

Article

Variational Deep Alliance: A Generative Auto-Encoding Approach to Longitudinal Data Analysis

Shan Feng , Wenxian Xie and Yufeng Nie 

School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710129, China; wenxianxie@nwpu.edu.cn

* Correspondence: fengshan1912@mail.nwpu.edu.cn (S.F.); yfnie@nwpu.edu.cn (Y.N.)

Abstract

Rapid advancements in the field of deep learning have had a profound impact on a wide range of scientific studies. This paper incorporates the power of deep neural networks to learn complex relationships in longitudinal data. The novel generative approach, Variational Deep Alliance (VaDA), is established, where an “alliance” is formed across repeated measurements via the strength of Variational Auto-Encoder. VaDA models the generating process of longitudinal data with a unified and well-structured latent space, allowing outcomes prediction, subjects clustering and representation learning simultaneously. The integrated model can be inferred efficiently within a stochastic Auto-Encoding Variational Bayes framework, which is scalable to large datasets and can accommodate variables of mixed type. Quantitative comparisons to those baseline methods are considered. VaDA shows high robustness and generalization capability across various synthetic scenarios. Moreover, a longitudinal study based on the well-known CelebFaces Attributes dataset is carried out, where we show its usefulness in detecting meaningful latent clusters and generating high-quality face images.

Keywords: longitudinal data; Variational Auto-Encoder; deep generative model; marginal model; representation learning; clustering

MSC: 68T07; 82M30; 62C10; 62H30; 62M10

1. Introduction

Longitudinal datasets naturally arise in a wide variety of fields and applications, such as social science, biomedicine and agriculture. They contain repeated measurements over multiple time points or spaces, allowing researchers to observe and investigate the temporal and spatial evolution about behaviors and relationships. Different to conventional cross-sectional data, where the sample points are collected independently, the correlated measurements from a subject make longitudinal data analysis a challenging problem. Marginal model [1] focusing on the population-averaged effects is a classical approach for longitudinal data analysis, where consistency inference results can be obtained by solving Generalized Estimating Equations (GEEs) with “working” correlation assumptions [2].

In recent years, rapid advancements in the field of deep learning have had a profound impact in a wide range of scientific studies. The representation learning techniques enable transforming high-dimensional data into their meaningful low-dimensional representations, facilitating a wide variety of downstream tasks hindered by the sheer size and complexity of the original datasets [3]. The Variational Auto-Encoder (VAE) introduced by



Academic Editors: Deniz Genççağ and Rita Singh

Received: 9 December 2025

Revised: 2 January 2026

Accepted: 16 January 2026

Published: 18 January 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Kingma and Welling [4] has been extensively used for probabilistic representation learning. It investigates the underlying deep generative model within an efficient stochastic Auto-Encoding Variational Bayes (AEVB) framework.

In this paper, we develop the Variational Deep Alliance (VaDA), a full deep generative model for longitudinal data analysis, where the data of covariates and responses are generated hierarchically from a unified and well-structured latent space. An important observation about the classical marginal model is that it cannot handle complex non-linear effects of covariates and is often hindered by the high-dimensional feature spaces that are intrinsically sparse [5]. In addition, the GEE approach with “working” correlation assumptions mainly models simple covariance structures of measurements, the misspecification of which can damage the estimation efficiency of marginal effects and further influence the prediction accuracy [1]. These are where we can incorporate the power of deep neural networks to learn complex relationships. Specifically, VaDA forms a deep “alliance” across repeated measurements via the strength of VAE, breaking out of the “working” assumptions from the GEEs based marginal modeling approach. From an overall perspective, the auto-encoding in VaDA proceeds in two directions, i.e., cross-sectionally (with X-encoder and X-decoder) and longitudinally (with Y-encoder and Y-decoder), to learn structured latent embeddings and make accurate predictions by probabilistic decoding.

In addition, the subjects in a longitudinal study can originate from subgroups characterized by distinguishable clusters in the latent space. Modeling such heterogeneity is significant in various situations, such as customer segmentation, precision medicine, etc. In VaDA, clustering of subjects is realized by forging it with a mixture model skeleton. The integrated model can be inferred efficiently within a stochastic AEVB framework, which is scalable to large datasets and can accommodate covariates and responses of mixed type. We use synthetic datasets to demonstrate the robustness and generalization capability of VaDA in prediction and clustering. A longitudinal study based on the well-known CelebFaces Attributes (CelebA) dataset [6] in computer vision is carried out, where we show its usefulness in detecting meaningful latent clusters and generating high-quality face images.

2. Related Works

2.1. Classical Approaches for Longitudinal Data Analysis

Besides the marginal model, the conditional model [7,8] and the mixed effects model [9,10] are two alternative classical approaches for longitudinal data analysis. The conditional model assumes directly a dependence structure between components of repeated measurements and models the joint probability as a multiplication of several transition probabilities. In the mixed effects model, the random subject-specific effects are introduced to induce the dependence relationship between measurements. Generally, these classical approaches are not accessible immediately to high-dimensional datasets. Dimension reduction or feature selection procedures should be conducted for data preprocessing. Moreover, these methods mainly tackle the linear effects of covariates. In [11,12], feature transformations were examined to optimize the non-linear regression relationships.

2.2. Deep Learning-Based Longitudinal Data Analysis

Following the efforts of VAE, Sohn et al. [13] proposed the deep conditional VAE approach as a way to incorporate auxiliary information of covariates for representation learning and prediction. However, the standard VAE and the conditional VAE are both designed for i.i.d. observations. In [14], the Gaussian Process Prior VAE (GPPVAE) approach was proposed to account for the association between sample points, which is crucial for correct model specification and optimal inference results in a longitudinal study. GPPVAE

replaces the i.i.d. priors over the latent space with a Gaussian process prior, enabling specification of within-subject correlations through covariance functions [15]. It has been adapted for heterogeneous types of covariates [16] and responses [17], and for situations with missing data [18].

While GPPVAE models the temporal and spatial correlations via a covariance function in the latent space, the proposed VaDA approach induces the dependence relationships through a deep generative network. In addition, the covariates introduced in GPPVAE mainly serve as auxiliary information fed into the covariance function and are typically low-dimensional, thus not significantly reducing the inference efficiency. VaDA models responses and covariates in a full generative manner, where the probabilistic auto-encoding networks in two separate but unified directions allow handling response variables and covariates both of high-dimensional. Regarding the data privacy, such an integrated generative model riches uncertainty over the latent space and benefits the out-of-sample data generation essential for data sharing [19]. Moreover, GPPVAE is challenged by handling the cubic computational complexity in the length of repeated measurements, where additional assumptions such as low-rank factorization of the covariance matrix are required to simplify the computation and to reduce the memory burden [14]. In contrast, borrowing the power of deep neural networks, VaDA can handle theoretically arbitrary complex dependence structures in the data.

A separate body of related works for correlated observations structures the latent space of time series by dynamical systems. In [20], the VAE approach was adapted by introducing a deep Kalman filter to the latent representations. Chung et al. [21] and Johnson et al. [22] embedded the VAE into a recurrent neural network for modeling high-dimensional correlated sequences. However, these models are more suitable to time series and incorporating the information of covariates is still an area to be explored.

Recent efforts concatenating the power of deep learning with longitudinal data re-assembled the mixed effects model with a feed-forward neural network to account for the complex non-linear effects of covariates [23]. As an extension of the multilayer perceptron regressor, this model is not a full generative model and mainly fitted for prediction purpose.

2.3. Clustering of Longitudinal Data

Clustering of longitudinal data remains a relatively under-explored area. The mixture of marginal models [24] introduces an overall mixture model framework to specify the possible heterogeneity of marginal effects in different clusters. The “working” correlation matrix is applied to model the dependence between measurements in each cluster, resulting in a maximization step in the EM algorithm as solving K (number of clusters) systems of GEEs. The grouped GEE approach [25] is a distance-based clustering algorithm for longitudinal data. It solves the GEEs in separate groups of subjects and optimizes the grouping using the Magalanobis distance with the estimated “working” correlation matrices. The two steps are implemented iteratively until reaching a stable point. The methods mentioned inherit the limitations of traditional models in handling non-linear effects of covariates and are not suitable for high-dimensional settings.

The Variational Deep Embedding (VaDE) approach [26] is a widely acknowledged variant of VAE for clustering analysis. It specifies a Gaussian mixture prior for the latent variables and is aimed at learning good representations that capture the statistical structure of the data. In a recent study about survival data clustering [27], the Variational Deep Survival Clustering (VaDeSC) approach related the latent representation of feature data from a VaDE encoder with the survival outcome by cluster-specific survival models to allow simultaneous clustering, representation learning and survival prediction [28]. Nevertheless, these approaches do not consider the relationships between sample points and assume i.i.d.

observations. In VaDA, we rich the latent space of VaDE with cluster-specific marginal models to realize simultaneous clustering, representation learning and prediction for longitudinal data.

3. Methods

In this section, we present VaDA—a generative auto-encoding approach to longitudinal data analysis, which is built on the success of VAE, VaDE and VaDeSC.

3.1. Problem Settings

The longitudinal dataset under consideration consists of N targeting subjects, such as stocks, patients and varieties of crop. The experimental design includes observing each subject over L different time points or locations on some outcome of interest and P relevant features, which constitute the observed data for response and covariates. Denote the covariate data of subject n , $n = 1, 2, \dots, N$ as $\mathbf{X}_n = (\mathbf{x}_{n1}, \mathbf{x}_{n2}, \dots, \mathbf{x}_{nL})^T$, where $\mathbf{x}_{nl} = (x_{nl1}, x_{nl2}, \dots, x_{nlP})^T$ is the data at time point or location l , $l = 1, 2, \dots, L$. The response information of subject n is denoted as $\mathbf{y}_n = (y_{n1}, y_{n2}, \dots, y_{nL})^T$. In our study, the P features can be of mixed type, i.e., x_{nlp} can be real-valued or discrete for $p = 1, 2, \dots, P$. To ease interpretation, we only consider the univariate outcome, which can be real-valued or discrete. Generalization to the multivariate case can be realized within the proposed framework. We leave formal exploration of this part to future work. For clustering of the longitudinal data, we consider for each subject a latent class label c_n , which is unobserved and takes value in $\{1, 2, \dots, K\}$. K is the total number of clusters. The problem then is threefold: (i) to make prediction of the outcome; (ii) to uncover a clustering for the subjects; (iii) to obtain the low-dimensional representations for the feature data.

3.2. Generative Process

We assume that the observed data $\{\mathbf{y}_n, \mathbf{X}_n\}_{n=1}^N$ are independently and identically generated from a random process consisting of the following steps.

1. Choose a class label $c_n \sim \text{Cat}(\boldsymbol{\pi})$.
2. For $l = 1, 2, \dots, L$:
 - (a) Choose a latent vector $\mathbf{z}_{nl} \sim \mathcal{N}(\boldsymbol{\mu}_{c_n}, \text{diag}(\boldsymbol{\sigma}_{c_n}^2))$.
 - (b) Generate $\mathbf{x}_{nl}$ by
 - i. if x_{nlp} is real-valued, choose $x_{nlp} \sim \mathcal{N}(\tilde{\mu}_{x,p}, \tilde{\sigma}_{x,p}^2)$,
 - ii. if x_{nlp} is categorical, choose $x_{nlp} \sim \text{Cat}(\tilde{\boldsymbol{\mu}}_{x,p})$,
 where the parameters are computed by

$$\{(\tilde{\mu}_{x,p}, \tilde{\sigma}_{x,p}^2), p \in \mathcal{V}(\text{con}); \tilde{\boldsymbol{\mu}}_{x,p}, p \in \mathcal{V}(\text{cat})\} = f_x(\mathbf{z}_{nl}; \boldsymbol{\theta}).$$
 - (c) Generate latent quantity $u_{nl} \sim \mathcal{N}(\beta_{c_n,0} + \mathbf{z}_{nl}^T \boldsymbol{\beta}_{c_n}, \sigma_{u,c_n}^2)$.
3. Generate $\mathbf{y}_n$ by
 - (a) For real-valued outcome, choose $\mathbf{y}_n \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_y, \text{diag}(\tilde{\boldsymbol{\sigma}}_y^2))$ with

$$\{\tilde{\boldsymbol{\mu}}_y, \tilde{\boldsymbol{\sigma}}_y^2\} = f_y(\mathbf{u}_n; \boldsymbol{\phi}),$$

where $\tilde{\boldsymbol{\mu}}_y = (\tilde{\mu}_{y,1}, \tilde{\mu}_{y,2}, \dots, \tilde{\mu}_{y,L})^T$ and $\tilde{\boldsymbol{\sigma}}_y^2 = (\tilde{\sigma}_{y,1}^2, \tilde{\sigma}_{y,2}^2, \dots, \tilde{\sigma}_{y,L}^2)^T$.

- (b) For categorical outcome, choose $\mathbf{y}_n \sim \prod_{l=1}^L \text{Cat}(\tilde{\boldsymbol{\mu}}_{y,l})$ with

$$\tilde{\boldsymbol{\mu}}_y = f_y(\mathbf{u}_n; \boldsymbol{\phi}),$$

where $\tilde{\boldsymbol{\mu}}_y = \{\tilde{\boldsymbol{\mu}}_{y,1}, \tilde{\boldsymbol{\mu}}_{y,2}, \dots, \tilde{\boldsymbol{\mu}}_{y,L}\}$.

We use $\text{Cat}(\tau)$ to denote the categorical distribution parameterized by τ , where $\tau = (\tau_1, \tau_2, \dots, \tau_G)^T \in \mathbb{R}_+^G$ and $\sum_{g=1}^G \tau_g = 1$. G is the number of categories. $\mathcal{N}(\eta, \Sigma)$ denotes the Gaussian distribution with mean η and covariance Σ . $\mathbf{z}_{nl} = (z_{nl1}, z_{nl2}, \dots, z_{nlD})^T$ is the latent representation or encoding of x_{nl} . D is the dimensions of $\mathbf{z}_{nl}$ and we have $D < P$. Denote the data of latent representations as $\mathbf{Z}_n = (\mathbf{z}_{n1}, \mathbf{z}_{n2}, \dots, \mathbf{z}_{nL})^T$. For $k = 1, 2, \dots, K$, $\boldsymbol{\mu}_k = (\mu_{k,1}, \mu_{k,2}, \dots, \mu_{k,D})^T \in \mathbb{R}^D$ and $\text{diag}(\sigma_k^2)$, where $\sigma_k^2 = (\sigma_{k,1}^2, \sigma_{k,2}^2, \dots, \sigma_{k,D}^2)^T \in \mathbb{R}_+^D$ are the mean vector and covariance matrix of $\mathbf{z}_{nl}$ in class k . Denote $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}_{k=1}^K$ and $\sigma^2 = \{\sigma_k^2\}_{k=1}^K$.

