Supplementary Material for: Learning from Nested Data with Ornstein Auto-Encoders

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

1. Proofs

Proof Theorem 3.1. We first need to check if $\mathcal{Q}$ is not empty. Consider a distribution $Q_{Z_0,X_0|B}$ having marginals $P_{Z_0|B}$ and $P_{X_0|B}$. A trivial example of such a conditional distribution is $P_{Z_0|B}P_{X_0|B}$. Then, we can find a conditional distribution $Q_{Z_0|X_0,B}$ satisfying that $Q_{Z_0|X_0,B}P_{X_0|B} = Q_{Z_0,X_0|B}$, so that $\int_{\mathcal{X}} Q_{Z_0|X_0,B}dP_{X_0|B} = P_{Z_0|B}$. With this distribution $Q_{Z_0|X_0,B}$, we can construct a conditional distribution $Q_{\mathbf{Z}|\mathbf{X},B}$ such that the joint distribution $Q_{\mathbf{Z}|\mathbf{X},B}P_{\mathbf{X},B}$ of $(\mathbf{X}, \mathbf{Z}, B)$ has a marginal $\big[\prod_{j=1}^{n} Q_{Z_0|X_0,B}\big]P_{\mathbf{X}_{1:n},B}$ on $(\mathbf{X}_{1:n}, \mathbf{Z}_{1:n}, B)$ for any n. Then, its marginal on $\mathbf{Z}_{1:n}$ is

$$
\begin{aligned}
&\int_{\mathcal{X}^n\times\mathcal{B}} \big[\prod_{j=1}^{n} Q_{Z_0|X_0,B}\big] dP_{\mathbf{X}_{1:n},B} \\
&= \int_{\mathcal{X}^n\times\mathcal{B}} \big[\prod_{j=1}^{n} Q_{Z_0|X_0,B}\big] d\big[P_B \prod_{j=1}^{n} P_{X_0|B}\big] \\
&= \int_{\mathcal{B}} \prod_{j=1}^{n} \big[\int_{\mathcal{X}} Q_{Z_0|X_0,B}dP_{X_0|B}\big] dP_B \\
&= \int_{\mathcal{B}} \big[\prod_{j=1}^{n} P_{Z_0|B}\big] dP_B \\
&= P_{\mathbf{Z}_{1:n}},
\end{aligned}
$$

for any n. From the Kolmogorov extension theorem, it follows that $Q_{\mathbf{Z}|\mathbf{X},B} \in \mathcal{Q}$. Thus $\mathcal{Q}$ is not empty.

Recall that $\mathcal{P}_s(P_{\mathbf{X}}, P_{\mathbf{Z}})$ is the set of all jointly stationary distributions of $(\mathbf{X}, \mathbf{Z}) \in \mathcal{X}^\infty \times \mathcal{Z}^\infty$ having $P_{\mathbf{X}}$ and $P_{\mathbf{Z}}$ as marginals. Let $\mathcal{P}_{\mathbf{X},\mathbf{Z},B}$ be the set of all joint distributions $Q_{\mathbf{Z}|\mathbf{X},B}P_{\mathbf{X},B}$ of $(\mathbf{X}, \mathbf{Z}, B) \in \mathcal{X}^\infty \times \mathcal{Z}^\infty \times \mathcal{B}$ for any $Q_{\mathbf{Z}|\mathbf{X},B} \in \mathcal{Q}$, and $\mathcal{P}_{\mathbf{X},\mathbf{Z}}$ be the set of marginal distributions on $(\mathbf{X}, \mathbf{Z})$ induced by $\mathcal{P}_{\mathbf{X},\mathbf{Z},B}$. For any $\pi_{\mathbf{X},\mathbf{Z}} \in \mathcal{P}_{\mathbf{X},\mathbf{Z}}$, there exists $\pi_{\mathbf{X},\mathbf{Z},B} \in P_{\mathbf{X},\mathbf{Z},B}$ such that its marginal is $\pi_{\mathbf{X},\mathbf{Z}}$. Since any joint distribution in $\mathcal{P}_{\mathbf{X},\mathbf{Z},B}$ has marginals $P_{\mathbf{X}}$ and $P_{\mathbf{Z}}$, it follows that $\pi_{\mathbf{X},\mathbf{Z}}$ has marginals $P_{\mathbf{X}}$ and $P_{\mathbf{Z}}$. Also, $\pi_{\mathbf{X},\mathbf{Z},B}$ has a marginal $\big[\prod_{j=1}^{n} Q_{Z_0|X_0,B}\big]P_{\mathbf{X}_{1:n},B} = \big[\prod_{j=1}^{n} P_{X_0|B}Q_{Z_0|X_0,B}\big]P_B$ on $\mathcal{X}^n \times \mathcal{Z}^n \times \mathcal{B}$ for some $Q_{Z_0|X_0,B}$ and for any n. This means that $\pi_{\mathbf{X},\mathbf{Z}}$ is jointly exchangeable i.e., $\{(X_j, Y_j)\}_{j=-\infty}^{j=\infty}$ is exchangeable, thus stationary. Then, $\pi_{\mathbf{X},\mathbf{Z}} \in \mathcal{P}_s(\mathbf{X}, \mathbf{Z})$, so that $\mathcal{P}_s(\mathbf{X}, \mathbf{Z}) \supset \mathcal{P}_{\mathbf{X},\mathbf{Z}}$. Now starting from (3), we can conclude

