the "identity variables" can be considered by the *prototype image*: for unit i, compute $\hat{b}^i$ as in exemplar generation. Then $g(\hat{b}^i, \mu_{E_0})$ is the prototype image of unit i, where μ_{E_0} is the mean of E_0.

For all the experiments, $\mathcal{X} = \mathbb{R}^{d_\mathcal{X}}$ and $\mathcal{Z} = \mathbb{R}^{d_\mathcal{Z}}$ with Euclidean metric $d(x, x') = ||x - x'||_2$. The prior distribution $P_\mathbf{Z}$ of the latent variable $\mathbf{Z}$ follows model (12). The independent standard normal prior $P_{E_0} = \mathcal{N}(0, \mathbf{I}_{d_\mathcal{V}})$ and $P_B = \mathcal{N}(0, \mathbf{I}_{d_\mathcal{I}})$ were set over $\mathcal{I} = \mathbb{R}^{d_\mathcal{I}}$ and $\mathcal{V} = \mathbb{R}^{d_\mathcal{V}}$, respectively. The identity encoder $Q_{B|X_0}$ and the within-unit variation encoder $Q_{E_0|B,X_0}$ were also Gaussian:

$$B^i|\{X_0 = x_j^i\} \overset{\text{iid}}{\sim} \mathcal{N}(\mu_B(x_j^i), \sigma_B^2(x_j^i)\mathbf{I}_{d_\mathcal{I}}),$$
$$E_j^i|\{B = b^i, X_0 = x_j^i\} \overset{\text{iid}}{\sim} \mathcal{N}(\mu_E(x_j^i, b^i), \sigma_E^2(x_j^i, b^i)\mathbf{I}_{d_\mathcal{V}}),$$

with the mean functions $\mu_B : \mathcal{X} \to \mathcal{I}$, $\mu_E : \mathcal{X} \times \mathcal{I} \to \mathcal{V}$ and the variance functions $\sigma_B^2 : \mathcal{X} \to \mathbb{R}_{++}$, $\sigma_E^2 : \mathcal{X} \times \mathcal{I} \to \mathbb{R}_{++}$. For the VGGFace2 experiments, μ_B and σ_B^2 were initialized with a pre-trained classifier (Cao et al., 2018). The μ_E and σ_E were designed to share most of the network to prevent overfitting. Note that, for MNIST, joint optimization worked equally well to the alternating optimization scheme, but the latter was necessary with VGGFace2 for good performance. The optimization was conducted with the ADAM optimizer (Kingma & Ba, 2014). For VGGFace2, it took 3300 epochs (100 iterations per epoch) to declare convergence, where most of significant reductions occurred within the first 700 epochs. Hyperparameters were hand-tuned using the performance on validation datasets. The network architectures was adapted from the WAE at Tolstikhin et al. (2018). The quality of generated sample was compared with WAE, in which observational units are not preserved. Within-unit sample generations were compared with the RIOAE. The quality of samples were also compared with CAAE for units present in the training data by interpreting each unit as a class. Further implementation details are given in the Supplement.

5.1. MNIST

For the MNIST data, randomly selected 40,357 images were used for training, and the rest were used for testing.

Imbalanced MNIST. In order to impose imbalance in the training data, 90% of the training images of digits of 1, 2, and 6 were removed. Generated images and t-SNE maps (van der Maaten & Hinton, 2008) of the within-unit variation in the latent space are shown in Figure 1. In panel B, we can note that PSOAE is superior in matching the prior distribution, from which samples are plotted in translucent blue