We use $\mathcal{V}(\text{con})$ and $\mathcal{V}(\text{cat})$ to denote the sets of indexes for real-valued and categorical features, respectively. It has $\mathcal{V}(\text{con}) \cap \mathcal{V}(\text{cat}) = \emptyset$ and $\mathcal{V}(\text{con}) \cup \mathcal{V}(\text{cat}) = \{1, 2, \dots, P\}$. Collectively, the parameters to generate x_{nl} are derived from a decoding neural network f_x with input $\mathbf{z}_{nl}$ and parameterized by θ .

The quantities $\mathbf{u}_n = (u_{n1}, u_{n2}, \dots, u_{nL})^T$ are the intermediate latent states of the correlated response variables, which are mutually independent given the class label and structured by the underlying marginal model with cluster-specific marginal mean functions $\beta_{k,0} + \mathbf{z}_{nl}^T \boldsymbol{\beta}_k$, $k = 1, 2, \dots, K$. We assume that z_{nld} , $d = 1, 2, \dots, D$ has linear effects on u_{nl} and denote the marginal effects in cluster k as $\boldsymbol{\beta}_k = (\beta_{k,1}, \beta_{k,2}, \dots, \beta_{k,D})^T \in \mathbb{R}^D$ and residual variance as $\sigma_{u,k}^2 \in \mathbb{R}_+$. $\beta_{k,0} \in \mathbb{R}$ allows modeling the intercept of the marginal mean function. Denote $\boldsymbol{\beta} = \{\beta_{k,0}, \boldsymbol{\beta}_k\}_{k=1}^K$ and $\sigma_u^2 = \{\sigma_{u,k}^2\}_{k=1}^K$. $\mathbf{u}_n$ has the practical meaning as “pure outcomes”.

The final observed outcomes $\mathbf{y}_n$ are produced through a non-observable mixing process, e.g., the gene–environment interaction in bioscience and the temporal and spatial evolution of behavior. We simulate this process artificially using the decoder f_y which has input $\mathbf{u}_n$ and is parameterized by ϕ . Modeling this process is essential in VaDA, where the measurements within a subject henceforth enter into a deep “alliance”. The overall generative process assumed by VaDA is depicted in Figure 1.

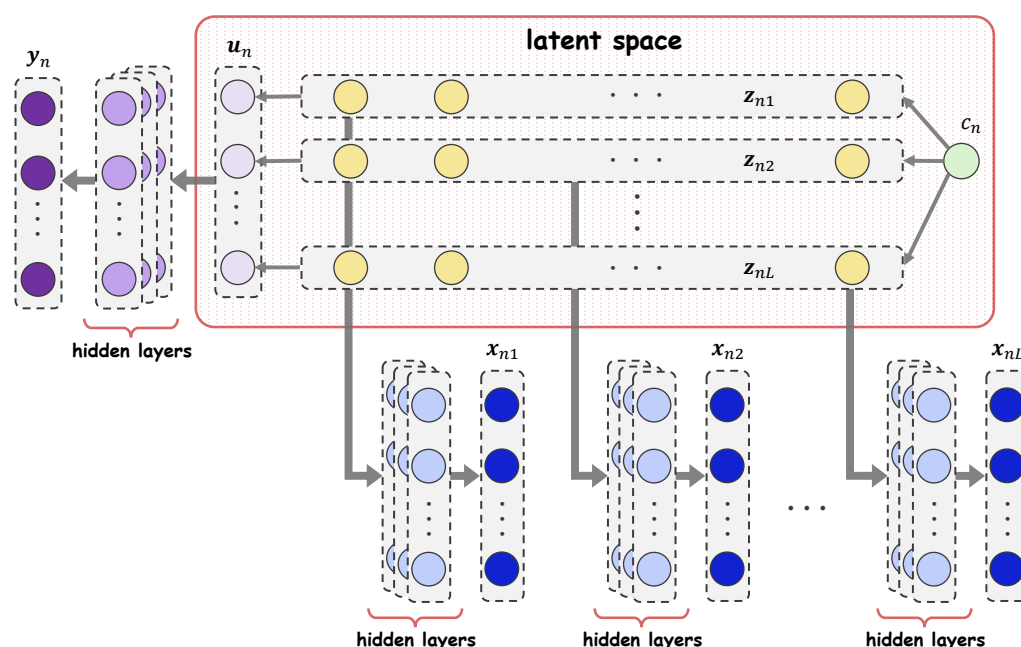


Figure 1. Data generating process of VaDA.

According to the generative process described above, the complete-data likelihood $p(\mathbf{y}_n, \mathbf{u}_n, \mathbf{X}_n, \mathbf{Z}_n, c_n)$ can be factorized as

$$p(\mathbf{y}_n, \mathbf{u}_n, \mathbf{X}_n, \mathbf{Z}_n, c_n) = p(\mathbf{y}_n | \mathbf{u}_n) p(\mathbf{u}_n | \mathbf{Z}_n, c_n) p(\mathbf{X}_n | \mathbf{Z}_n) p(\mathbf{Z}_n | c_n) p(c_n), \quad (1)$$

which can be further factorized by

$$\begin{aligned} p(\mathbf{u}_n | \mathbf{Z}_n, c_n) &= \prod_{l=1}^L p(u_{nl} | \mathbf{z}_{nl}, c_n), \\ p(\mathbf{X}_n | \mathbf{Z}_n) &= \prod_{l=1}^L p(\mathbf{x}_{nl} | \mathbf{z}_{nl}), \\ p(\mathbf{Z}_n | c_n) &= \prod_{l=1}^L p(\mathbf{z}_{nl} | c_n). \end{aligned}$$

Taking the real-valued outcome as an example, the probabilities composing (1) are defined by

$$\begin{aligned} p(c_n) &= \text{Cat}(c_n; \boldsymbol{\pi}), \\ p(\mathbf{z}_{nl} | c_n) &= \mathcal{N}(\mathbf{z}_{nl}; \boldsymbol{\mu}_{c_n}, \text{diag}(\sigma_{c_n}^2)), \\ p(\mathbf{x}_{nl} | \mathbf{z}_{nl}) &= \prod_{p \in \mathcal{V}(\text{con})} \mathcal{N}(x_{nlp}; \tilde{\mu}_{x,p}, \tilde{\sigma}_{x,p}^2) \prod_{p \in \mathcal{V}(\text{cat})} \text{Cat}(x_{nlp}; \tilde{\mu}_{x,p}), \\ p(u_{nl} | \mathbf{z}_{nl}, c_n) &= \mathcal{N}(u_{nl}; \beta_{c_n,0} + \mathbf{z}_{nl}^T \boldsymbol{\beta}_{c_n}, \sigma_{u,c_n}^2), \\ p(\mathbf{y}_n | \mathbf{u}_n) &= \mathcal{N}(\mathbf{y}_n; \tilde{\boldsymbol{\mu}}_y, \text{diag}(\tilde{\sigma}_y^2)). \end{aligned}$$

3.3. Variational Lower Bound

Based on the complete-data likelihood given in (1), the observed-data log-likelihood of VaDA can be written as

$$\log p(\mathbf{y}_n, \mathbf{X}_n) = \log \int \sum_{c_n} p(\mathbf{y}_n | \mathbf{u}_n) p(\mathbf{u}_n | \mathbf{Z}_n, c_n) p(\mathbf{X}_n | \mathbf{Z}_n) p(\mathbf{Z}_n | c_n) p(c_n) \, d\mathbf{Z}_n \, d\mathbf{u}_n.$$

By Jensen's inequality, it follows

$$\log p(\mathbf{y}_n, \mathbf{X}_n) \geq \mathbb{E}_{q(\mathbf{u}_n, \mathbf{Z}_n, c_n | \mathbf{y}_n, \mathbf{X}_n)} \left[\log \frac{p(\mathbf{y}_n | \mathbf{u}_n) p(\mathbf{u}_n | \mathbf{Z}_n, c_n) p(\mathbf{X}_n | \mathbf{Z}_n) p(\mathbf{Z}_n | c_n) p(c_n)}{q(\mathbf{u}_n, \mathbf{Z}_n, c_n | \mathbf{y}_n, \mathbf{X}_n)} \right] =: \mathcal{L}_{\text{ELBO}}(\mathbf{y}_n, \mathbf{X}_n),$$

where $\mathcal{L}_{\text{ELBO}}(\mathbf{y}_n, \mathbf{X}_n)$ is the evidence lower bound objective (ELBO) of the log-likelihood $\log p(\mathbf{y}_n, \mathbf{X}_n)$, and $q(\mathbf{u}_n, \mathbf{Z}_n, c_n | \mathbf{y}_n, \mathbf{X}_n)$ is the variational auxiliary function to approximate the inaccessible true posterior $p(\mathbf{u}_n, \mathbf{Z}_n, c_n | \mathbf{y}_n, \mathbf{X}_n)$. In VaDA, it is assumed that the auxiliary $q(\mathbf{u}_n, \mathbf{Z}_n, c_n | \mathbf{y}_n, \mathbf{X}_n)$ is a mean-field distribution and can be factorized as

$$q(\mathbf{u}_n, \mathbf{Z}_n, c_n | \mathbf{y}_n, \mathbf{X}_n) = q(\mathbf{u}_n | \mathbf{y}_n) q(\mathbf{Z}_n | \mathbf{X}_n) q(c_n | \mathbf{y}_n, \mathbf{X}_n), \quad (2)$$

where

$$q(\mathbf{Z}_n | \mathbf{X}_n) = \prod_{l=1}^L q(\mathbf{z}_{nl} | \mathbf{x}_{nl}).$$

Under some mathematical manipulations, it follows that the lower bound $\mathcal{L}_{\text{ELBO}}(\mathbf{y}_n, \mathbf{X}_n)$ can be written as

$$\begin{aligned} \mathcal{L}_{\text{ELBO}}(\mathbf{y}_n, \mathbf{X}_n) &= \mathbb{E} [\log p(\mathbf{y}_n | \mathbf{u}_n) + \log p(\mathbf{u}_n | \mathbf{Z}_n, c_n) + \log p(\mathbf{X}_n | \mathbf{Z}_n) + \log p(\mathbf{Z}_n | c_n) + \log p(c_n) \\ &\quad - \log q(\mathbf{u}_n | \mathbf{y}_n) - \log q(\mathbf{Z}_n | \mathbf{X}_n) - \log q(c_n | \mathbf{y}_n, \mathbf{X}_n)], \end{aligned} \quad (3)$$

where the expectation is taken w.r.t. $q(\mathbf{u}_n, \mathbf{Z}_n, c_n | \mathbf{y}_n, \mathbf{X}_n)$ with factorization (2).

We use neural networks to model $q(\mathbf{u}_n | \mathbf{y}_n)$ and $q(\mathbf{z}_{nl} | \mathbf{x}_{nl})$, i.e.,

$$q(\mathbf{u}_n | \mathbf{y}_n) = \mathcal{N}(\mathbf{u}_n; \tilde{\boldsymbol{\mu}}_u, \text{diag}(\tilde{\sigma}_u^2)), \quad \{\tilde{\boldsymbol{\mu}}_u, \tilde{\sigma}_u^2\} = g_u(\mathbf{y}_n; \boldsymbol{\lambda}), \quad (4)$$

where $\tilde{\mu}_u = (\tilde{\mu}_{u,1}, \tilde{\mu}_{u,2}, \dots, \tilde{\mu}_{u,L})^T$ and $\tilde{\sigma}_u^2 = (\tilde{\sigma}_{u,1}^2, \tilde{\sigma}_{u,2}^2, \dots, \tilde{\sigma}_{u,L}^2)^T$ are parameters for the posterior density function of $\mathbf{u}_n$. They are derived from the encoder network g_u with input $\mathbf{y}_n$ and parameters λ . Similarly,

$$q(\mathbf{z}_{nl}|\mathbf{x}_{nl}) = \mathcal{N}(\mathbf{z}_{nl}; \tilde{\mu}_z, \text{diag}(\tilde{\sigma}_z^2)), \quad \{\tilde{\mu}_z, \tilde{\sigma}_z^2\} = g_z(\mathbf{x}_{nl}; \gamma), \quad (5)$$

where $\tilde{\mu}_z = (\tilde{\mu}_{z,1}, \tilde{\mu}_{z,2}, \dots, \tilde{\mu}_{z,D})^T$ and $\tilde{\sigma}_z^2 = (\tilde{\sigma}_{z,1}^2, \tilde{\sigma}_{z,2}^2, \dots, \tilde{\sigma}_{z,D}^2)^T$ are parameters for the posterior distribution of $\mathbf{z}_{nl}$. They are derived from the encoder network g_z with input $\mathbf{x}_{nl}$ and parameters γ .

We do not introduce excessive assumptions for distribution $q(c_n|\mathbf{y}_n, \mathbf{X}_n)$ but obtain it by integrating $\mathbf{u}_n$ and $\mathbf{Z}_n$ from the accessible posterior $p(c_n|\mathbf{u}_n, \mathbf{Z}_n)$, i.e.,

$$q(c_n|\mathbf{y}_n, \mathbf{X}_n) = \mathbb{E}_{q(\mathbf{u}_n|\mathbf{y}_n)q(\mathbf{Z}_n|\mathbf{X}_n)} [p(c_n|\mathbf{u}_n, \mathbf{Z}_n)],$$

where

$$p(c_n|\mathbf{u}_n, \mathbf{Z}_n) = \frac{p(\mathbf{u}_n|\mathbf{Z}_n, c_n)p(\mathbf{Z}_n|c_n)p(c_n)}{\sum_{c_n} p(\mathbf{u}_n|\mathbf{Z}_n, c_n)p(\mathbf{Z}_n|c_n)p(c_n)} =: h(\mathbf{u}_n, \mathbf{Z}_n, c_n; \mathcal{H}).$$

Therefore, $h(\mathbf{u}_n, \mathbf{Z}_n, c_n; \mathcal{H})$ is a known function with inputs $(\mathbf{u}_n, \mathbf{Z}_n, c_n)$ and parameters $\mathcal{H} = \{\beta, \sigma_u^2, \mu, \sigma^2, \pi\}$.