[*]Equal contribution [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 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

that,

$$\begin{aligned}
\bar{\rho}(P_{\mathbf{X}}, \mathbf{g}_{\sharp}P_{\mathbf{Z}}) = &\inf_{Q_{\mathbf{Z}|\mathbf{X}} \in \mathcal{Q}_{\mathbf{Z}|\mathbf{X}}} \mathbb{E}_{P_{\mathbf{X}}} \mathbb{E}_{Q_{\mathbf{Z}|\mathbf{X}}} d^p(X_0, [\mathbf{g}(\mathbf{Z})]_0) \\
&= \inf_{\pi_{\mathbf{X},\mathbf{Z}} \in \mathcal{P}_s(P_{\mathbf{X}}, P_{\mathbf{Z}})} \mathbb{E}_{\pi_{\mathbf{X},\mathbf{Z}}} d^p(X_0, [\mathbf{g}(\mathbf{Z})]_0) \\
&\leq \inf_{\pi_{\mathbf{X},\mathbf{Z}} \in \mathcal{P}_{\mathbf{X},\mathbf{Z}}} \mathbb{E}_{\pi_{\mathbf{X},\mathbf{Z}}} d^p(X_0, [\mathbf{g}(\mathbf{Z})]_0) \\
&= \inf_{\pi_{\mathbf{X},\mathbf{Z},B} \in \mathcal{P}_{\mathbf{X},\mathbf{Z},B}} \mathbb{E}_{\pi_{\mathbf{X},\mathbf{Z},B}} d^p(X_0, [\mathbf{g}(\mathbf{Z})]_0) \\
&= \inf_{Q_{\mathbf{Z}|\mathbf{X},B} \in \mathcal{Q}} \mathbb{E}_{P_{\mathbf{X},B}} \mathbb{E}_{Q_{\mathbf{Z}|\mathbf{X},B}} d^p(X_0, [\mathbf{g}(\mathbf{Z})]_0).
\end{aligned}$$

□

Proof of Proposition 3.1. For any conditional distributions $Q_{\mathbf{Z}|\mathbf{X},B} \in \mathcal{Q}^{RI}$, the joint distribution $P_{\mathbf{X},B}Q_{\mathbf{Z}|\mathbf{X},B}$ has a marginal

$$\begin{aligned}
&\int_{\mathcal{X}^n \times \mathcal{Z}} \big[\prod_{j=1}^n Q_{Z_0|X_0,B}\big] dP_{\mathbf{X}_{1:n},B} \\
&= \int_{\mathcal{X}^n \times \mathcal{Z}} \big[\prod_{j=1}^n Q_{Z_0|X_0,B}\big] d\big[P_B \prod_{j=1}^n P_{X_0|B}\big] \\
&= \int_{\mathcal{Z}} \prod_{j=1}^n \big[\int_{\mathcal{X}} Q_{Z_0|X_0,B} dP_{X_0|B}\big] dP_B \\
&= \int_{\mathcal{Z}} \big[\prod_{j=1}^n P_{Z_0|B}\big] dP_B \\
&= P_{\mathbf{Z}_{1:n}},
\end{aligned}$$

on $\mathbf{Z}_{1:n}$ for any n. The third equality holds because $Q_{Z_0|X_0,B}$, which promotes $Q_{\mathbf{Z}|\mathbf{X},B}$, achieves the constraint (9). Thus $Q_{\mathbf{Z}|\mathbf{X},B} \in \mathcal{Q}$ and we have $\mathcal{Q}^{RI} \subset \mathcal{Q}$. □

Proof of Theorem 4.1. Note that $Q_{Z_0|X_0,B}$ fully parameterizes $\mathcal{Q}$, and under model (12), we have $Q_{Z_0|X_0,B} = Q_{B,E_0|X_0,B} = Q_{B|X_0,B}Q_{E_0|X_0,B}$, where $Q_{B|X_0,B}(\cdot|x_0,b)$ is the Dirac measure on b for any $b \in \mathcal{I}$. From the construction (12), for $b \in \mathcal{I}$ and $e_j \in \mathcal{V}$, $j = 1, ..., n$,

$$\begin{aligned}
&P_{\mathbf{Z}_{\mathbf{1:n}}}((b, e_1), (b, e_2), ..., (b, e_n)) \\
&= \int_{\mathcal{I}} \big[\prod_{j=1}^n P_{Z_0|B}((b, e_j)|b')\big] dP_B(b') \\
&= \int_{\mathcal{I}} \big[\prod_{j=1}^n P_{B,E_0|B}((b, e_j)|b')\big] dP_B(b') \\
&= \int_{\mathcal{I}} \big[\prod_{j=1}^n P_{E_0|B}(e_j|b')P_{B|B}(b|b')\big] dP_B(b') \\
&= \big[\prod_{j=1}^n P_{E_0|B}(e_j|b)\big] P_B(b)
\end{aligned}$$

for any n. Also, for any $((b, e_1), (b, e_2), ..., (b, e_n)) \in \mathcal{I} \times \mathcal{V}^n$, $\int \big[\prod_{j=1}^n Q_{Z_0|X_0,B}\big] dP_{\mathbf{X}_{1:n},B}$ can be formulated as

$$P_{\mathbf{Z}_{\mathbf{1:n}}}((b, e_1), (b, e_2), ..., (b, e_n))$$

$$
\begin{aligned}
&= \int_{\mathcal{X}^n \times \mathcal{I}} \big[\prod_{j=1}^{n} Q_{Z_0|X_0,B}((b', e_j)|x_j, b)\big] dP_{\mathbf{X}_{1:n},B}(x_1, \cdots, x_n, b') \\
&= \int_{\mathcal{I}} \int_{\mathcal{X}^n} \big[\prod_{j=1}^{n} Q_{Z_0|X_0,B}((b', e_j)|x_j, b)\big] dP_{\mathbf{X}_{1:n}|B}(x_1, ..., x_n|b')\, dP_B(b') \\
&= \int_{\mathcal{I}} \left[\prod_{j=1}^{n} \int_{\mathcal{X}} Q_{Z_0|X_0,B}((b', e_j)|x_j, b) dP_{X_0|B}(x_j|b')\right] dP_B(b') \\
&= \int_{\mathcal{I}} \left[\prod_{j=1}^{n} \int_{\mathcal{X}} Q_{B|X_0,B}(b'|x_j, b) Q_{E_0|X_0,B}(e_j|x_j, b) dP_{X_0|B}(x_j|b') P_{X_0|B}(x_j|b')\right] dP_B(b') \\
&= \left[\prod_{j=1}^{n} \int_{\mathcal{X}} Q_{E_0|X_0,B}(e_j|x_j, b) dP_{X_0|B}(x_j|b)\right] P_B(b),
\end{aligned}
$$