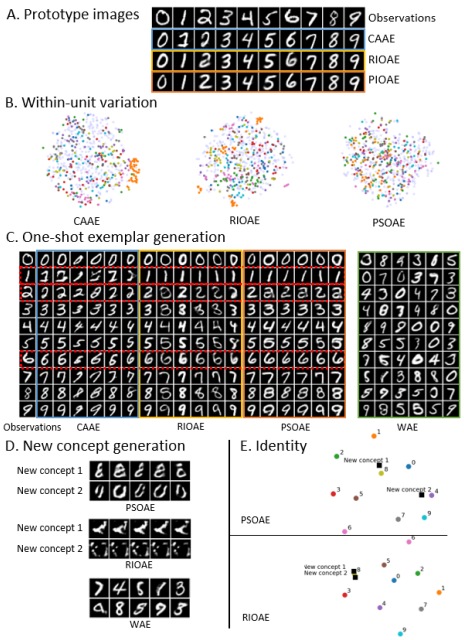

Figure 1. Sample generation from imbalanced MNIST. Panel A: prototype images obtained by CAAE, RIOAE, and PIOAE. Panel C: generated variations of each digit. The minority classes are highlighted in red. Panel D: illustration of new concept generation, samples of "digit-like" units. Panels B and E: t-SNE maps of within-unit variations and identities in the latent space, respectively. Different colors represent different units (digits). Larger figures are provided in the Supplement.

dots as a reference. Note that for CAAE and the RIOAE, the distribution of the encoded within-unit variation shows non-ideal clusters. In particular, the digit 1 (orange dots), which is a minority class, is distinctly clustered. A similar phenomenon can be observed for the digit 2 (green). This is an indicator of training instability of the RIOAE, leading to the poor quality in the prototype images of digits 2 and 6 in panel A. Because of the class imbalance, CAAE also shows poor quality in the prototype images of digit 1 and 2. In panel C, for each method, the first column carries the prototype images from one observation. The rest of columns correspond to newly generated variations of the prototypes. Samples enclosed by dashed red lines represent the minority classes, i.e., the digits 1, 2, and 6. PSOAE shows the best quality of within-unit sample generation for the minority classes CAAE and RIOAE fail to preserve the key features of digit 2 in some variations, while PSOAE maintains those features well. Panel D shows the ability of the model to generate new units (concepts) not present in the training data. While PSOAE successfully generates the new concepts containing key features of handwriting like strokes, RIOAE fails to generate meaningful concepts more than noise. WAE is not designed to generate samples with inborn nested structure. Note that WAE only generates samples on existing digits and does not generalize to any new concept.

overall classes				
	Accuracy	One-shot accuracy	SSIM	Sharpness
WAE	-	-	-	**0.047**(±**0.003**)
CAAE	0.860(±0.008)	-	**0.244**(±**0.020**)	0.041(±0.007)
RIOAE	0.919(±0.033)	0.873(±0.059)	0.292(±0.017)	0.025(±0.004)
PSOAE	**0.939**(±**0.019**)	**0.878**(±**0.029**)	0.263(±0.008)	0.032(±0.008)
Testset	0.994(±0.002)	-	0.229(±0.009)	0.075(±0.004)
minority classes (1,2,6)				
	Accuracy	One-shot accuracy	SSIM	Sharpness
CAAE	0.609 (±0.017)	-	**0.253 (±0.043)**	**0.041 (±0.007)**
RIOAE	0.945 (±0.057)	0.877 (±0.150)	0.403(±0.182)	0.028 (±0.003)
PSOAE	**0.948 (±0.057)**	**0.946 (±0.096)**	0.342(±0.104)	0.032 (±0.010)
Testset	0.992 (±0.004)	-	0.263 (±0.124)	0.077 (±0.006)
majority classes (0,3,4,5,7,8,9)				
	Accuracy	One-shot accuracy	SSIM	Sharpness
CAAE	**0.96 (±0.007)**	-	0.249 (±0.058)	**0.052 (±0.009)**
RIOAE	0.914 (±0.013)	0.853 (±0.073)	0.251 (±0.085)	0.040 (±0.004)
PSOAE	0.918 (±0.025)	**0.875 (±0.047)**	**0.243 (±0.061)**	0.038 (±0.004)
Testset	0.995 (±0.003)	-	0.214 (±0.052)	0.077 (±0.003)

Table 1. Performance metrics on imbalanced MNIST for overall/minority/majority classes, averaged over 10 repetitions. Standard deviations are shown in parentheses. Measures from WAE are omitted in minority/majority classes as WAE was unable to generate specific digits conditionally.