3.4. Understanding the ELBO of VaDA

The ELBO in (3) can be rearranged as

$$\mathcal{L}_{\text{ELBO}}(\mathbf{y}_n, \mathbf{X}_n) = \mathbb{E}_{q(\mathbf{u}_n|\mathbf{y}_n)q(\mathbf{Z}_n|\mathbf{X}_n)} [\log p(\mathbf{y}_n, \mathbf{X}_n|\mathbf{u}_n, \mathbf{Z}_n)] - \text{D}_{\text{KL}}[q(\mathbf{u}_n, \mathbf{Z}_n, c_n|\mathbf{y}_n, \mathbf{X}_n) || p(\mathbf{u}_n, \mathbf{Z}_n, c_n)].$$

The first term is the reconstruction term, which encourages VaDA to explain the dataset well. It can be further separated into two parts as

$$\mathbb{E}_{q(\mathbf{u}_n|\mathbf{y}_n)q(\mathbf{Z}_n|\mathbf{X}_n)} [\log p(\mathbf{y}_n, \mathbf{X}_n|\mathbf{u}_n, \mathbf{Z}_n)] = \mathbb{E}_{q(\mathbf{u}_n|\mathbf{y}_n)} [\log p(\mathbf{y}_n|\mathbf{u}_n)] + \sum_{l=1}^L \mathbb{E}_{q(\mathbf{z}_{nl}|\mathbf{x}_{nl})} [\log p(\mathbf{x}_{nl}|\mathbf{z}_{nl})],$$

corresponding to the reconstructions for outcomes and feature data, respectively. The second term is the Kullback-Leibler (KL) divergence from the prior of $(\mathbf{u}_n, \mathbf{Z}_n, c_n)$ to their auxiliary posterior, which therefore regularizes the latent embedding $(\mathbf{u}_n, \mathbf{Z}_n)$ to lie on a structured manifold as a mixture model. Appendix B illustrates the evolution of model structures and the induced changes of ELBOs between VAE, VaDE and VaDA.

Due to the complexity of the decoding and encoding networks, the expectation for the reconstruction term cannot be computed exactly. As in VAE and VaDE, the Stochastic Gradient Variational Bayes (SGVB) estimator and the reparameterization trick [4] are applied, which gives

$$\mathbb{E}_{q(\mathbf{u}_n|\mathbf{y}_n)} [\log p(\mathbf{y}_n|\mathbf{u}_n)] \approx \frac{1}{M} \sum_{m=1}^M \log \mathcal{N}(\mathbf{y}_n; \tilde{\mu}_y^{(m)}, \text{diag}(\tilde{\sigma}_y^{2,(m)})),$$

and

$$\mathbb{E}_{q(\mathbf{z}_{nl}|\mathbf{x}_{nl})} [\log p(\mathbf{x}_{nl}|\mathbf{z}_{nl})] \approx \frac{1}{M} \sum_{m=1}^M \left[\sum_{p \in \mathcal{V}(\text{con})} \log \mathcal{N}(x_{nl,p}; \tilde{\mu}_{x,p}^{(m)}, \tilde{\sigma}_{x,p}^{2,(m)}) + \sum_{p \in \mathcal{V}(\text{cat})} \log \text{Cat}(x_{nl,p}; \tilde{\mu}_{x,p}^{(m)}) \right].$$

We compute $\tilde{\mu}_y^{(m)}$ and $\tilde{\sigma}_y^{2,(m)}$ as

$$\{\tilde{\mu}_y^{(m)}, \tilde{\sigma}_y^{2,(m)}\} = f_y(\mathbf{u}_n^{(m)}; \phi),$$

where $\mathbf{u}_n^{(m)}$ is the m th sample from $q(\mathbf{u}_n|\mathbf{y}_n)$ to produce the Monte Carlo (MC) approximation and can be obtained by

$$\mathbf{u}_n^{(m)} = \tilde{\boldsymbol{\mu}}_u + \tilde{\boldsymbol{\sigma}}_u \cdot \boldsymbol{\epsilon}_u^{(m)}, \quad \boldsymbol{\epsilon}_u^{(m)} = (\epsilon_{u,1}^{(m)}, \epsilon_{u,2}^{(m)}, \dots, \epsilon_{u,L}^{(m)})^T \text{ and } \epsilon_{u,l}^{(m)} \sim \mathcal{N}(0, 1).$$

$\tilde{\boldsymbol{\mu}}_u$ and $\tilde{\boldsymbol{\sigma}}_u$ are obtained from the encoding network g_u in (4). Meanwhile,

$$\{(\tilde{\boldsymbol{\mu}}_{x,p}^{(m)}, \tilde{\boldsymbol{\sigma}}_{x,p}^{2,(m)}), p \in \mathcal{V}(\text{con}); \tilde{\boldsymbol{\mu}}_{x,p}^{(m)}, p \in \mathcal{V}(\text{cat})\} = f_x(\mathbf{z}_{nl}^{(m)}; \boldsymbol{\theta}),$$

where $\mathbf{z}_{nl}^{(m)}$ is the m th sample from $q(\mathbf{z}_{nl}|\mathbf{x}_{nl})$ generated by

$$\mathbf{z}_{nl}^{(m)} = \tilde{\boldsymbol{\mu}}_z + \tilde{\boldsymbol{\sigma}}_z \cdot \boldsymbol{\epsilon}_z^{(m)}, \quad \boldsymbol{\epsilon}_z^{(m)} = (\epsilon_{z,1}^{(m)}, \epsilon_{z,2}^{(m)}, \dots, \epsilon_{z,D}^{(m)})^T \text{ and } \epsilon_{z,d}^{(m)} \sim \mathcal{N}(0, 1).$$

$\tilde{\boldsymbol{\mu}}_z$ and $\tilde{\boldsymbol{\sigma}}_z$ are obtained from network g_z in (5). Denote $\mathbf{Z}_n^{(m)} = (\mathbf{z}_{n1}^{(m)}, \mathbf{z}_{n2}^{(m)}, \dots, \mathbf{z}_{nL}^{(m)})^T$.

Given the MC samples of $\mathbf{u}_n$ and $\mathbf{Z}_n$, we can also form an approximation for $q(c_n|\mathbf{y}_n, \mathbf{X}_n)$ as

$$q(c_n = k|\mathbf{y}_n, \mathbf{X}_n) \approx \frac{1}{M} \sum_{m=1}^M p(c_n = k|\mathbf{u}_n^{(m)}, \mathbf{Z}_n^{(m)}) = \frac{1}{M} \sum_{m=1}^M h(\mathbf{u}_n^{(m)}, \mathbf{Z}_n^{(m)}, k; \mathcal{H}) =: \tilde{\pi}_k, \quad (6)$$

for $k = 1, 2, \dots, K$, under which the KL divergence between $q(\mathbf{u}_n, \mathbf{Z}_n, c_n|\mathbf{y}_n, \mathbf{X}_n)$ and $p(\mathbf{u}_n, \mathbf{Z}_n, c_n)$ can be computed analytically (see Appendix A for detailed derivations), given by

$$\begin{aligned} \mathbf{D}_{\text{KL}}[q(\mathbf{u}_n, \mathbf{Z}_n, c_n|\mathbf{y}_n, \mathbf{X}_n)||p(\mathbf{u}_n, \mathbf{Z}_n, c_n)] \\ = \frac{1}{2} \sum_{k=1}^K \tilde{\pi}_k \sum_{l=1}^L \left\{ \log \sigma_{u,k}^2 + (\sigma_{u,k}^2)^{-1} [(\tilde{\boldsymbol{\mu}}_{u,l} - \beta_{k,0} - \tilde{\boldsymbol{\mu}}_z^T \boldsymbol{\beta}_k)^2 + \tilde{\boldsymbol{\sigma}}_{u,l}^2 + \boldsymbol{\beta}_k^T \text{diag}(\tilde{\boldsymbol{\sigma}}_z^2) \boldsymbol{\beta}_k] \right\} \\ + \frac{1}{2} \sum_{k=1}^K \tilde{\pi}_k \sum_{l=1}^L \sum_{d=1}^D \left\{ \log \sigma_{k,d}^2 + (\sigma_{k,d}^2)^{-1} [(\tilde{\boldsymbol{\mu}}_{z,d} - \mu_{k,d})^2 + \tilde{\sigma}_{z,d}^2] \right\} - \sum_{k=1}^K \tilde{\pi}_k \log \pi_k \\ - \frac{1}{2} \sum_{l=1}^L \log \tilde{\sigma}_{u,l}^2 - \frac{1}{2} \sum_{l=1}^L \sum_{d=1}^D \log \tilde{\sigma}_{z,d}^2 + \sum_{k=1}^K \tilde{\pi}_k \log \tilde{\pi}_k. \end{aligned}$$

It is noticeable that the overall inference framework conducts decoding and encoding processes in two separate directions. The first is the cross-sectional direction from the latent space to the feature space with encoder g_z and decoder f_x , which is the same as for VAE and VaDE. The other is the longitudinal or allaying direction from the latent space to the outcomes with encoder g_u and decoder f_y . To ease interpretation, we term g_z and f_x as X-encoder and X-decoder, and term g_u and f_y as Y-encoder and Y-decoder.

3.5. Clustering and Prediction

In the inference framework of VaDA, clustering of subjects can be realized in two paths. The first is the complete-data case where besides the feature data $\mathbf{X}_n$ we have already observed the outcomes $\mathbf{y}_n$. The clustering therefore can be obtained by assigning subject n to cluster

$$k^* = \arg \max_{k \in \{1, 2, \dots, K\}} q(c_n = k|\mathbf{y}_n, \mathbf{X}_n), \quad (7)$$

where the probability $q(c_n = k | \mathbf{y}_n, \mathbf{X}_n)$ can be approximated by $\tilde{\pi}_k$ defined in (6). The second path is when the outcomes $\mathbf{y}_n$ have not been observed and are targets to be predicted. The clustering of subject n is given by

$$k^* = \arg \max_{k \in \{1, 2, \dots, K\}} q(c_n = k | \mathbf{X}_n), \quad (8)$$

where

$$q(c_n = k | \mathbf{X}_n) \approx \frac{1}{M} \sum_{m=1}^M p(c_n = k | \mathbf{Z}_n^{(m)}) = \frac{1}{M} \sum_{m=1}^M \frac{p(\mathbf{Z}_n^{(m)} | c_n = k) p(c_n = k)}{\sum_{k=1}^K p(\mathbf{Z}_n^{(m)} | c_n = k) p(c_n = k)}.$$

Prediction of $\mathbf{y}_n$ from the observed data $\mathbf{X}_n$ can be obtained by the following steps. The first step is to transform the feature data $\mathbf{X}_n = (x_{n1}, x_{n2}, \dots, x_{nL})^T$ to their latent representations $\hat{\mathbf{Z}}_n = (\hat{z}_{n1}, \hat{z}_{n2}, \dots, \hat{z}_{nL})^T$ through the learned X-encoder. We have $\hat{z}_{nl} = \hat{\mu}_z$, which is the output of $g_z(x_{nl}; \gamma)$. In the second step, the representations are inserted into the K latent cluster-specific marginal models to obtain the estimated pure outcomes $\hat{\mathbf{u}}_n^k = (\hat{u}_{n1}^k, \hat{u}_{n2}^k, \dots, \hat{u}_{nL}^k)^T, k = 1, 2, \dots, K$, where $\hat{u}_{nl}^k = \beta_{k,0} + \hat{z}_{nl}^T \beta_k$. Then, we use the Y-decoder to decode $\hat{\mathbf{u}}_n^k$ back to the outcomes $\hat{\mathbf{y}}_n^k = \hat{\mu}_y^k$, which are the outputs of $f_y(\hat{\mathbf{u}}_n^k; \phi)$. The final prediction can be obtained by

$$\hat{\mathbf{y}}_n = \sum_{k=1}^K q(c_n = k | \mathbf{X}_n) \cdot \hat{\mathbf{y}}_n^k,$$

where we have used the probability mass function $q(c_n | \mathbf{X}_n)$ not $q(c_n | \mathbf{y}_n, \mathbf{X}_n)$ as $\mathbf{y}_n$ are unobserved.

4. Synthetic Study

In this section, we assess the proposed method for longitudinal data analysis through different simulated data scenarios. The objective is to evaluate the quality of clustering and prediction performance obtained coherently using VaDA. The baseline methods include VaDE+MM, VAE+GMM+MM and GMM+MM. VaDE+MM for longitudinal data analysis fits cluster-specific marginal models (MM) based on the representations of feature data and clustering derived simultaneously from VaDE. In VAE+GMM+MM, the clustering is realized by fitting a Gaussian Mixture Model (GMM) based on the obtained representations from VAE. Then, the cluster-specific marginal models are fitted for longitudinal prediction. GMM+MM applies the GMM for clustering and fits the cluster-specific marginal models based on the original dataset.

When fitting marginal models by the GEE approach, three types of “working” correlation assumption were considered, i.e., independence (MM0), ar1 (MM1) and exchangeable (MM2), with increased complexity on the allowed correlation structures [2]. Note that the assumptions MM1 and MM2 are equivalent when the length of measurements $L = 2$. Another problem is that the clustering results either from VaDE or GMM may differ within subject. To obtain a valid clustering for longitudinal data, we made subtle modifications in the algorithms of VaDE and GMM, where the clustering results for each subject were merged across measurements by summarizing the probability of belonging in each cluster and assigning the subject to cluster having the highest total probability. Due to the frequently appearing “rank deficiency” problem in the marginal models [29], we used the Principal Component Analysis (PCA) [30,31] to project further the feature embeddings in VaDE+MM and VAE+GMM+MM and the original feature data in GMM+MM to orthogonal principal components before fitting the cluster-specific marginal models to ensure smooth implementation of these baselines. The detailed instructions are presented in Appendix G.

Four synthetic scenarios were considered. The generated feature data had dimensions $P = 100$ and the length of measurements for a subject was $L = 4$. We used a number of $N = 2500$ subjects. Therefore, there were totally $N \times L = 10,000$ observations. In Scenarios 11 and 01, we considered the data generated from three clusters with ten-dimensional latent embeddings, where the marginal effects were non-linear. Scenario 11 corresponded to the well-specified case where the data were generated using the VaDA decoders in two separate directions. Scenario 01 used only the X-decoder to generate the feature data and the outcome variables for each subject were mutually independent given by the pure outcomes. In Scenarios 10 and 00, we considered the data generated from three clusters with linear cluster-specific marginal effects. Scenario 10 used only the Y-decoder to induce dependence between outcomes and the feature data were generated directly from a mixture model with diagonal component covariance matrices. In Scenario 00, we considered the baseline where the data were generated following a simple linear regression model in each cluster where neither the X-decoder nor the Y-decoder was applied. See Appendix C for detailed generating processes and parameter configurations in the four scenarios. Here, we consider the real-valued outcome. The categorical case is examined in the following section with the CelebA dataset. In each scenario, 30 random datasets were generated independently to make reliable assessment on the methods.