where $Q_{B|X_0,B}(\cdot|x_0, b)$ is the Dirac measure on b for any $b \in \mathcal{I}$. for all n, since $Q_{B|X_0,B}$ is the Dirac measure on b. Thus

$$
\int_{\mathcal{X}} Q_{E_0|X_0,B} dP_{X_0|B} = P_{E_0} \quad \text{a.s.} \tag{i}
$$

and $Q_{E_0|X_0,B}$ fully parameterizes $\mathcal{Q}$ under the constraint (i) and that B satisfies $P_{\mathbf{X_{1:n}},B} = \big[\prod_{j=1}^{n} P_{X_0|B}\big] P_B$ for all n. Thus, $D_{\text{OAE}}(P_{\mathbf{X}}, P_{\mathbf{Y}}; P_{\mathbf{Z}})$ has the formulation in terms of a encoder $Q_{E_0|X_0,B}$:

$$
\begin{aligned}
&\mathcal{D}_{\text{OAE}}(P_{\mathbf{X}}, P_{\mathbf{Y}}; P_{\mathbf{Z}}) \\
&= \inf_{Q_{\mathbf{Z}|\mathbf{X},B} \in \mathcal{Q}} \mathrm{E}_{P_{\mathbf{X},B}} \mathrm{E}_{Q_{\mathbf{Z}|\mathbf{X},B}} \mathrm{E}_{Q_{\mathbf{Y}|\mathbf{Z}}} d^p(X_0, Y_0), \\
&= \inf_{Q_{E_0|X_0,B} \in \mathcal{Q}_{E_0}} \mathrm{E}_{P_{\mathbf{X}}} \mathrm{E}_{P_{B|\mathbf{X}}} \mathrm{E}_{Q_{E_0|X_0,B}} d^p(X_0, g(B, E_0)),
\end{aligned}
$$

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

2. Implementation details

Conditional adversarial auto-encoders When all the observational units are present in the training data, we compared the quality of samples from the PSOAE with CAAE as well, by interpreting each unit as a class. We set the CAAE with conditional Gaussian latent variables:

$$
Z_j^i | \{Y^i = k\} \overset{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{I}_{d_{\mathcal{Z}}}),
$$

for $k = 1, \ldots, C$ where Y is a given class label of X_0 and C is the number of subjects. For each class $k = 1, \ldots, C$, the encoder $Q_{Z_0|Y X_0}$ of CAAE were designed to be a Gaussian encoder:

$$
Z_j^i | \{X_0 = x_j^i, Y = k\} \sim \mathcal{N}\big(\mu_k(x^i, y^i), \sigma_k^2(x^i, y^i)\mathbf{I}_{d_{\mathcal{Z}}}\big),
$$

where $\mu_k : \mathcal{X} \times \{1, \ldots, C\} \to \mathcal{Z}$, $\sigma_k^2 : \mathcal{X} \times \{1, \ldots, C\} \to \mathbb{R}_{++}$ are parameterized by a deep neural network. The decoder $g : \mathcal{Z} \times \{1, \ldots, C\} \to \mathcal{X}$ was also parameterized by a deep neural network.

Random-intercept OAE For the RIOAE, we followed the recipe in (Choi & Won, 2019). The prior distribution P_{Z_0} was a random intercept model with Gaussian noise:

$$
Z_j^i | \{B^i = b^i\} \overset{\text{iid}}{\sim} \mathcal{N}(b^i, \mathbf{I}_{d_{\mathcal{Z}}}), \quad B^i \overset{\text{iid}}{\sim} \mathcal{N}(0, 100\mathbf{I}_{d_{\mathcal{Z}}}),
$$

and the encoder pair $(Q_{Z_0|B,X_0}, Q_{B|X_0})$ were designed to be a random-intercept Gaussian encoder pair:

$$
Z_j^i | \{B = \tilde{b}^i, X_0 = x_j^i\} \overset{\text{iid}}{\sim} \mathcal{N}(\mu(x_j^i) + \tilde{b}^i, \sigma^2(x_j^i)\mathbf{I}), \quad B | \{X_0 = x_j^i\} \overset{\text{iid}}{\sim} \mathcal{N}(\nu(x_j^i), \tau^2\mathbf{I}),
$$

where the mean functions $\mu : \mathcal{X} \to \mathcal{Z}$, $\nu : \mathcal{X} \to \mathcal{Z}$, the variance function $\sigma^2 : \mathcal{X} \to \mathbb{R}_{++}$, and the decoder $g : \mathcal{Z} \to \mathcal{X}$ were parameterized by deep neural networks. The hyperparameter τ was kept small.