Panel E shows the t-SNE map of the identities (the B) in the latent space. Note that PSOAE closely encodes the similar digits (e.g., digits 3 and 5; 7 and 9). The new concepts presented in Panel D are marked as black squares. Observe that the new concepts generated are mapped close to 8 and 4, where in Panel D they indeed look similar to (but distinct from) digits 8 and 4. Table 1 reports several measures of reconstruction quality on overall/minority/majority classes, averaged over 10 repetitions of 100 exemplar generations. (One-shot) accuracy measures the classification accuracy of an MNIST-trained deep digit classifier for five (one) generated images per digit. The classifier used for calculating the classification accuracy is a CNN with 898k parameters, trained with the common training data used for all generative models. It was tested with test data used for calculating values in Table 1, and trained until its test accuracy became 0.994. Also, the structural similarity (SSIM) (Odena et al., 2017) and the sharpness measured by using the Laplace filter (Tolstikhin et al., 2018) are provided to assess the per-image quality of the generated digits. For overall classes, samples from CAAE and both OAEs exhibit similar performance metrics with the unconditionally generated samples from WAE. However, it may be misleading to focus on a single metric. SSIM and sharpness denotes the variety and crispiness of generated images, not whether they are proper digits. If classification accuracy (first two columns) is low, then those images cannot be recognized as a digit however variable and sharp they are. This is the case with CAAE: while SSIM is close to the testset and sharpness is high, but the generated digit 1, a minority class, does not look like a 1. Noticeably, the finite-number-of-units generation accuracy of CAAE is even lower than the one-shot accuracy of OAEs. When restricted to minority classes, the accuracy of CAAE decreases to 0.609. Between the two OAEs, the PSOAE outperforms by all metrics. The sharpness of WAE was computed unconditionally, where for other models this metric was computed as an average over an equal number of images per digit; WAE virtually did not generate images of the minority/majority classes since the model ignores the class information.

Figure 2. Exemplar generation from MNIST models trained without digit 6. Observation of the unit not used in training (digit 6) and its new variations are enclosed by the dashed red box.

MNIST with a missing digit. To be more informative, we removed the digit 6 completely from the training set and conducted generation using it as an exemplar. Figure 2 clearly shows the superiority of PSOAE. This is a strong support for the value of novel bound $\mathcal{D}_{\text{OAE}}$ (Remark 4.1). The novel optimization problem (15) and Algorithm 1 help the encoder to distinguish the "identity" latent variable B and "variation" latent variable E with a limited number of units (digits).

5.2. VGGFace2

For the VGGFace2 dataset, each face image was cropped and rescaled to a common size of 128 by 128 pixels. The training set consisted of 2,513,512 images from 8,631 randomly chosen people. Two separate test sets were applied to measure the performance. One consisted of the 628,378 images of the 8,631 people used in the training. The other test set was comprised of 169,396 images from 500 people who were not included in the training set. In Figure 3, the proposed method demonstrates the superiority in the identity-preserving quality, without any loss of other image generation qualities (e.g., variety and sharpness). Panel A shows that CAAE fails to preserve the identity of the people *used in the training*. This is likely because of the large number of classes (identities) in the data with severe imbalance. On the other hand, the PSOAE successfully preserves the identities of the input images in the prototypes with more details, regardless of their presence in the model training. In this task, the RIOAE was not as good as PSOAE. Notably, panel B emphasizes the key ability of OAE, generating identity-preserving samples nested within a given unit, where giving original variations (one-shot exemplar generation) or those transferred from other data (style transfer) are both possible. Panel B shows the example of the two tasks. For each example, we generated samples preserving the identity of the target person (1st column) with variations