The deep learning methods VaDA, VaDE and VAE were implemented under the backbone of TensorFlow [32] and trained with the Adam method for optimization [33] at the default learning rate of 0.001. Due to the difference between the inference frameworks, the batch size for VaDA was set as 128, and for VaDE and VAE it was set as $128 \times L = 512$. For all the three models, the number of training epochs was set as 500, which had been decided approximately by cross-validation (Appendix E). To make a fair comparison, the decoder and encoder network architectures in VaDE and VAE were set the same as the X-decoder and X-encoder in VaDA, respectively. See Appendix D for specific network configurations. We used $M = 1$ MC sample for the SGVB estimator as its usual settings. Further explorations on this hyperparameter are provided in Appendix F. And opposed to the practice in many deep unsupervised learning methods, we did not pretrain the networks. But a re-scaling of the response and covariate variables to zero-mean and unit-variance was conducted before training the models. The training/testing ratio of the data was set as 4/1. As the latent dimensions D and the number of clusters K were unknown a priori, we varied the settings of $D \in \{2, 5, 10, 20, 100\}$ and $K \in \{2, 3, 4\}$ when fitting each model, which were reasonable quantities considered in practice.

Comparison between methods was based on the metrics of mean square error (MSE) and adjusted Rand index (ARI) [27]. The metric MSE was used to assess the accuracy of prediction, while ARI was computed to evaluate the quality of clustering. As we know the data generating process, the ground-truth class labels can be used post hoc to unambiguously assess the clustering performance. ARI allows evaluation between two data classifications with different number of classes and is not affected by the label switching problem [34].

Table 1 compares the results of MSE on the testing datasets when setting $K = 3$ in the four scenarios. The fitting results on the training datasets are presented in the Supplementary Materials (Section S1) as a reference. As can be seen, VaDA significantly outperforms the other methods exhibiting high generalization performance across the four scenarios. It is noticeable that VaDA has a remarkably higher performance in Scenarios 11 and 10 (having within-subject correlations), which implies the importance of correctly modeling the within-subject correlations. VaDA can simulate more sophisticated dependence relationships compared with the marginal model with commonly used “working” correlation structures. Additionally, a decreasing on the MSE can be observed in Scenarios 11 and 01

(with non-linear marginal effects) as the dimensions D of latent representation increases. While small D value could cause a loss of information, larger D allows capturing more latent features that improve the prediction accuracy. In our experiments on VaDA, increasing the latent dimensions D has not resulted in an over-fitting problem, which however becomes severe when using the methods VaDE+MM and VAE+GMM+MM in Scenarios 10 and 00 (with linear marginal effects). In Scenarios 01 and 00 (no within-subject correlation), VaDA performs slightly inferior to the baseline method GMM+MM. This bias could be a result of overly modeling the within-subject correlations which however do not exist.

Table 1. Prediction results on the testing datasets evaluated by the mean square error with varied settings of the latent dimensions D when the number of clusters $K = 3$.

Scenario	Method	$D = 2$	$D = 5$	$D = 10$	$D = 20$	$D = 100$
11	VaDA *	0.281 ± 0.117	0.122 ± 0.045	0.068 ± 0.007	0.059 ± 0.005	0.062 ± 0.008
	VaDE+MM0	0.769 ± 0.026	0.749 ± 0.025	0.730 ± 0.021	0.728 ± 0.020	0.740 ± 0.022
	VaDE+MM1	0.773 ± 0.028	0.752 ± 0.027	0.732 ± 0.022	0.729 ± 0.020	0.735 ± 0.021
	VaDE+MM2	0.774 ± 0.027	0.751 ± 0.026	0.729 ± 0.020	0.726 ± 0.021	0.726 ± 0.021
	VAE+GMM+MM0	0.725 ± 0.020	0.726 ± 0.021	0.726 ± 0.020	0.728 ± 0.021	0.740 ± 0.022
	VAE+GMM+MM1	0.728 ± 0.020	0.729 ± 0.021	0.729 ± 0.020	0.730 ± 0.021	0.734 ± 0.021
	VAE+GMM+MM2	0.725 ± 0.020	0.726 ± 0.021	0.725 ± 0.021	0.726 ± 0.021	0.726 ± 0.021
	GMM+MM0	—	—	—	—	0.787 ± 0.078
	GMM+MM1	—	—	—	—	0.757 ± 0.044
	GMM+MM2	—	—	—	—	0.726 ± 0.021
01	VaDA	0.233 ± 0.097	0.069 ± 0.040	0.053 ± 0.009	0.048 ± 0.011	0.048 ± 0.010
	VaDE+MM0	0.397 ± 0.062	0.274 ± 0.066	0.092 ± 0.057	0.007 ± 0.004	0.005 ± 0.001
	VaDE+MM1	0.781 ± 0.137	0.470 ± 0.192	0.119 ± 0.098	0.007 ± 0.004	0.005 ± 0.001
	VaDE+MM2	0.782 ± 0.139	0.541 ± 0.208	0.120 ± 0.097	0.007 ± 0.004	0.005 ± 0.001
	VAE+GMM+MM0	0.046 ± 0.001	0.035 ± 0.001	0.015 ± 0.024	0.013 ± 0.029	0.015 ± 0.025
	VAE+GMM+MM1	0.046 ± 0.001	0.035 ± 0.001	0.015 ± 0.024	0.013 ± 0.029	0.015 ± 0.025
	VAE+GMM+MM2	0.046 ± 0.001	0.035 ± 0.001	0.015 ± 0.024	0.013 ± 0.029	0.015 ± 0.025
	GMM+MM0	—	—	—	—	0.012 ± 0.019
	GMM+MM1	—	—	—	—	0.012 ± 0.019
	GMM+MM2	—	—	—	—	0.012 ± 0.019
10	VaDA	0.092 ± 0.005	0.092 ± 0.005	0.092 ± 0.004	0.092 ± 0.004	0.091 ± 0.005
	VaDE+MM0	1.003 ± 0.026	1.003 ± 0.026	1.004 ± 0.026	73.75 ± 243.5	3002 ± 2948
	VaDE+MM1	1.004 ± 0.026	1.004 ± 0.026	1.005 ± 0.026	77.57 ± 262.2	3023 ± 2961
	VaDE+MM2	1.002 ± 0.026	1.003 ± 0.026	1.002 ± 0.026	15.03 ± 23.92	1108 ± 2936
	VAE+GMM+MM0	1.003 ± 0.026	1.003 ± 0.026	1.005 ± 0.026	88.12 ± 290.5	3961 ± 1976
	VAE+GMM+MM1	1.004 ± 0.026	1.005 ± 0.026	1.006 ± 0.026	89.30 ± 293.8	4029 ± 2010
	VAE+GMM+MM2	1.002 ± 0.026	1.003 ± 0.026	1.003 ± 0.026	19.52 ± 56.10	933.7 ± 419.5
	GMM+MM0	—	—	—	—	1.020 ± 0.028
	GMM+MM1	—	—	—	—	1.022 ± 0.028
	GMM+MM2	—	—	—	—	1.004 ± 0.026

Table 1. Cont.

Scenario	Method	$D = 2$	$D = 5$	$D = 10$	$D = 20$	$D = 100$
00	VaDA	0.108 ± 0.007	0.109 ± 0.011	0.108 ± 0.010	0.106 ± 0.008	0.103 ± 0.003
	VaDE+MM0	0.112 ± 0.017	0.108 ± 0.013	0.107 ± 0.016	5.987 ± 16.04	169.1 ± 79.02
	VaDE+MM1	0.112 ± 0.017	0.111 ± 0.029	0.107 ± 0.016	5.993 ± 16.12	168.4 ± 79.18
	VaDE+MM2	0.112 ± 0.017	0.111 ± 0.024	0.107 ± 0.016	5.959 ± 15.93	168.2 ± 78.26
	VAE+GMM+MM0	0.104 ± 0.011	0.103 ± 0.011	0.101 ± 0.012	7.666 ± 24.27	385.4 ± 88.45
	VAE+GMM+MM1	0.104 ± 0.011	0.103 ± 0.011	0.101 ± 0.012	7.555 ± 24.18	386.4 ± 91.34
	VAE+GMM+MM2	0.104 ± 0.011	0.103 ± 0.011	0.101 ± 0.012	7.558 ± 24.06	385.2 ± 89.04
	GMM+MM0	—	—	—	—	0.018 ± 0.026
	GMM+MM1	—	—	—	—	0.018 ± 0.026
	GMM+MM2	—	—	—	—	0.018 ± 0.026

* Visual highlighting with gray background is applied to the results of our proposed method to facilitate quick identification and comparison with baseline methods.

Table 2 compares the results of MSE on the testing datasets when setting $D = 10$ in the four scenarios. We consider the settings $K \in \{2, 3, 4\}$ while the true number of latent clusters is 3. By an overall inspection, it can be found that the prediction results are not very sensitive to the specification of K . VaDA shows stable performance with varied K values across the four scenarios and significantly outperforms the other methods in Scenarios 11 and 10 where the within-subject correlations exist. In addition, the baseline method GMM+MM can achieve a good prediction accuracy in Scenarios 01 and 00 (no within-subject correlation). However, its performance degrades heavily in Scenarios 11 and 10 when dependence relationships between outcomes are present. Overall, modeling of the within-subject correlations has a more significant impact on the prediction results in our experiments. The unexpected good performance of GMM+MM in Scenario 01 with non-linear marginal effects could be a result of adequate sample size and no severe rank-deficiency problem in the design matrix transformed by PCA.

The clustering results on the testing datasets assessed by the ARI value are provided in Tables 3 and 4. The values of ARI closer to one indicate higher degree of matching between two classifications. There are totally five different clustering results under comparison provided by VaDA-YX, VaDA-X, VaDE, VAE+GMM and GMM, respectively. We denote clustering rule (7) of VaDA under complete data as VaDA-YX, and rule (8) with unobserved outcomes as VaDA-X. In Table 3, we fix on $K = 3$ and test the clustering performance with varied latent dimensions D . As can be seen, VaDA-YX and VaDA-X show great robustness in retrieving the ground-truth clusters in all the four scenarios. Even with insufficient latent dimensions D , the methods can provide clustering results matching well with the original grouping of the subjects. It is noticeable that there is little difference between the clustering results of VaDA-YX and VaDA-X, which is in line with expectation. In principle, the clustering rules VaDA-YX and VaDA-X differ by an integral with respect to the predictive distribution $p(\mathbf{y}_n | \mathbf{X}_n)$. If $p(\mathbf{y}_n | \mathbf{X}_n)$ peaks at the true value of $\mathbf{y}_n$, the two clustering rules will probabilistically give the same results. The architecture of VaDA can incorporate the diversity of different data generating mechanisms in the four synthetic scenarios and has obtained high prediction accuracy as shown in Tables 1 and 2. The combined results imply that the overall AEVB framework of VaDA could provide a good approximation for the true predictive distribution, which gives consistent estimators for the outcomes.

Table 2. Prediction results on the testing datasets evaluated by the mean square error with varied settings of the number of clusters K when the latent dimensions $D = 10$.

Scenario	Method	$K = 2$	$K = 3$	$K = 4$
11	VaDA *	0.071 ± 0.020	0.068 ± 0.007	0.067 ± 0.007
	VaDE+MM0	0.731 ± 0.022	0.730 ± 0.021	0.734 ± 0.021
	VaDE+MM1	0.734 ± 0.024	0.732 ± 0.022	0.737 ± 0.022
	VaDE+MM2	0.730 ± 0.022	0.729 ± 0.020	0.734 ± 0.021
	VAE+GMM+MM0	0.732 ± 0.022	0.726 ± 0.020	0.727 ± 0.020
	VAE+GMM+MM1	0.735 ± 0.024	0.729 ± 0.020	0.729 ± 0.020
	VAE+GMM+MM2	0.731 ± 0.022	0.725 ± 0.021	0.725 ± 0.021
	GMM+MM0	0.753 ± 0.033	0.787 ± 0.078	0.838 ± 0.115
	GMM+MM1	0.742 ± 0.024	0.757 ± 0.044	0.782 ± 0.061
	GMM+MM2	0.726 ± 0.021	0.726 ± 0.021	0.727 ± 0.020
01	VaDA	0.051 ± 0.007	0.053 ± 0.009	0.051 ± 0.006
	VaDE+MM0	0.114 ± 0.031	0.092 ± 0.057	0.129 ± 0.046
	VaDE+MM1	0.118 ± 0.042	0.119 ± 0.098	0.154 ± 0.085
	VaDE+MM2	0.120 ± 0.042	0.120 ± 0.097	0.162 ± 0.097
	VAE+GMM+MM0	0.120 ± 0.037	0.015 ± 0.024	0.007 ± 0.001
	VAE+GMM+MM1	0.120 ± 0.037	0.015 ± 0.024	0.007 ± 0.001
	VAE+GMM+MM2	0.120 ± 0.037	0.015 ± 0.024	0.007 ± 0.001
	GMM+MM0	0.028 ± 0.025	0.012 ± 0.019	0.007 ± 0.008
	GMM+MM1	0.028 ± 0.025	0.012 ± 0.019	0.007 ± 0.008
	GMM+MM2	0.028 ± 0.025	0.012 ± 0.019	0.007 ± 0.008
10	VaDA	0.096 ± 0.012	0.092 ± 0.004	0.093 ± 0.008
	VaDE+MM0	1.004 ± 0.026	1.004 ± 0.026	1.003 ± 0.026
	VaDE+MM1	1.005 ± 0.026	1.005 ± 0.026	1.005 ± 0.026
	VaDE+MM2	1.002 ± 0.026	1.002 ± 0.026	1.003 ± 0.026
	VAE+GMM+MM0	1.004 ± 0.026	1.005 ± 0.026	1.005 ± 0.027
	VAE+GMM+MM1	1.006 ± 0.026	1.006 ± 0.026	1.006 ± 0.027
	VAE+GMM+MM2	1.003 ± 0.026	1.003 ± 0.026	1.003 ± 0.026
	GMM+MM0	1.015 ± 0.026	1.020 ± 0.028	1.020 ± 0.028
	GMM+MM1	1.016 ± 0.026	1.022 ± 0.028	1.021 ± 0.028
	GMM+MM2	1.003 ± 0.026	1.004 ± 0.026	1.004 ± 0.026
00	VaDA	0.107 ± 0.006	0.108 ± 0.010	0.110 ± 0.015
	VaDE+MM0	0.105 ± 0.011	0.107 ± 0.016	0.104 ± 0.010
	VaDE+MM1	0.105 ± 0.011	0.107 ± 0.016	0.104 ± 0.010
	VaDE+MM2	0.105 ± 0.011	0.107 ± 0.016	0.104 ± 0.010
	VAE+GMM+MM0	0.104 ± 0.011	0.101 ± 0.012	0.097 ± 0.004
	VAE+GMM+MM1	0.104 ± 0.011	0.101 ± 0.012	0.097 ± 0.004
	VAE+GMM+MM2	0.104 ± 0.011	0.101 ± 0.012	0.097 ± 0.004
	GMM+MM0	0.069 ± 0.027	0.018 ± 0.026	0.022 ± 0.027
	GMM+MM1	0.069 ± 0.027	0.018 ± 0.026	0.022 ± 0.027
	GMM+MM2	0.069 ± 0.027	0.018 ± 0.026	0.022 ± 0.027

* Visual highlighting with gray background is applied to the results of our proposed method to facilitate quick identification and comparison with baseline methods.