Algorithm i Random-intercept OAE training

```
Input: Exchangeable sequences $(x^i_1, ..., x^i_{n_i})$ for $i = 1, ..., L$
Output: Encoder pair $(Q_{B|X_0}, Q_{Z_0|X_0,B})$ and decoder $g$
Require: $P_B, P_{Z_0|B}$, regularization coefficients $\lambda_1, \lambda_2$, positive definite kernel $\kappa$
Initialize: parameters of $(Q_{Z_0|X_0,B}, Q_{B|X_0})$, $g$, and discriminator $f$
while $Q_{B|X_0}$, $Q_{Z_0|X_0,B}$, $f$, $g$ not converged do
    Sample observational unit $i = 1, \ldots, n$ and sequence $(x^i_1, \ldots, x^i_{m_i})$ for each unit $i$ from the training set
    Sample $b^i$ from $P_B$ for $i = 1, \ldots, n$
    Sample $(z^i_1, \ldots, z^i_{m_i})$ from $P_{Z_0|B}$ given $b^i$ for $i = 1, \ldots, n$
    Sample $\hat{b}^i_j \sim Q_{B|X_0}(\cdot|x^i_j)$ for each $j = 1, \ldots, m_i$ and aggregate $\hat{b}^i = \frac{1}{m_i}\sum_{j=1}^{m_i} \hat{b}^i_j$ for $i = 1, \ldots, n$.
    Sample $(\hat{z}^i_1, \cdots, \hat{z}^i_{m_i})$ from $Q_{Z_0|X_0,B}$ given $\hat{b}^i$ and $(x^i_1, \cdots, x^i_{m_i})$ for $i = 1, ..., n$.
    Update $Q_{Z_0|X_0,B}$, $Q_{B|X_0}$, and $g$ by descending:
        $\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i} d^p(x^i_j, g(\hat{z}^i_j)) - \frac{\lambda_1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}\log f(\hat{z}^i_j)$
        $+\frac{\lambda_2}{n(n-1)}\Big(\sum_{i\neq l}\kappa(b^i, b^l) + \sum_{i\neq l}\kappa(\hat{b}^i, \hat{b}^l)\Big) - \frac{2\lambda_2}{n^2}\sum_{i,l}\kappa(b^i, \hat{b}^l)$
    Update $f$ by ascending: $\sum_{i=1}^{n}\sum_{j=1}^{m_i}\log f(z^i_j) + \log(1 - f(\hat{z}^i_j))$
end while
```

2.1. Architectures

For all the convolutional layers used in the networks, padding and truncated normal initialization were applied.

Imbalanced MNIST Tables i, ii, and iii provide the details of the architecture of the PSOAE. “Batch norm” indicates whether there was a batch normalization layer (Ioffe & Szegedy, 2015). We used sigmoid activation to decode the image, and thus its range was transformed from [0,225] to [0,1]. The network architectures for CAAE, WAE, and the RIOAE were almost the same as the PSOAE except for the output layers of the encoder pair and the input of the decoder; CAAE required additional input for the one-hot encoding of the label information for both the encoder and the decoder. The encoder-decoder architecture had 1M parameters and the discriminator architecture had 17k parameters.

VGGFace2 Tables iv through vii summarize the details of the PSOAE architecture trained for the VGGFace2 data. We transformed image range from [0,225] to [-1,1] and used a hyperbolic tangent activation for the decoder output. For PSOAE, features of the last 2048 dimensional hidden layer from the pre-trained VGGFace2 classifier (Cao et al., 2018) were employed as input to the identity encoder $Q_{B|X_0}$ (Table iv). This pre-trained VGGFace2 classifier (Cao et al., 2018) has an ResNet-50-based architecture (He et al., 2016) with squeeze-and-excitation (SE) blocks (Hu et al., 2018). It was trained with the training set comprised of 8631 identities. Except from the input layer and the last layer, the architectures of encoders and decoders from the cAAE, WAE, and the RIOAE were mostly the same as the within-unit encoder and the decoder from PSOAE, respectively; CAAE required additional input for the one-hot encoding of the label information for both the encoder and the decoder. The encoder-decoder architecture had 19M parameters except CAAE. CAAE had an encoder-decoder architecture with 24M parameters, mainly because of the 8631 number of class information. The discriminator architectures had 855K parameters.

2.2. Training details

The Adam optimizer (Kingma & Ba, 2014) was used to train the model, with $\beta_1 = 0.9$ for updating the first moment estimate and $\beta_2 = 0.999$ for updating the second moment estimate.

Details of the imbalanced MNIST training All models were trained for 10,000 iterations with mini-batch of size 600 with no need of the alternating optimization. We updated the models with the learning rates of 0.001 for the encoder-decoder pair and 0.0005 for the discriminator. Both learning rates were decayed by multiplying $1/1.0001$ after every 100 iterations.

Layer	Operation	Filters	Kernel	Strides	Batch norm	Activation	Linked layer
1	Convolution	64	4x4	2x2	Yes	ReLU	Input
2	Convolution	64	4x4	1x1	Yes	ReLU	1
3	Convolution	128	4x4	2x2	Yes	ReLU	2
4	Convolution	128	4x4	1x1	Yes	ReLU	3
5	Dense	64	-	-	Yes	ReLU	4
6	Dense	32	-	-	Yes	ReLU	5
7	Dense	16	-	-	Yes	ReLU	6
μ_B	Dense	8	-	-	No	Linear	7
σ_B^2	Dense	8	-	-	No	Linear	7
8	Dense	32	-	-	Yes	ReLU	5
μ_E	Dense	8	-	-	No	Linear	8, $B\|X_0$
σ_E^2	Dense	8	-	-	No	Linear	8, $B\|X_0$
Output ($B\|X_0$)	Sample $B\|X_0$	-	-	-	-	-	μ_B, σ_B^2
Output ($E_0\|BX_0$)	Sample $E_0\|BX_0$	-	-	-	-	-	μ_E, σ_E^2

Table i. MNIST encoder pair $(Q_{E_0|BX_0}, Q_{B|X_0})$. $d_{\mathcal{I}} = 8$ and $d_{\mathcal{V}} = 8$
.