		IS	FID	Sharpness
K	WAE	-	-	-
	CAAE	2.029($\pm$0.010)	115.767($\pm$2.796)	**0.001**($\pm$**0.000**)
	RIOAE	2.068($\pm$0.011)	107.961($\pm$2.371)	**0.001**($\pm$**0.000**)
	PSOAE	**2.146**($\pm$**0.104**)	**98.525**($\pm$**2.487**)	**0.001**($\pm$**0.000**)
	Testset	3.883($\pm$0.146)	-	0.004($\pm$0.001)
U	WAE	**2.125**($\pm$**0.016**)	106.250($\pm$3.024)	**0.001**($\pm$**0.000**)
	CAAE	-	-	-
	RIOAE	2.067($\pm$0.020)	102.476($\pm$3.363)	**0.001**($\pm$**0.000**)
	PSOAE	**2.125**($\pm$**0.118**)	**94.287**($\pm$**2.323**)	**0.001**($\pm$**0.000**)
	Testset	3.807($\pm$0.164)	-	0.003($\pm$0.001)

Table 2. VGGFace2 performance measures. 'K': identities used in training; 'U': identities not used in training. Standard deviations are provided in parentheses.

given from the source person (2nd to 6th column; one-shot style transfer) and with original variations (7th to 11th column; one-shot exemplar generation). For these examples, PSOAE shows better performance in "identity-preservation" for one-shot exemplar generation and one-shot style-transfer. This superiority suggests the potential of PSOAE as an attractive data augmentation method for many applications that suffer from data imbalance. Remarkably, PSOAE also succeeded in generating completely new identities (neither in the training set nor the test set) with shared pose variations that RIOAE fails (panel C). This failure is due to the interference with E, as B was unable to sufficiently encode identity, as described in Section 4.2. WAE may generate new identities, but they cannot create the systematic variations as OAEs. Class-conditional models like CAAE are clearly incapable of this task. In Table 2, the quality of one-shot exemplar generation is quantified by the inception score (IS) (Salimans et al., 2016), the sharpness of the generative images, and the Frechet inception distance (FID) between the generated images distribution and the original image distributions (Heusel et al., 2017). These measures were averaged over 10 repetitions of 30 exemplar generations. For each unit, 300 samples were generated. For the identity not used in training, the quality of the generated sequence of portraits nested within a single person was compared with the unstructured samples generated from WAE. All measures favor OAE over WAE and CAAE. For FID, PSOAE shows the best result, which implies that the data generated from PSOAE is closer to the original data than RIOAE.

6. Conclusion

OAEs are a promising new family of models for learning disentangled representation when data have a nested structure. The key attraction is their ability to generate observational-unit preserving samples for an unknown number of observational-units and variations within-unit. Especially, when the data exhibit exchangeability, OAEs yield tractable learning algorithms. We have shown that the previous approach to OAE (RIOAE) actually minimizes a loose

Figure 3. Sample generation from VGGFace2. Panel A: prototype images of individuals obtained by CAAE, RIOAE, and PSOAE. Panel B: examples of one-shot style transfer and one-shot example generation, nested within a given target. Panel C: generated new (virtual) individuals (WAE, OAEs) and their variations (RIOAE, PSOAE).Larger figures are provided in the Supplement.

upper bound of Orstein's d-bar distance, an optimal transport distance between stationary processes. Our approach, PSOAE, alleviates the instability of RIOAE by minimizing a tighter upper bound, and successfully separates within-unit and between-unit variations in the representation space. The power of this separation is exhibited with three key tasks on representation learning: exemplar generation, style transfer, and new concept generation, all in which PSOAE shows a favorable performance.

Acknowledgements

This work was supported in part by Samsung Electronics Co., Ltd. through the SNU-Samsung Smart Campus research program. The authors were also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A2C1007126).

References

Bousquet, O., Gelly, S., Tolstikhin, I., Simon-Gabriel, C.-J., and Schölkopf, B. From optimal transport to generative modeling: the VEGAN cookbook. *arXiv preprint arXiv:1705.07642*, 2017.

Cao, Q., Shen, L., Xie, W., Parkhi, O. M., and Zisserman, A. VGGFace2: A dataset for recognising faces across pose and age. In *Proc. 13th IEEE Int. Conf. Automat. Face Gesture Recogn. (FG 2018)*, pp. 67–74, 2018.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Adv. Neural Inf. Process. Syst. (NeurIPS 2016)*, pp. 2172–2180, 2016.