Table 3. Clustering results on the testing datasets evaluated by the adjusted Rand index with varied settings of the latent dimensions D when the number of clusters $K = 3$.

Scenario	Method	$D = 2$	$D = 5$	$D = 10$	$D = 20$	$D = 100$
11	VaDA-YX *	0.986 ± 0.073	0.942 ± 0.147	0.957 ± 0.129	0.973 ± 0.102	0.914 ± 0.172
	VaDA-X	0.986 ± 0.073	0.942 ± 0.147	0.957 ± 0.129	0.973 ± 0.102	0.914 ± 0.172
	VaDE	0.381 ± 0.144	0.420 ± 0.155	0.782 ± 0.217	0.985 ± 0.079	1.000 ± 0.000
	VAE+GMM	1.000 ± 0.000	0.982 ± 0.100	1.000 ± 0.000	0.946 ± 0.165	0.930 ± 0.146
	GMM	—	—	—	—	0.816 ± 0.265
01	VaDA-YX	1.000 ± 0.000	1.000 ± 0.000	0.972 ± 0.105	0.987 ± 0.071	0.913 ± 0.174
	VaDA-X	1.000 ± 0.000	1.000 ± 0.000	0.972 ± 0.105	0.987 ± 0.071	0.913 ± 0.174
	VaDE	0.309 ± 0.159	0.412 ± 0.136	0.731 ± 0.201	1.000 ± 0.001	1.000 ± 0.000
	VAE+GMM	1.000 ± 0.000	1.000 ± 0.000	0.945 ± 0.168	0.964 ± 0.138	0.921 ± 0.169
	GMM	—	—	—	—	0.816 ± 0.265
10	VaDA-YX	0.942 ± 0.147	0.956 ± 0.132	0.957 ± 0.131	0.956 ± 0.133	0.972 ± 0.104
	VaDA-X	0.942 ± 0.147	0.956 ± 0.132	0.957 ± 0.131	0.956 ± 0.133	0.972 ± 0.104
	VaDE	0.611 ± 0.273	0.540 ± 0.247	0.590 ± 0.170	0.592 ± 0.292	0.672 ± 0.271
	VAE+GMM	0.802 ± 0.265	0.853 ± 0.248	0.816 ± 0.265	0.908 ± 0.209	0.852 ± 0.249
	GMM	—	—	—	—	0.914 ± 0.176
00	VaDA-YX	0.930 ± 0.157	0.940 ± 0.147	1.000 ± 0.000	0.986 ± 0.077	0.942 ± 0.148
	VaDA-X	0.930 ± 0.157	0.942 ± 0.148	1.000 ± 0.000	0.986 ± 0.077	0.942 ± 0.148
	VaDE	0.655 ± 0.216	0.573 ± 0.253	0.631 ± 0.212	0.651 ± 0.218	0.580 ± 0.207
	VAE+GMM	0.778 ± 0.260	0.816 ± 0.264	0.778 ± 0.276	0.743 ± 0.280	0.926 ± 0.192
	GMM	—	—	—	—	0.912 ± 0.179

* Visual highlighting with gray background is applied to the results of our proposed method to facilitate quick identification and comparison with baseline methods.

Table 4. Clustering results on the testing datasets evaluated by the adjusted Rand index with varied settings of the number of clusters K when the latent dimensions $D = 10$.

Scenario	Method	$K = 2$	$K = 3$	$K = 4$
11	VaDA-YX *	0.569 ± 0.023	0.957 ± 0.129	0.984 ± 0.084
	VaDA-X	0.569 ± 0.023	0.957 ± 0.129	0.984 ± 0.084
	VaDE	0.533 ± 0.058	0.782 ± 0.217	0.455 ± 0.141
	VAE+GMM	0.567 ± 0.022	1.000 ± 0.000	0.871 ± 0.081
	GMM	0.567 ± 0.025	0.816 ± 0.265	0.875 ± 0.178
01	VaDA-YX	0.574 ± 0.027	0.972 ± 0.105	0.986 ± 0.073
	VaDA-X	0.574 ± 0.027	0.972 ± 0.105	0.986 ± 0.073
	VaDE	0.564 ± 0.033	0.731 ± 0.201	0.561 ± 0.151
	VAE+GMM	0.577 ± 0.025	0.945 ± 0.168	0.890 ± 0.088
	GMM	0.567 ± 0.025	0.816 ± 0.265	0.875 ± 0.178
10	VaDA-YX	0.569 ± 0.015	0.957 ± 0.131	1.000 ± 0.000
	VaDA-X	0.569 ± 0.015	0.957 ± 0.131	1.000 ± 0.000
	VaDE	0.570 ± 0.024	0.590 ± 0.170	0.616 ± 0.232
	VAE+GMM	0.572 ± 0.019	0.816 ± 0.265	0.823 ± 0.173
	GMM	0.569 ± 0.019	0.914 ± 0.176	0.877 ± 0.200
00	VaDA-YX	0.572 ± 0.021	1.000 ± 0.000	0.957 ± 0.128
	VaDA-X	0.572 ± 0.021	1.000 ± 0.000	0.957 ± 0.128
	VaDE	0.577 ± 0.023	0.631 ± 0.212	0.636 ± 0.147
	VAE+GMM	0.571 ± 0.023	0.778 ± 0.276	0.820 ± 0.181
	GMM	0.568 ± 0.019	0.912 ± 0.179	0.873 ± 0.198

* Visual highlighting with gray background is applied to the results of our proposed method to facilitate quick identification and comparison with baseline methods.

The clustering quality of VaDE deteriorates considerably as the latent dimensions D is reduced in Scenarios 11 and 01 (with non-linear marginal effects). Comparison between the results of VaDA and VaDE highlights the expressiveness provided by jointly modeling the marginal regression and clustering in the latent space. The VaDA approach exploits the interactions between the covariate data and the response data, which could calibrate the clustering bias with inadequate covariate information. The representations learned by VAE improve the clustering from GMM in Scenarios 11 and 01 (with non-linear marginal effects), but degrade the performance in Scenarios 10 and 00 (with linear marginal effects).

In Table 4, the clustering performances with varied K values when fixing on $D = 10$ are compared. Again, the approaches VaDA-YX and VaDA-X show better generalization performance compared with the other methods. It is noticeable that there is a dramatic decreasing on the ARI value when K is smaller than the ground-truth. This observation is conformable with the property of ARI as two subjects originally separated in two classes now could have to be inserted into one cluster. Generally, the performances of VAE+GMM and GMM arrive their optima at the ground-truth K value.

Overall, VaDA can achieve superior clustering performance while hardly sacrificing the prediction accuracy. It is robust and shows high generalization performance in those misspecified cases, which provides us with great confidence in applying the VaDA approach for longitudinal data analysis in those real-world settings.

5. Application on CelebA Dataset

In this section, we consider a novel application of the proposed method on the CelebA dataset. CelebA is a large-scale face attribute dataset containing 202,599 face images from 10,177 celebrities. The 40 binary attributes annotations per image provide rich source of variability to establish a comprehensive longitudinal study.

We focus on the recognition task on whether a celebrity wears sunglasses from the face image, which constitutes a supervised learning problem with a binary response variable. To form a longitudinal dataset, two face images for each celebrity were randomly chosen: one was wearing sunglasses and the other was not (the order was shuffled). Therefore, there exists apparent within-subject dependence. It can be expected that correctly modeling the relationship between the outcomes of a subject will enhance the performance of representation learning, clustering and image generating. There were totally 3169 celebrities (1199 female celebrities and 1970 male celebrities) who had both the face images with sunglasses and without sunglasses. We used the images from 1200 male celebrities to balance the gender ratio. Finally, the number of subjects in the longitudinal study was $N = 2399$, the length of measurements was $L = 2$ and there were totally $N \times L = 4798$ observations. The feature data were organized by cropping and scaling the corresponding face images to 64×64 pixels as in [35], which therefore had an input tensor shape $(64, 64, 3)$.

The methods VaDA, VaDE+MM and VAE+GMM+MM, which can realize prediction, clustering and representation learning, were implemented. We did not consider the baseline method GMM+MM here as it can hardly be implemented with efficiency when the data is high-dimensional. While VaDE and VAE inspect the face images in the CelebA longitudinal dataset as fully independent samples, the VaDA method treats the paired image data. As we considered a binary outcome, the link function in the marginal models for VaDE+MM and VAE+GMM+MM was set as “logit” [29]. We also changed the reconstruction term about y_n in the ELBO of VaDA to the negative logistic loss function (cross-entropy) [36] to match with the type of outcome. As the values for the latent dimensions D and the number of clusters K were unknown a priori, we took experiments on $D \in \{64, 128, 256, 512\}$ and $K \in \{2, 3, 4\}$, which were conventional settings. The training/testing ratio was set as 4/1. Again, we used the Adam method for optimization with a learning rate 0.0005. The batch

size for VaDA was set as 128 and for VaDE and VAE it was set as $128 \times L = 256$. For all the three methods, the number of training epochs was set as 100. To accelerate the progress of training, we used the off-the-shelf network architectures from the improved VAE model in [35] for the X-decoder and X-encoder, which had been used to process the CelebA dataset and achieved good performance in representation learning. Appendix D provides details for the network architectures. Again, no pretraining was conducted. Image data were scaled into $[0, 1]$ by dividing the maximum pixel value 255.

5.1. Prediction

Table 5 compares the prediction accuracy of different methods on the training/testing dataset, evaluated through the value of classification error rate. VaDA shows a high generalization performance with various settings of D and K . It fits well to the training data while making even better prediction on the testing data. Oppositely, VaDE+MM and VAE+GMM+MM show a sign of over-fitting with large D values. Specifically, the prediction accuracy of VaDE+MM or VAE+GMM+MM is gradually improved on the training data but worsen on the testing data as the assumed latent dimensions D increases. In addition, changing the “working” correlation assumption in the two methods does not save the performance. They even perform better when the independence assumption (MM0) is made. Through the following experiments, we will see that the deteriorated performance could largely come from fitting the cluster-specific marginal models, which are specified and fitted separately from the representation learning stage in VaDE+MM and VAE+GMM+MM.

Table 5. Prediction results on the CelebA training/testing dataset evaluated by the classification error rate with varied settings of the latent dimensions D and the number of clusters K .

Dimensions	Method	$K = 2$	$K = 3$	$K = 4$
$D = 64$	VaDA *	0.102/0.070	0.098/0.067	0.097/0.071
	VaDE+MM0	0.155/0.180	0.146/0.168	0.143/0.191
	VaDE+MM1	0.172/0.177	0.162/0.180	0.160/0.212
	VAE+GMM+MM0	0.149/0.189	0.153/0.185	0.149/0.187
	VAE+GMM+MM1	0.482/0.479	0.364/0.406	0.167/0.210
$D = 128$	VaDA	0.097/0.077	0.108/0.082	0.095/0.060
	VaDE+MM0	0.129/0.175	0.134/0.188	0.106/0.210
	VaDE+MM1	0.155/0.184	0.165/0.208	0.161/0.239
	VAE+GMM+MM0	0.131/0.174	0.122/0.194	0.114/0.206
	VAE+GMM+MM1	0.195/0.233	0.129/0.187	0.351/0.397
$D = 256$	VaDA	0.090/0.069	0.090/0.066	0.098/0.069
	VaDE+MM0	0.102/0.208	0.094/0.206	0.090/0.228
	VaDE+MM1	0.128/0.231	0.124/0.213	0.092/0.229
	VAE+GMM+MM0	0.097/0.204	0.063/0.268	0.040/0.313
	VAE+GMM+MM1	0.129/0.234	0.070/0.271	0.039/0.313
$D = 512$	VaDA	0.094/0.070	0.103/0.077	0.098/0.080
	VaDE+MM0	0.045/0.350	0.000/0.361	0.048/0.337
	VaDE+MM1	0.045/0.359	0.000/0.370	0.068/0.352
	VAE+GMM+MM0	0.051/0.324	0.000/0.372	0.000/0.419
	VAE+GMM+MM1	0.051/0.324	0.000/0.377	0.000/0.421

* Visual highlighting with gray background is applied to the results of our proposed method to facilitate quick identification and comparison with baseline methods.

5.2. Clustering

When evaluating the clustering results of CelebA data, we have no prior known subject label to match with as what we have done in the synthetic experiments. Recently, a study

demonstrates that visual concepts such as face pose and gender could be manipulated by simple vector arithmetic in the latent space, giving rise to the definition of concept vector for image editing [37]. This strategy provides us a way to visually investigate properties of different clusters by identifying the concept vector from one cluster to another. Specifically, we encode each observed image x_{nl} to its latent representation $\hat{z}_{nl}$ through the X-encoder and take averaging over the latent representations within each cluster. Denote the obtained latent vector in cluster k as $\bar{z}_k$. Then, the concept vector from cluster k' to k can be defined by $\bar{z}_k - \bar{z}_{k'}$, which indicates a direction in the latent space encoding certain variations in the image data from cluster k' to k . If we map $\hat{z}_{nl} + \alpha(\bar{z}_k - \bar{z}_{k'})$ back to the image space through the X-decoder, an increasing of the value α will edit the image x_{nl} following the direction from cluster k' to k . While clustering results of high quality can provide meaningful concept vectors, good generalization performance of X-encoder and X-decoder is also necessary to ensure a satisfactory manifestation of the editing effects.

In Figures 2–4, we provide results of image editing on face images from five subjects as examples. The image editing manifested here is based on the clustering and representation learning results of VaDA. Given the space limitations, we have placed the image editing results based on VaDE and VAE+GMM in the Supplementary Materials (Section S2). To make consistent comparison with the other methods, the clustering rule VaDA-X is applied.



Figure 2. Image editing by concept vector between each pair of clusters based on the results of VaDA with $K = 2$. The gradient α takes value in $[-2, 2]$. The results at $\alpha = 0$ (enclosed by green box) correspond to the reconstruction of original images.

Figure 2 corresponds to the results when the subjects are separated into two clusters. We can see smooth transition between dark and light backgrounds from cluster $k = 0$ to $k = 1$. It is also noticeable that little by little the hair becomes shorter while other facial attributes are largely retained.

Figure 3 manifests the image editing results when three clusters are assumed. Again, the dominant variations are from the background and the hair. From cluster $k = 0$ to $k = 1$, the background becomes lighter, the hair gets shorter and the hair color is slightly deepened. From cluster $k = 1$ to $k = 2$, reversed changes are presented. Nevertheless, the left side of image editing from $k = 0$ to $k = 1$ shows a warmer color palette than the right side of image editing from $k = 1$ to $k = 2$. Interestingly, this subtle difference can be validated by image editing from $k = 2$ to $k = 0$, where the overall color tone shifts towards warmer hues. In addition, there also exhibits slight variation on the fringe. While the fringe gets thinner from $k = 1$ to $k = 2$, it becomes heavier from $k = 2$ to $k = 0$. This therefore corresponds to the little change of fringe from cluster $k = 0$ to $k = 1$.