Layer	Operation	Filters	Kernel	Strides	Batch norm	Activation	Linked layer
1	Dense	7x7x128	-	-	No	ReLU	Input
2	Reshape to (7,7,128)	-	-	-	-	-	1
3	Transpose Convolution	64	4x4	2x2	Yes	ReLU	2
4	Transpose Convolution	32	4x4	2x2	Yes	ReLU	3
5	Transpose Convolution	16	4x4	1x1	Yes	ReLU	4
Output	Convolution	1	4x4	1x1	No	Sigmoid	5

Table ii. MNIST decoder g

Layer	Operation	Filters	Kernel	Strides	Batch norm	Activation	Linked layer
1	Dense	64	-	-	No	ReLU	Input
2	Dense	64	-	-	No	ReLU	1
3	Dense	64	-	-	No	ReLU	2
4	Dense	64	-	-	No	ReLU	3
5	Dense	64	-	-	No	ReLU	4
Output	Dense	1	-	-	No	Sigmoid	5

Table iii. MNIST discriminator f

Layer	Operation	Filters	Kernel	Strides	Batch norm	Activation	Linked layer
1	Dense	384	-	-	Yes	ReLU	Input
2	Dense	256	-	-	Yes	ReLU	1
μ_B	Dense	64	-	-	No	Linear	2
σ_B^2	Dense	64	-	-	No	Linear	2
Output ($B\|X_0$)	Sample $B\|X_0$	-	-	-	-	-	μ_B, σ_B^2

Table iv. VGGFace2 identity encoder $Q_{B|X_0}$; $d_{\mathcal{I}} = 64$

Layer	Operation	Filters	Kernel	Strides	Batch norm	Activation	Linked layer
1	Convolution	64	5x5	2x2	Yes	ReLU	Input
2	Convolution	128	5x5	2x2	Yes	ReLU	1
3	Convolution	256	5x5	2x2	Yes	ReLU	2
4	Convolution	512	5x5	2x2	Yes	ReLU	3
5	Convolution	256	3x3	2x2	Yes	ReLU	4
6	Dense	128	-	-	Yes	ReLU	5
7	Dense	256	-	-	Yes	ReLU	6, $B\|X_0$
8	Dense	128	-	-	Yes	ReLU	6
μ_E	Dense	128	-	-	No	Linear	8
σ_E^2	Dense	128	-	-	No	Linear	8
Output ($E_0\|BX_0$)	Sample $E_0\|BX_0$	-	-	-	-	-	μ_E, σ_E^2

Table v. VGGFace2 within-unit encoder $Q_{E|BX_0}$; $d_{\mathcal{V}} = 128$

Layer	Operation	Filters	Kernel	Strides	Batch norm	Activation	Linked layer
1	Dense	8x8x512	-	-	Yes	ReLU	Input
2	Reshape to (8,8,512)	-	-	-	-	-	1
3	Transpose Convolution	256	5x5	2x2	Yes	ReLU	2
4	Transpose Convolution	128	5x5	2x2	Yes	ReLU	3
5	Transpose Convolution	64	5x5	2x2	Yes	ReLU	4
6	Transpose Convolution	32	5x5	2x2	Yes	ReLU	5
7	Convolution	32	5x5	1x1	Yes	ReLU	6
Output	Convolution	3	3x3	1x1	No	Hyperbolic tangent	7

Table vi. VGGFace2 decoder g

Layer	Operation	Filters	Kernel	Strides	Batch norm	Activation	Linked layer
1	Dense	512	-	-	No	ReLU	Input
2	Dense	512	-	-	No	ReLU	1
3	Dense	512	-	-	No	ReLU	2
4	Dense	512	-	-	No	ReLU	3
Output	Dense	1	-	-	No	Sigmoid	4

Table vii. VGGFace2 discriminator f

We set $\lambda_1 = 1$, $\lambda_2 = 1$, and $\lambda_3 = 1$. On average, 100 iterations took 6 seconds.

Details of the VGGFace2 training We first trained the model with the alternating optimization: 1) Fix the parameters of the within-unit variation encoder $Q_{E_0|BX_0}$ and train the identity encoder $Q_{B|X_0}$ and decoder g for 10,000 iterations with $\lambda_1 = 100$, $\lambda_2 = 0$, and $\lambda_3 = 1000$; 2) Fix the parameters of the identity encoder $Q_{B|X_0}$ and train the identity encoder $Q_{E_0|BX_0}$ and the decoder g for 10,000 iterations with $\lambda_1 = 0$, $\lambda_2 = 100$, and $\lambda_3 = 1000$. We repeated step 1 and 2 for 15 times, then fine-tuned the model, without the alternating optimization, for 30,000 iterations with $\lambda_1 = 100$, $\lambda_2 = 100$, and $\lambda_3 = 1000$. For total 330,000 iterations, we set the size of the mini-batch to 600 and the learning rates to 0.001 for the encoder-decoder and 0.001 for the discriminator. On average, 100 iterations took 151 seconds.

Computing infrastructure For the imbalanced MNIST dataset, we trained a single model with 5 Intel(R) Xeon(R) CPU Silver 4114 @ 2.20GHz processors and one NVIDIA TITAN V GPUs which had 5120 CUDA cores, 640 tensor cores, and 12GB memory. For the VGGFace2 experiments, we trained a single model with 18 CPU processes and 4 NVIDIA TITAN V GPUs. All the implementations were based on Python 3.6, Tensorflow 1.15.0 and Keras 2.3.1.

3. Additional figures from the imbalanced MNIST experiment

In this section, we show a large version of Figure 1 in Section 5.1 for visual aid.