Choi, Y. and Won, J.-H. Ornstein auto-encoders. In *Proc. Int. Joint Conf. Artif. Intell. (IJCAI 2019)*, pp. 2172–2178. AAAI Press, 2019.

Diggle, P. J., Heagerty, P., Heagerty, P. J., Liang, K.-Y., Zeger, S., et al. *Analysis of longitudinal data*. Oxford University Press, Oxford, UK, 2002.

Fitzmaurice, G. M., Laird, N. M., and Ware, J. H. *Applied longitudinal analysis*, volume 998. John Wiley & Sons, New York, USA, 2012.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Adv. Neural Inf. Process. Syst. (NeurIPS 2014)*, pp. 2672–2680, 2014.

Gray, R. M., Neuhoff, D. L., and Shields, P. C. A generalization of Ornstein's $\bar{d}$ distance with applications to information theory. *Ann. Probab.*, pp. 315–328, 1975.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. Measuring statistical dependence with Hilbert-Schmidt norms. In *Int. Conf. Algorithmic Learn. Theory (ALT 2005)*, pp. 63–77. Springer, 2005.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(Mar):723–773, 2012.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Adv. Neural Inf. Process. Syst. (NeurIPS 2017)*, pp. 6626–6637, 2017.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *Proc. Int. Conf. Learn. Represent. (ICLR 2014)*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *Proc. Int. Conf. Learn. Represent. (ICLR 2014)*, 2014.

Kingma, D. P., Rezende, D. J., Mohamed, S., and Welling, M. Semi-supervised learning with deep generative models. In *Adv. Neural Inf. Process. Syst. (NeurIPS 2014)*, pp. 3581–3589, 2014.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. The omniglot challenge: A 3-year progress report. *Curr. Opin. Behav. Sci.*, 29:97–104, 2019.

Lopez, R., Regier, J., Jordan, M. I., and Yosef, N. Information constraints on auto-encoding variational Bayes. In *Adv. Neural Inf. Process. Syst. (NeurIPS 2018)*, pp. 6114–6125, 2018.

Louizos, C., Swersky, K., Li, Y., Welling, M., and Zemel, R. S. The variational fair autoencoder. In *Proc. Int. Conf. Learn. Represent. (ICLR 2016)*, 2016.

Makhzani, A., Shlens, J., Jaitly, N., and Goodfellow, I. Adversarial autoencoders. In *Proc. Int. Conf. Learn. Represent. (ICLR 2016)*, 2016.

Odena, A., Olah, C., and Shlens, J. Conditional image synthesis with auxiliary classifier GANs. In *Proc. Int. Conf. Mach. Learn. (ICML 2017)*, pp. 2642–2651, 2017.

Olshen, R. A note on exchangeable sequences. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 28(4):317–321, 1974.

Ornstein, D. S. An application of ergodic theory to probability theory. *Ann. Probab.*, 1(1):43–58, 1973.

Patrini, G., van den Berg, R., Forre, P., Carioni, M., Bhargav, S., Welling, M., Genewein, T., and Nielsen, F. Sinkhorn autoencoders. In *Uncertain. Artif. Intell. (UAI 2020)*, pp. 733–743, 2020.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training GANs. In *Adv. Neural Inf. Process. Syst. (NeurIPS 2016)*, pp. 2234–2242, 2016.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schölkopf, B. Wasserstein auto-encoders. In *Proc. Int. Conf. Learn. Represent. (ICLR 2018)*, 2018.

van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9(Nov):2579–2605, 2008.

Zhao, S., Song, J., and Ermon, S. Learning hierarchical features from deep generative models. In *Proc. Int. Conf. Mach. Learn. (ICML 2017)*, pp. 4091–4099, 2017.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. IEEE Int. Conf. Comput. Vision (ICCV 2017)*, pp. 2223–2232, 2017.