Figure 4 shows the image editing results decoded by the six concept vectors from a four-cluster separation of the subjects. Besides the background changes and the hair length and color transitions, we detect two additional meaningful variation axes: age and face

pose. It can be seen clearly that there are transitions on age from cluster $k = 0$ to $k = 1$, from $k = 0$ to $k = 3$, from $k = 2$ to $k = 1$ and from $k = 2$ to $k = 3$ with the appearance getting older. The pattern of transition from cluster $k = 0$ to $k = 1$ shares similarity with that from $k = 0$ to $k = 3$, where the background becomes darker along with the growing age. Nevertheless, difference exists between cluster $k = 1$ and $k = 3$. The appearance gets old little by little from $k = 3$ to $k = 1$, which completes a circle from $k = 0$ to $k = 3$, from $k = 3$ to $k = 1$ and finally from $k = 1$ to $k = 0$ with the age getting older and older and then returning young. The similar circle can be figured out from cluster $k = 2$ to $k = 3$, from $k = 3$ to $k = 1$ and finally from $k = 1$ to $k = 2$ but with a reversed transition on the darkness of background. Additionally, smooth transitions on the face pose can be found from cluster $k = 1$ to $k = 0$, from $k = 1$ to $k = 2$ and from $k = 1$ to $k = 3$, where the faces gradually turn from profile to full front. This detection illustrates that the face images from cluster $k = 1$ could be characterized as facing sideways. We summarize the transitions between different clusters in Figure 5.



Figure 3. Image editing by concept vector between each pair of clusters based on the results of VaDA with $K = 3$. The gradient α takes value in $[-2.5, 2.5]$. The results at $\alpha = 0$ (enclosed by green box) correspond to the reconstruction of original images.

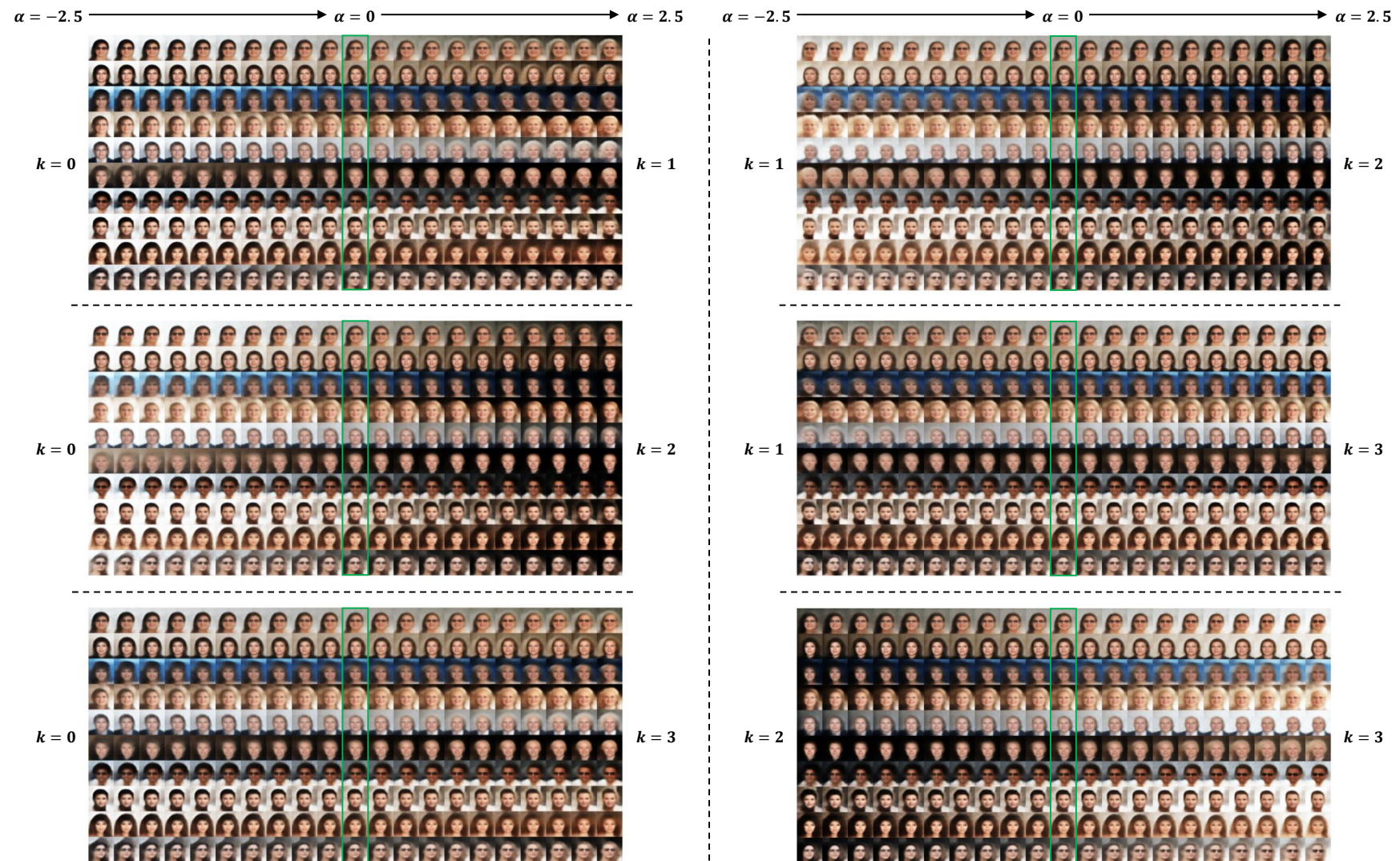


Figure 4. Image editing by concept vector between each pair of clusters based on the results of VaDA with $K = 4$. The gradient α takes value in $[-2.5, 2.5]$. The results at $\alpha = 0$ (enclosed by green box) correspond to the reconstruction of original images.

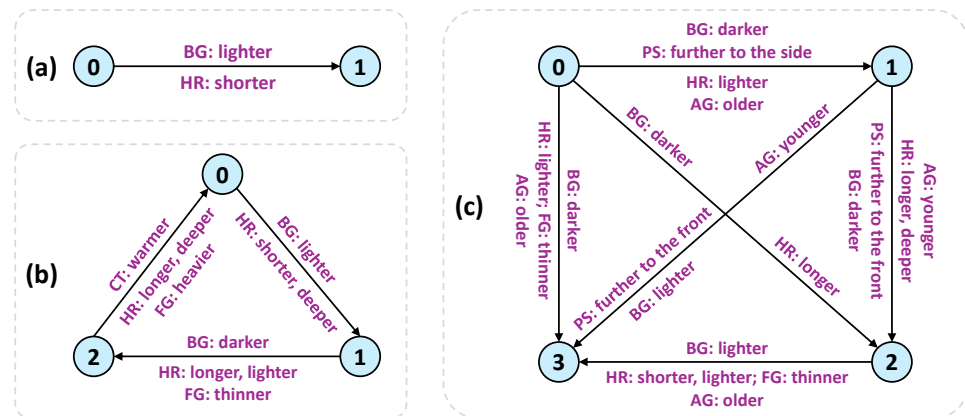


Figure 5. Dominant visual transitions directed by the concept vectors from VaDA when $K = 2$ (a), $K = 3$ (b) and $K = 4$ (c). The blue nodes represent the clusters. We use BG, CT, PS, HR, FG and AG as an abbreviation for background, color tone, face pose, hair, fringe and age, respectively.

Compared to the proposed method, the clustering performances of VaDE and VAE+GMM are largely unsatisfactory. The VaDE method which performs representation learning and clustering simultaneously gives distinguishable clusters under different settings of the number of clusters. However, it has limited generalization capability as shown in case $K = 3$ (Figure S2) and $K = 4$ (Figure S3). The approach to cluster $k = 0$ produces images that gradually blur. Generally, the latent space learned by VAE is not responsive to the clustering result of GMM (Figures S4–S6). There is nearly imperceptible variation when changing the α value.

5.3. Image Reconstruction and Generation

The image reconstruction results are compared between the methods VaDA, VaDE and VAE. We map the testing images to the latent space using the X-encoder and decode them back to the image space using the X-decoder. Ten reconstruction examples are presented in Figure 6. The three methods sharing the same encoder and decoder structures exhibit comparable performance and there is nearly no difference that can be identified given the latent dimensions D and the number of clusters K . While varying the setting of K has little impact on the image reconstruction, increasing the latent dimensions D gradually smooths the images. The overall results demonstrate the effectiveness of the X-encoder and X-decoder established in our experiments, which can capture the main properties of the image data.

Figure 7 shows the face images generated by VaDA, VaDE and VAE with $K = 2$. Random face images were obtained by decoding the latent vectors drawn from the estimated latent space. Overall, compared with VaDA, the methods VaDE and VAE are less advantaged in out-of-sample generalization. Even though the generated images by VaDA are not perfect and tend to be blurry on the hair, they are more realistic on the face color and have compatible and harmonious overall appearance than the images generated by VaDE and VAE. VaDE is more likely to produce weird faces in our experiments and the images generated by VAE tend to be blurrier though the overall spatial face structures are preserved. Nevertheless, the three methods show improved data generating performance as the latent dimensions D is increased. The out-of-sample generalization performance relies heavily on correct modeling of the latent space structures. The outperforming of VaDA is in line with expectation as it has a better-structured latent space allowing within-subject dependence and unifying the latent information from outcome and features.

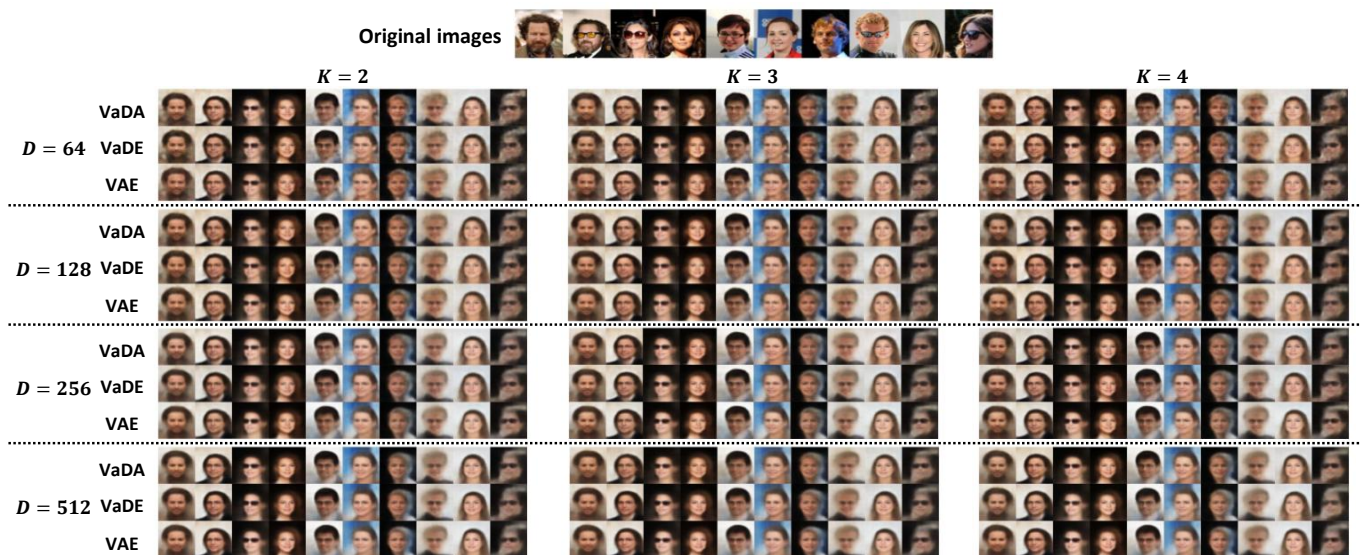


Figure 6. Image reconstruction by method VaDA, VaDE and VAE under different settings of D and K .

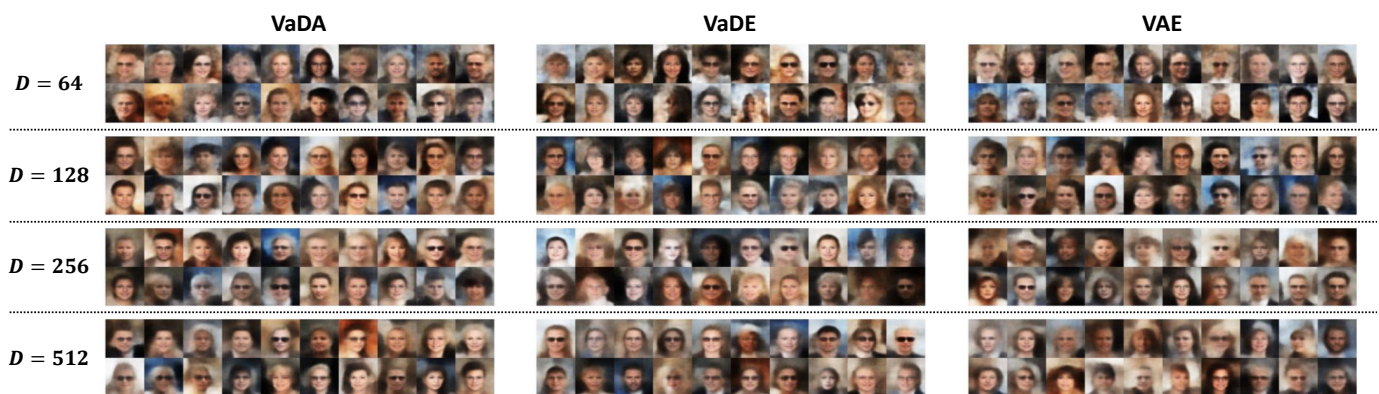


Figure 7. Image generation by method VaDA, VaDE and VAE under different settings of latent dimensions D given $K = 2$.

6. Conclusions

In this paper, we developed a generative auto-encoding approach to longitudinal data analysis. VaDA retains the strengths of VAE to learn well-structured latent space and is an enhanced version of VaDE to accommodate the nature of longitudinal data. Moreover, it goes beyond the GEEs based marginal modeling approach by forging a deep “alliance” across measurements to incorporate more complicated dependence structures. VaDA integrates prediction, clustering and representation learning in a unified AEVB framework, which alleviates the loss of information between different stages and fortifies interaction and coordination across all parts of the model.

Synthetic and real-world data experiments were implemented. The results demonstrated the robustness and effectiveness of the proposed method, which achieved high prediction accuracy and showed excellent ability in recovering the underlying grouping of the data under varied assumptions of data generating mechanism. The VaDA approach well captured the latent structure of CelebA longitudinal data, where we obtained meaningful clustering results of the face images. Moreover, it exhibited high generalization capability. The generated face images have more realistic and harmonious appearance than those from VAE or VaDE.