4. Additional figures from the VGGFace2 experiment

In this section, we first provide a large version of Figure 3 in Section 5.2. Then, we present additional figures from the VGGFace2 experiment in Section 5.2. Figure iii compares quality of sample generation for the people who were *not* used

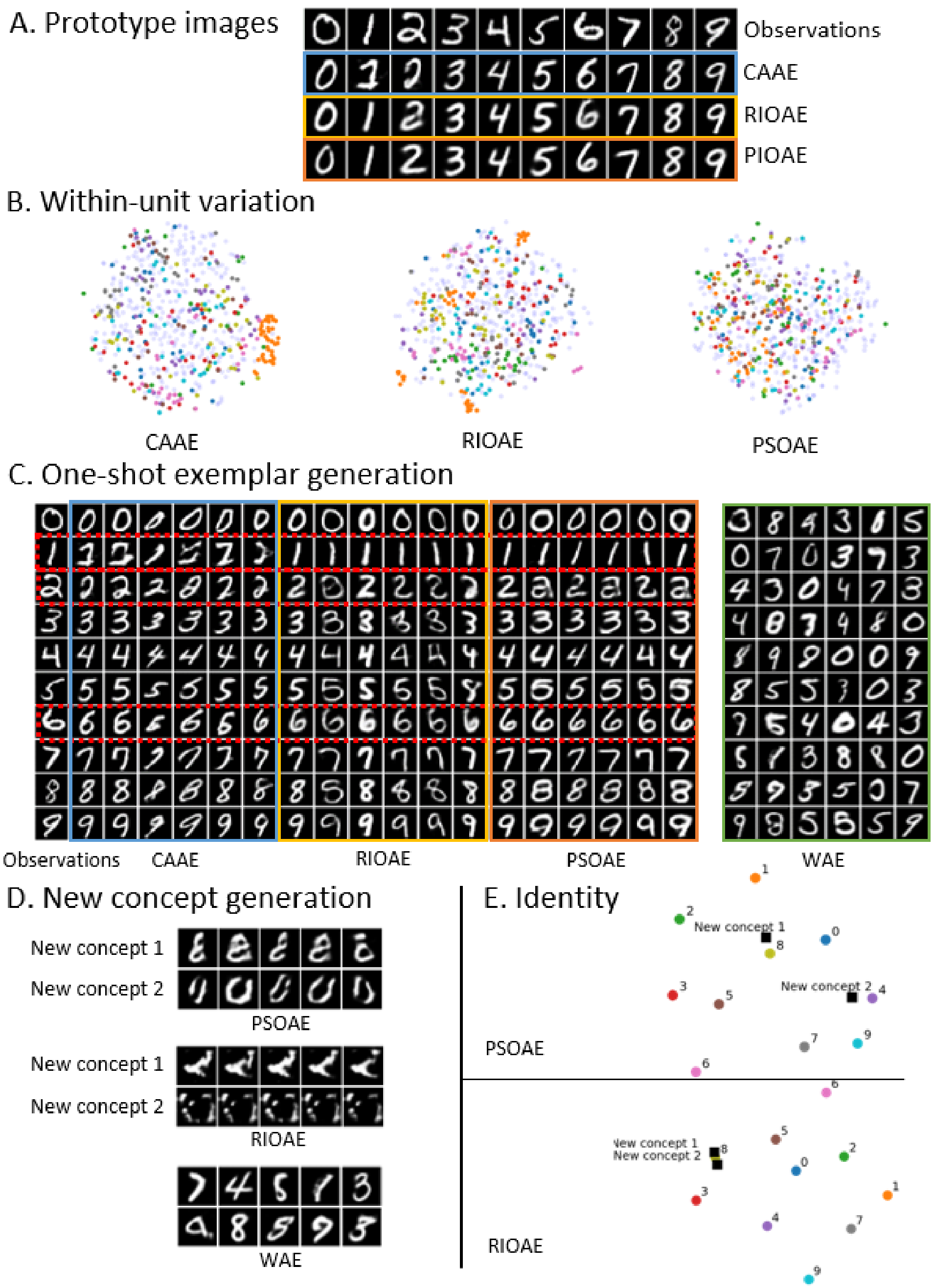

Figure i. Large version of Figure 1

in training. Additional generated samples are shown in panels A and B. Panels C, D, and E show the t-SNE maps of the latent variables, encoded identity variable ($\hat{b}_j^i$ in Algorithm 1), and the encoded within-unit variation ($\hat{e}_j^i$ in Algorithm 1) in the representation (latent) space, respectively. Same person is plotted in same color. The PSOAE shows the best quality in separating observational units in the representation space, while WAE could not provide meaningful separation in the representation. For the PSOAE, the t-SNE map of the encoded identity (panel D) shows better clustering power than the