Nevertheless, there is still room further improvements. In the experiments, we have used given decoder and encoder network structures. For synthetic data, the networks

similar to the ones in the data generating process were adopted. For CelebA data, we used the available convolution neural network structures established in previous studies for face images and a simple dense network to model the relationship between outcomes. A pilot study was conducted to investigate the impact of redundancy in encoder and decoder network architectures (see Section S3 in Supplementary Materials). While the redundancy was found to degrade the performance of VaDA, incorporating “dropout” layers [38] could mitigate this issue, resulting in a calibrated model. These findings highlight the need for a more comprehensive exploration of network structures in order to optimize the performance of VaDA for real-world applications.

Another potential enhancement of the method is to weight the different variational loss terms like the practice in [39], especially to consider the balance between reconstruction loss terms on response and covariates. In addition, a thorough analysis of the identifiability of VaDA, while beyond the scope of this work, remains an important theoretical question for further research. The study in [40] has demonstrated the identifiability properties of deep generative models like VAE and VaDE up to an affine transformation under moderate conditions. Our future work should explore the potentials of extending these results to VaDA to strengthen its theoretical foundation.

In current work, we have not considered the handling of missing data on covariates or response, which frequently arise in longitudinal study. Improper treatment of missingness could introduce bias under certain missing data mechanisms [41]. Nevertheless, the generative modeling framework of VaDA can be extended immediately to the missing data cases, where missing data mechanism can be integrated naturally into the ELBO to derive a systematic treatment. Such enhancement will strengthen the robustness and generalizability of current model.

While the proposed VaDA approach processes high flexibility in handling the high-dimensional longitudinal data, there are still limitations that it is only suitable to datasets with equal-length measurements across subjects (these datasets often appear in well scheduled and organized longitudinal experiments). Nevertheless, the inference framework of VaDA provides a solid foundation for combining the efforts of VAE with alternative classical methods for other types of longitudinal data. A potential is to go beyond the generalized neural network mixed effects model in [23] by extending it to a full generative framework, which is a part of our ongoing work.

The present study introduces a novel longitudinal perspective to the widely recognized CelebA dataset by reorganizing its structure and applying the VaDA method for improved representation learning, clustering and image generation. This perspective could offer a meaningful contribution to the pattern discovery based on CelebA and other benchmark vision datasets. Future work could explore the applicability of the method to longitudinal data with more complex and naturally evolving spatio-temporal dynamics such as those in clinical or behavioral studies. This may also involve extending the proposed framework to handle challenges including missing data and irregular length of measurements as noted in our earlier discussion, therefore further enhancing the VaDA’s practical utility in more real-world longitudinal studies.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/e28010113/s1>. Figure S1: image editing by concept vector between each pair of clusters based on the results of VaDE with $K = 2$; Figure S2: image editing by concept vector between each pair of clusters based on the results of VaDE with $K = 3$; Figure S3: image editing by concept vector between each pair of clusters based on the results of VaDE with $K = 4$; Figure S4: image editing by concept vector between each pair of clusters based on the results of VAE+GMM with $K = 2$; Figure S5: image editing by concept vector between each pair of clusters based on the results of VAE+GMM with $K = 3$; Figure S6: image editing by concept vector between

each pair of clusters based on the results of VAE+GMM with $K = 4$; Figure S7: architectures of encoder and decoder networks in VaDA for synthetic data with redundancy; Figure S8: architectures of encoder and decoder networks in VaDA for synthetic data with dropout; Table S1: prediction results on the training datasets evaluated by the mean square error with varied settings of the latent dimensions D when the number of clusters $K = 3$; Table S2: prediction results on the training datasets evaluated by the mean square error with varied settings of the number of clusters K when the latent dimensions $D = 10$; Table S3: clustering results on the training datasets evaluated by the adjusted Rand index with varied settings of the latent dimensions D when the number of clusters $K = 3$; Table S4: clustering results on the training datasets evaluated by the adjusted Rand index with varied settings of the number of clusters K when the latent dimensions $D = 10$; Table S5: the performance of VaDA with normal, redundant and dropout included network structures evaluated on the metrics of adjusted Rand index for clustering results from VaDA-YX (ARI-YX) and from VaDA-X (ARI-X), mean square error for prediction accuracy (MSE).

Author Contributions: Conceptualization, W.X. and Y.N.; methodology, S.F.; software, S.F.; validation, S.F. and W.X.; formal analysis, S.F.; writing—original draft preparation, S.F.; writing—review and editing, W.X.; visualization, S.F.; supervision, Y.N.; funding acquisition, Y.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China, grant number 2020YFA0713603, and the National Natural Science Foundation of China, grant number 11971386.

Data Availability Statement: The original CelebFaces Attributes (CelebA) Dataset was downloaded from the webpage: <https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html> (accessed on 12 July 2025). The deep learning methods VaDA, VaDE and VAE were implemented based on the programming language Python (version: 3.12.4) with the backbone “TensorFlow”. The marginal model, the GMM-based clustering and the PCA were implemented based on the programming language R (version: 4.5.1) using the packages “geepack” and “elasticnet”. The codes for the experimental study are available upon request from the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Derive Terms in KL Divergence

In this section, we explain how the final form of $\mathbf{D}_{\text{KL}}[q(\mathbf{u}_n, \mathbf{Z}_n, c_n | \mathbf{y}_n, \mathbf{X}_n) || p(\mathbf{u}_n, \mathbf{Z}_n, c_n)]$ is derived. Following the definition of KL divergence, we have

$$\begin{aligned} \mathbf{D}_{\text{KL}}[q(\mathbf{u}_n, \mathbf{Z}_n, c_n | \mathbf{y}_n, \mathbf{X}_n) || p(\mathbf{u}_n, \mathbf{Z}_n, c_n)] \\ = -\mathbb{E}[\log p(\mathbf{u}_n | \mathbf{Z}_n, c_n)] - \mathbb{E}[\log p(\mathbf{Z}_n | c_n)] - \mathbb{E}[\log p(c_n)] \\ + \mathbb{E}[\log q(\mathbf{u}_n | \mathbf{y}_n)] + \mathbb{E}[\log q(\mathbf{Z}_n | \mathbf{X}_n)] + \mathbb{E}[\log q(c_n | \mathbf{y}_n, \mathbf{X}_n)], \end{aligned} \quad (\text{A1})$$

where the expectation is taken w.r.t. $q(\mathbf{u}_n, \mathbf{Z}_n, c_n | \mathbf{y}_n, \mathbf{X}_n)$ with assumed decomposition given in (2). Computation of the terms in (A1) is given as follows.

$$(1) \mathbb{E}[\log p(\mathbf{u}_n | \mathbf{Z}_n, c_n)]$$

$$\begin{aligned} \mathbb{E}[\log p(\mathbf{u}_n | \mathbf{Z}_n, c_n)] &= \sum_{k=1}^K \tilde{\pi}_k \mathbb{E}[\log p(\mathbf{u}_n | \mathbf{Z}_n, c_n = k)] \\ &= \sum_{k=1}^K \tilde{\pi}_k \sum_{l=1}^L \mathbb{E}[\log p(u_{nl} | z_{nl}, c_n = k)] \\ &= \sum_{k=1}^K \tilde{\pi}_k \sum_{l=1}^L \mathbb{E}[\log \mathcal{N}(u_{nl}; \beta_{k,0} + \mathbf{z}_{nl}^T \boldsymbol{\beta}_k, \sigma_{u,k}^2)] \\ &= \sum_{k=1}^K \tilde{\pi}_k \sum_{l=1}^L \mathbb{E}\left[-\frac{1}{2} \log \sigma_{u,k}^2 - \frac{1}{2\sigma_{u,k}^2} (u_{nl} - \beta_{k,0} - \mathbf{z}_{nl}^T \boldsymbol{\beta}_k)^2\right] + \text{const}. \end{aligned} \quad (\text{A2})$$

As $q(u_{nl}|\mathbf{y}_n)$ and $q(\mathbf{z}_{nl}|\mathbf{x}_{nl})$ are both Gaussian distribution, i.e.,

$$\begin{aligned} q(u_{nl}|\mathbf{y}_n) &= \mathcal{N}(u_{nl}; \tilde{\mu}_{u,l}, \tilde{\sigma}_{u,l}^2), \\ q(\mathbf{z}_{nl}|\mathbf{x}_{nl}) &= \mathcal{N}(\mathbf{z}_{nl}; \tilde{\boldsymbol{\mu}}_z, \text{diag}(\tilde{\sigma}_z^2)), \end{aligned}$$

we have

$$\begin{aligned} \mathbf{E}(u_{nl}) &= \tilde{\mu}_{u,l}, \quad \mathbf{E}(u_{nl}^2) = \tilde{\mu}_{u,l}^2 + \tilde{\sigma}_{u,l}^2, \\ \mathbf{E}(\mathbf{z}_{nl}) &= \tilde{\boldsymbol{\mu}}_z, \quad \mathbf{E}(\mathbf{z}_{nl}\mathbf{z}_{nl}^T) = \tilde{\boldsymbol{\mu}}_z\tilde{\boldsymbol{\mu}}_z^T + \text{diag}(\tilde{\sigma}_z^2). \end{aligned}$$

Using these results to compute (A2), it follows

$$\mathbf{E}[\log p(\mathbf{u}_n|\mathbf{Z}_n, c_n)] = \sum_{k=1}^K \tilde{\pi}_k \sum_{l=1}^L \left\{ -\frac{1}{2} \log \sigma_{u,k}^2 - \frac{1}{2\sigma_{u,k}^2} \left[(\tilde{\mu}_{u,l} - \beta_{k,0} - \tilde{\boldsymbol{\mu}}_z^T \boldsymbol{\beta}_k)^2 + \tilde{\sigma}_{u,l}^2 + \boldsymbol{\beta}_k^T \text{diag}(\tilde{\sigma}_z^2) \boldsymbol{\beta}_k \right] \right\} + \text{const.}$$

$$(2) \mathbf{E}[\log p(\mathbf{Z}_n|c_n)]$$

$$\begin{aligned} \mathbf{E}[\log p(\mathbf{Z}_n|c_n)] &= \sum_{k=1}^K \tilde{\pi}_k \mathbf{E}[\log p(\mathbf{Z}_n|c_n = k)] \\ &= \sum_{k=1}^K \tilde{\pi}_k \sum_{l=1}^L \mathbf{E}[\log p(\mathbf{z}_{nl}|c_n = k)] \\ &= \sum_{k=1}^K \tilde{\pi}_k \sum_{l=1}^L \sum_{d=1}^D \mathbf{E}[\log \mathcal{N}(z_{nld}; \mu_{k,d}, \sigma_{k,d}^2)] \\ &= \sum_{k=1}^K \tilde{\pi}_k \sum_{l=1}^L \sum_{d=1}^D \mathbf{E} \left[-\frac{1}{2} \log \sigma_{k,d}^2 - \frac{1}{2\sigma_{k,d}^2} (z_{nld} - \mu_{k,d})^2 \right] + \text{const.} \end{aligned}$$

Using $\mathbf{E}(z_{nld}) = \tilde{\mu}_{z,d}$ and $\mathbf{E}(z_{nld}^2) = \tilde{\mu}_{z,d}^2 + \tilde{\sigma}_{z,d}^2$, we have

$$\mathbf{E}[\log p(\mathbf{Z}_n|c_n)] = \sum_{k=1}^K \tilde{\pi}_k \sum_{l=1}^L \sum_{d=1}^D \left\{ -\frac{1}{2} \log \sigma_{k,d}^2 - \frac{1}{2\sigma_{k,d}^2} \left[(\tilde{\mu}_{z,d} - \mu_{k,d})^2 + \tilde{\sigma}_{z,d}^2 \right] \right\} + \text{const.}$$

$$(3) \mathbf{E}[\log q(\mathbf{u}_n|\mathbf{y}_n)]$$

Using Lemma 1 of Appendix B in [26], we can directly obtain

$$\mathbf{E}[\log q(\mathbf{u}_n|\mathbf{y}_n)] = -\frac{1}{2} \sum_{l=1}^L \log \tilde{\sigma}_{u,l}^2 + \text{const.}$$

$$(4) \mathbf{E}[\log q(\mathbf{Z}_n|\mathbf{X}_n)]$$

Using again the Lemma 1 of Appendix B in [26], we obtain

$$\mathbf{E}[\log q(\mathbf{Z}_n|\mathbf{X}_n)] = \sum_{l=1}^L \mathbf{E}[\log q(\mathbf{z}_{nl}|\mathbf{x}_{nl})] = -\frac{1}{2} \sum_{l=1}^L \sum_{d=1}^D \log \tilde{\sigma}_{z,d}^2 + \text{const.}$$

$$(5) \mathbf{E}[\log p(c_n)] \text{ and } \mathbf{E}[\log q(c_n|\mathbf{y}_n, \mathbf{X}_n)]$$

$$\mathbf{E}[\log p(c_n)] = \sum_{k=1}^K \tilde{\pi}_k \log \pi_k,$$

$$\mathbf{E}[\log q(c_n|\mathbf{y}_n, \mathbf{X}_n)] = \sum_{k=1}^K \tilde{\pi}_k \log \tilde{\pi}_k.$$

Appendix B. The Evolution of Model Structures

We demonstrate the evolutionary pathways from VAE to VaDE and from VaDE to VaDA in Figure A1. In VAE and VaDE, the sample points are assumed mutually independent. While VAE models the latent representation using the Gaussian distribution, VaDE introduces further the latent class variable to realize clustering analysis. Therefore, the latent space of VaDE forms as a Gaussian Mixture Model. VaDA is a model for longitudinal data, where we consider the dependence relationships between observations. There are two modifications from the VaDE model. Firstly, the observations belonging to a subject share the same class label. Meanwhile, they are unified through a longitudinal neural network structure to produce the outcomes.