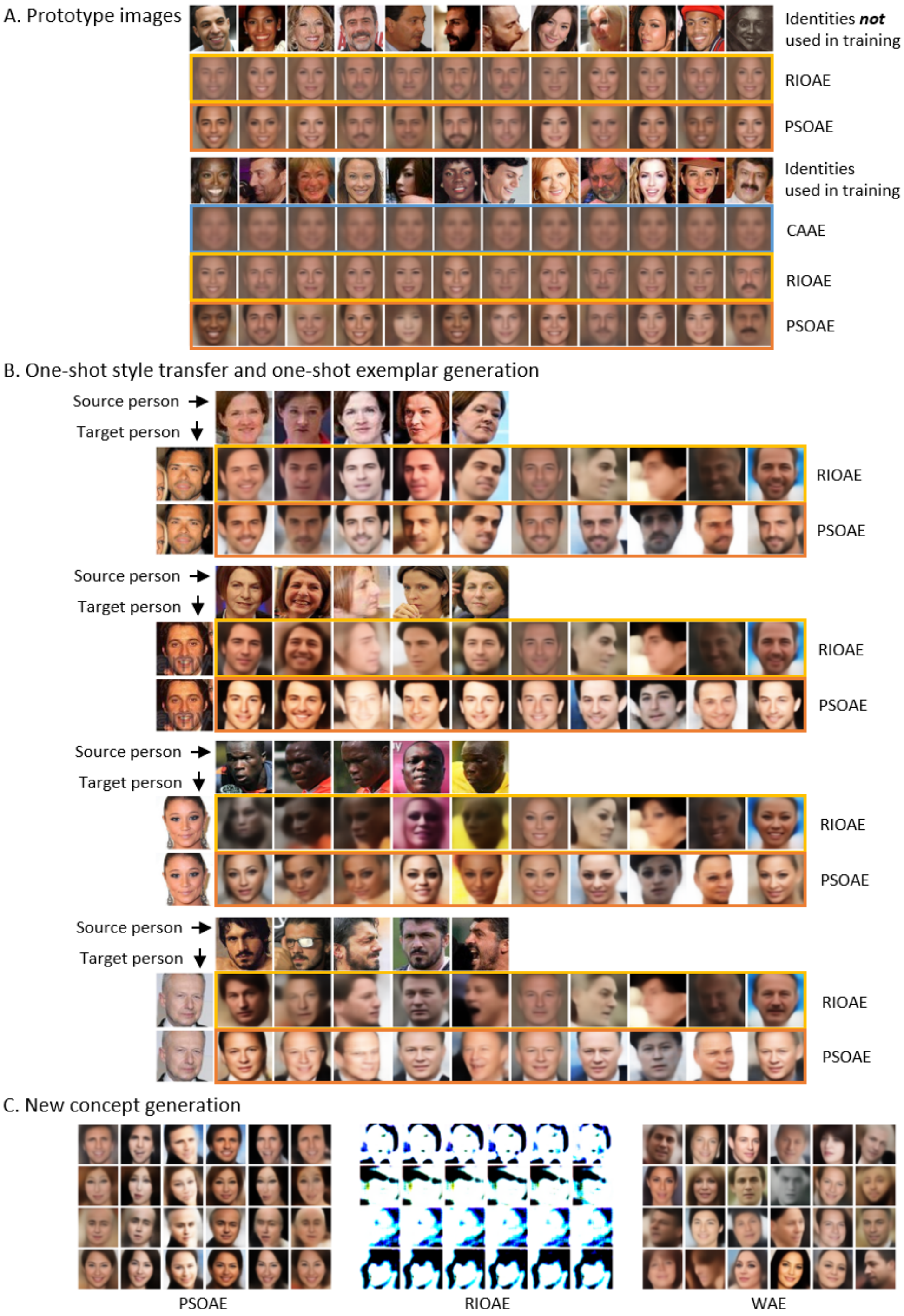

Figure ii. Large version of Figure 3

RIOAE. In panel E, the distribution of the encoded within-unit variation of the PSOAE matches well with the reference samples from the prior distribution, which are plotted in translucent blue dots. Figure iii shows additional generated images and the representations of the people who *were* used in training. Comparing Figure iii with iv, quality of the within-unit face generation of the people not in the training set was similar with that of the people who were in training set, which implies that both OAEs were well generalized. The PSOAE shows superiority in identity-preservation performance for both known and unknown people. Panel B also shows that CAAE failed in preserving the identity of the people *used* in training.

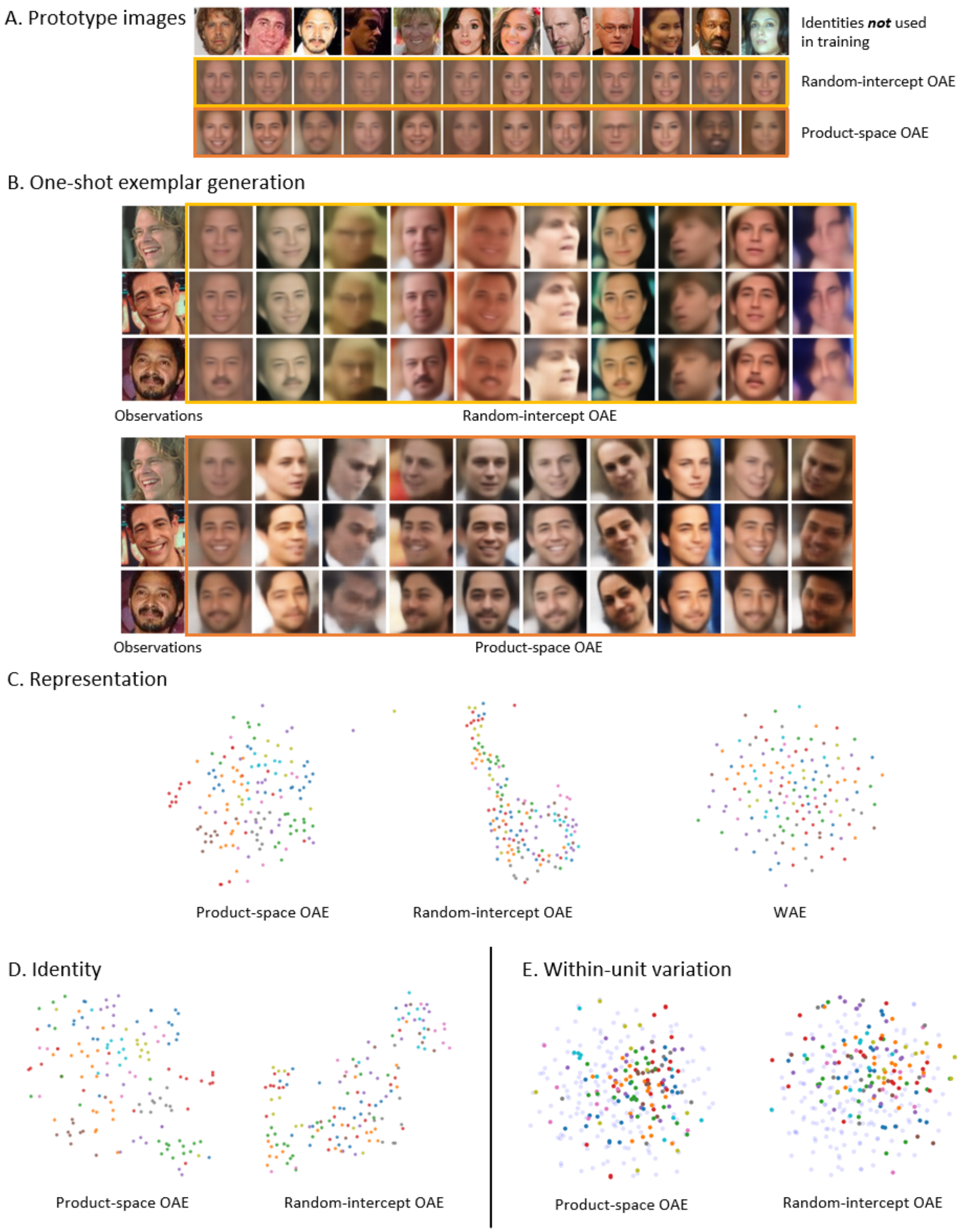

Figure iii. Additional sample generation from VGGFace2 (people not used in training)

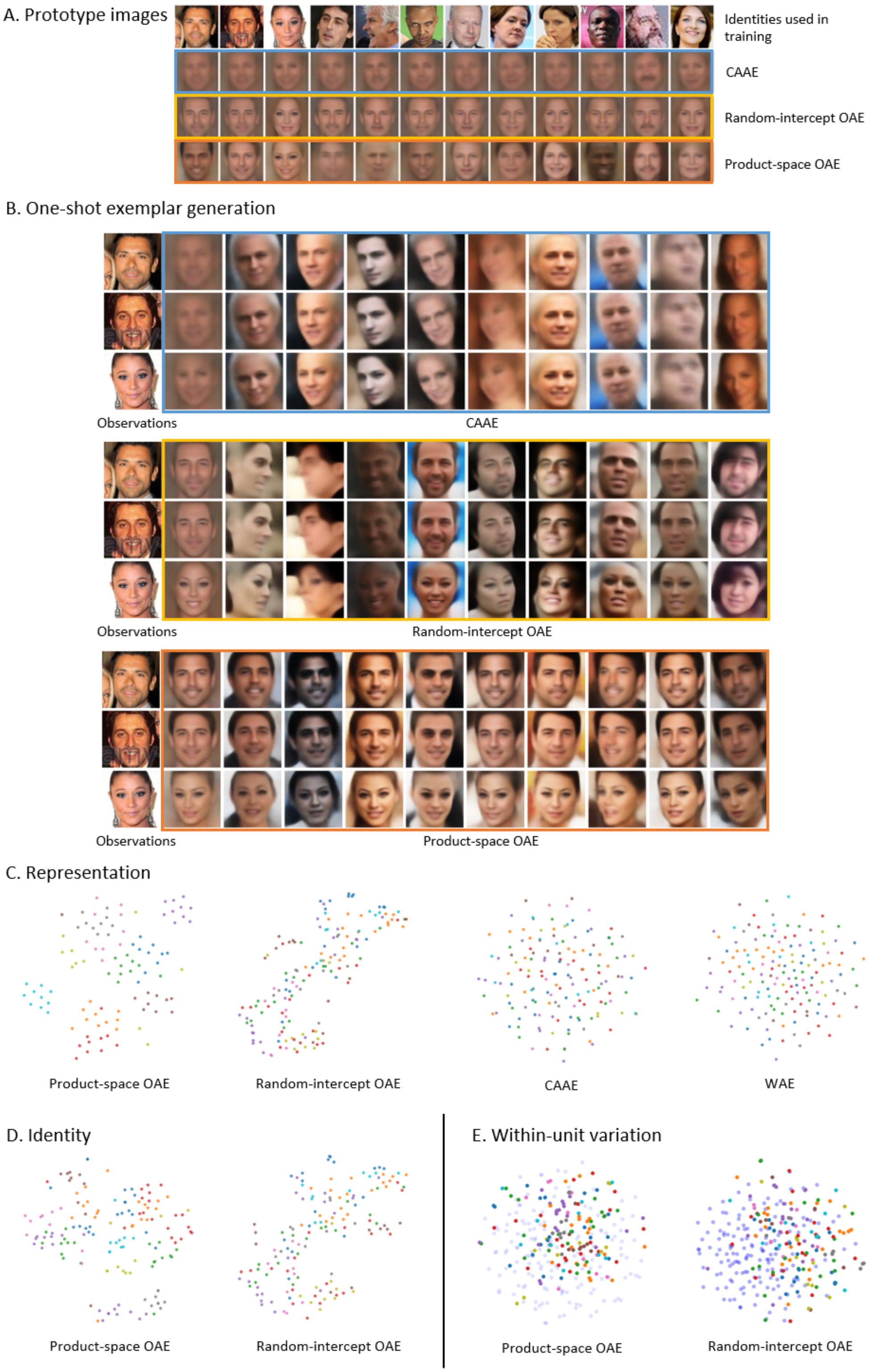

Figure iv. Additional sample generation from VGGFace2 (people used in training)

References

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.

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

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. Int. Conf. Mach. Learn. (ICML 2015)*, volume 37, pp. 448–456. PMLR, 2015.

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