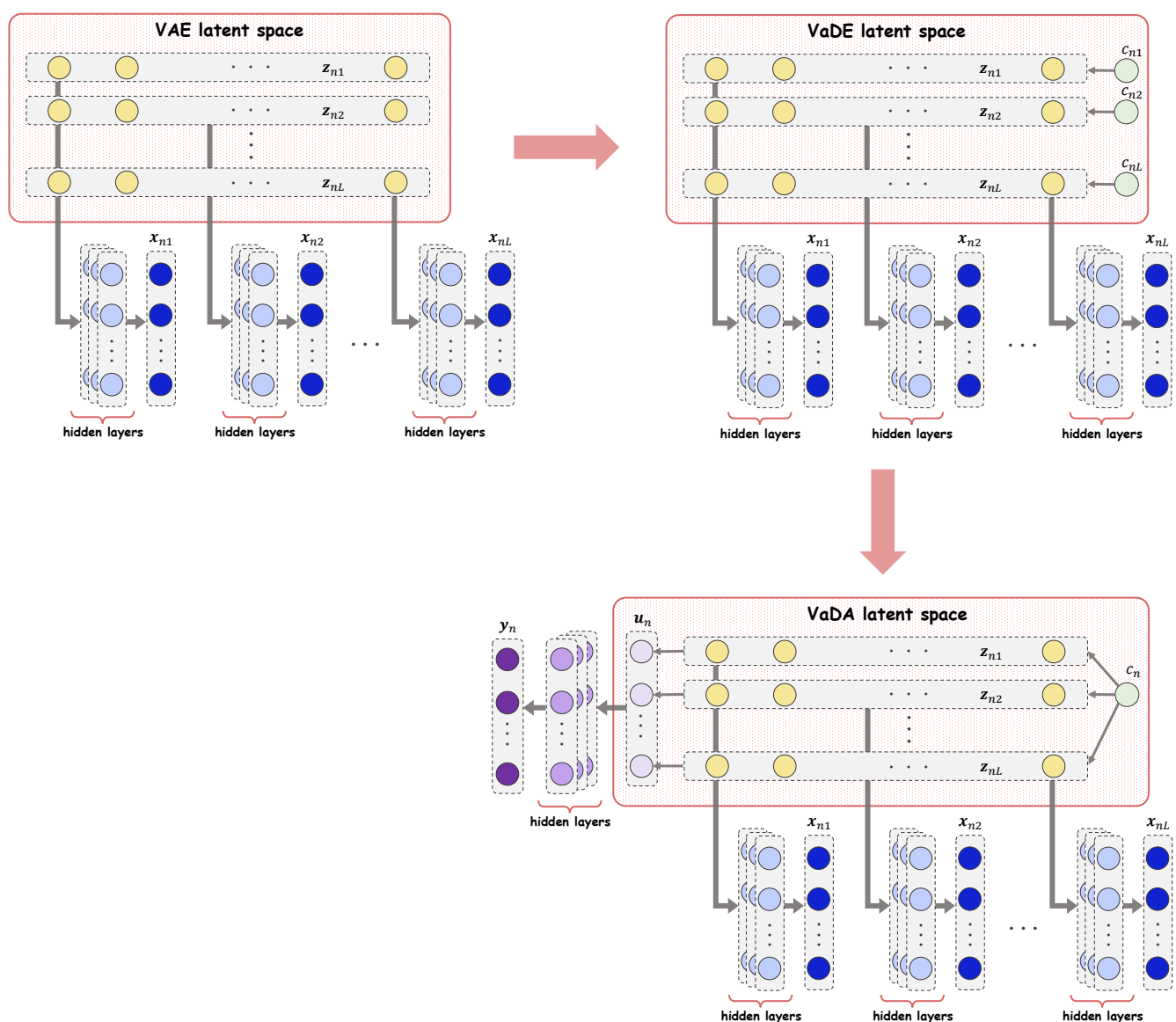


Figure A1. The evolution of model structures between the standard VAE, VaDE and VaDA.

Differences between the ELBOs of standard VAE, VaDE and VaDA are illustrated in Table A1.

Table A1. Comparison between the ELBOs of standard VAE, VaDE and VaDA.

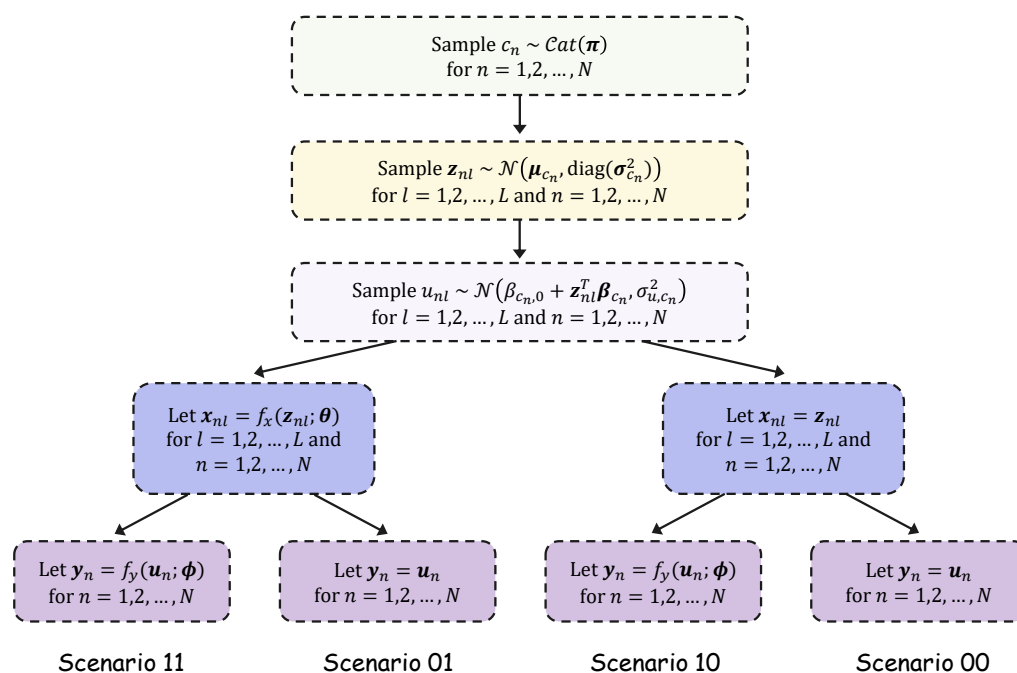
Model	Reconstruction Term	KL Divergence
VAE	$\sum_{l=1}^L \mathbf{E}_{q(\mathbf{z}_{nl} \mathbf{x}_{nl})} [\log p(\mathbf{x}_{nl} \mathbf{z}_{nl})]$	$\sum_{l=1}^L \mathbf{D}_{\text{KL}} [q(\mathbf{z}_{nl} \mathbf{x}_{nl}) p(\mathbf{z}_{nl})]$
VaDE	$\sum_{l=1}^L \mathbf{E}_{q(\mathbf{z}_{nl} \mathbf{x}_{nl})} [\log p(\mathbf{x}_{nl} \mathbf{z}_{nl})]$	$\sum_{l=1}^L \mathbf{D}_{\text{KL}} [q(\mathbf{z}_{nl}, c_{nl} \mathbf{x}_{nl}) p(\mathbf{z}_{nl}, c_{nl})]$
VaDA	$\sum_{l=1}^L \mathbf{E}_{q(\mathbf{z}_{nl} \mathbf{x}_{nl})} [\log p(\mathbf{x}_{nl} \mathbf{z}_{nl})] + \mathbf{E}_{q(\mathbf{u}_n \mathbf{y}_n)} [\log p(\mathbf{y}_n \mathbf{u}_n)]$	$\mathbf{D}_{\text{KL}} [q(\mathbf{u}_n, \mathbf{Z}_n, c_n \mathbf{y}_n, \mathbf{X}_n) p(\mathbf{u}_n, \mathbf{Z}_n, c_n)]$

Appendix C. Generating Process of Synthetic Data

Herein, we provide details on the generation of synthetic longitudinal data used in our experiments. The whole process is very similar to the generative process assumed by the proposed VaDA. We first illustrate how the model parameters were settled:

- Let $\pi_k = \frac{1}{K}$ for $k = 1, 2, \dots, K$.
- Sample $\mu_{k,d} \sim \mathcal{U}(-25, 25)$ and $\sigma_{k,d}^2$ random positive value (The diagonal covariance matrix σ_k^2 with diagonal elements $\sigma_{k,d}^2, d = 1, 2, \dots, D$ was generated using `make_spd_matrix(D) * np.eye(D)` (in Python), where the function `make_spd_matrix()` is from the “scikit-learn” module and `np.eye()` is a function in “numpy”). for $d = 1, 2, \dots, D$ and $k = 1, 2, \dots, K$. $\mathcal{U}(a, b)$ is the uniform distribution.
- Sample $\beta_{k,d} \sim \mathcal{U}(-10, 10)$ and $\log \sigma_{u,k}^2 \sim \mathcal{U}(-0.2, 0.2)$ for $d = 0, 1, 2, \dots, D$ and $k = 1, 2, \dots, K$.
- Let $f_x(\mathbf{z}; \boldsymbol{\theta}) = \mathbf{W}_2 \text{RELU}(\mathbf{W}_1 \text{RELU}(\mathbf{W}_0 \mathbf{z}))$, where $\mathbf{W}_0 \in \mathbb{R}^{H_0 \times D}$, $\mathbf{W}_1 \in \mathbb{R}^{H_1 \times H_0}$, $\mathbf{W}_2 \in \mathbb{R}^{P \times H_1}$ are random matrices ($\mathbf{W}_0, \mathbf{W}_1, \mathbf{W}_2, \mathbf{G}_0, \mathbf{G}_1$ and $\mathbf{G}_2$ were generated using the function `make_low_rank_matrix()` from module “scikit-learn”). H_0 and H_1 are the numbers of units in the two hidden layers of X-decoder.
- Let $f_y(\mathbf{u}; \boldsymbol{\phi}) = \mathbf{G}_2 \text{RELU}(\mathbf{G}_1 \text{RELU}(\mathbf{G}_0 \mathbf{u}))$, where $\mathbf{G}_0 \in \mathbb{R}^{Q_0 \times L}$, $\mathbf{G}_1 \in \mathbb{R}^{Q_1 \times Q_0}$, $\mathbf{G}_2 \in \mathbb{R}^{L \times Q_1}$ are random matrices. Q_0 and Q_1 are the numbers of units in the two hidden layers of Y-decoder.

In our experiment, we used $H_0 = 2^9$, $H_1 = 2^8$, $Q_0 = 2^6$ and $Q_1 = 2^5$. The procedures to generate the datasets of Scenarios 11, 01, 10 and 00 are summarized in Figure A2.

**Figure A2.** Data generating procedures for the synthetic datasets in Scenarios 11, 01, 10 and 00.

Appendix D. Network Architectures

Herein, we supplement the network architectures used for the Synthetic data and the CelebA data in Figures A3 and A4. “layer” is a module from “keras”, which is an API of TensorFlow. While function *layers.Dense()* defines a fully connected layer, *layers.Conv2D()* and *layers.Conv2DTranspose()* indicate the 2D convolutional layer and the 2D transposed convolutional layer respectively, which are the most basic layers used in computer vision. *layers.LeakyReLU()* defines an activation layer. “LeakyReLU” activation function improves the standard “ReLU” by preventing neurons from dying out [42]. In the X-decoder for CelebA, the activation function “sigmoid” was used to map the output value into $[0, 1]$, as we had scaled the original image data to $[0, 1]$. In the Y-decoder, the activation of the last layer was also set as “sigmoid”, as the outcome of CelebA data was binary variable.

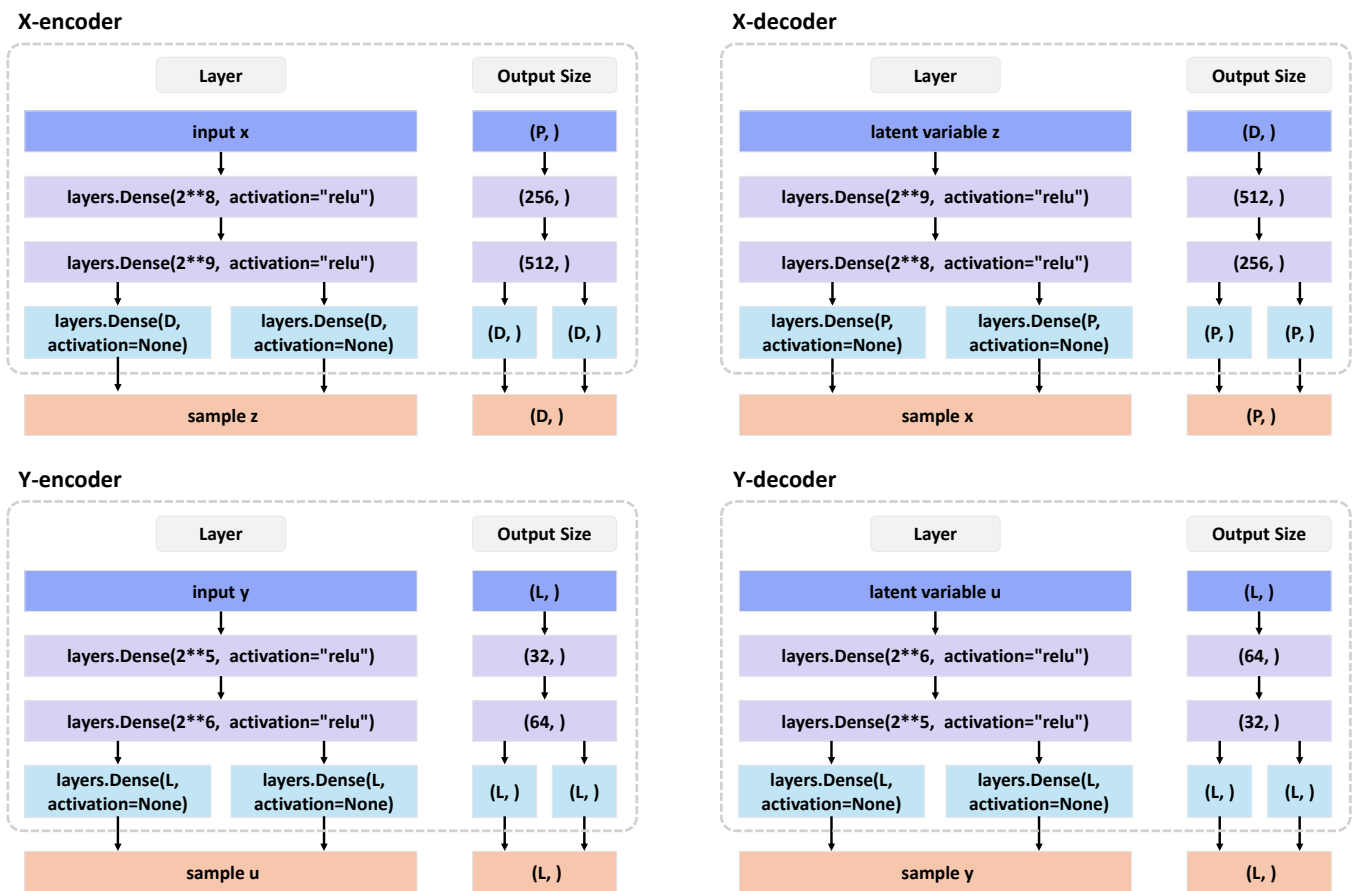


Figure A3. Architectures of encoder and decoder networks used in the experiments for synthetic data. The network layers are represented by their corresponding Python code.

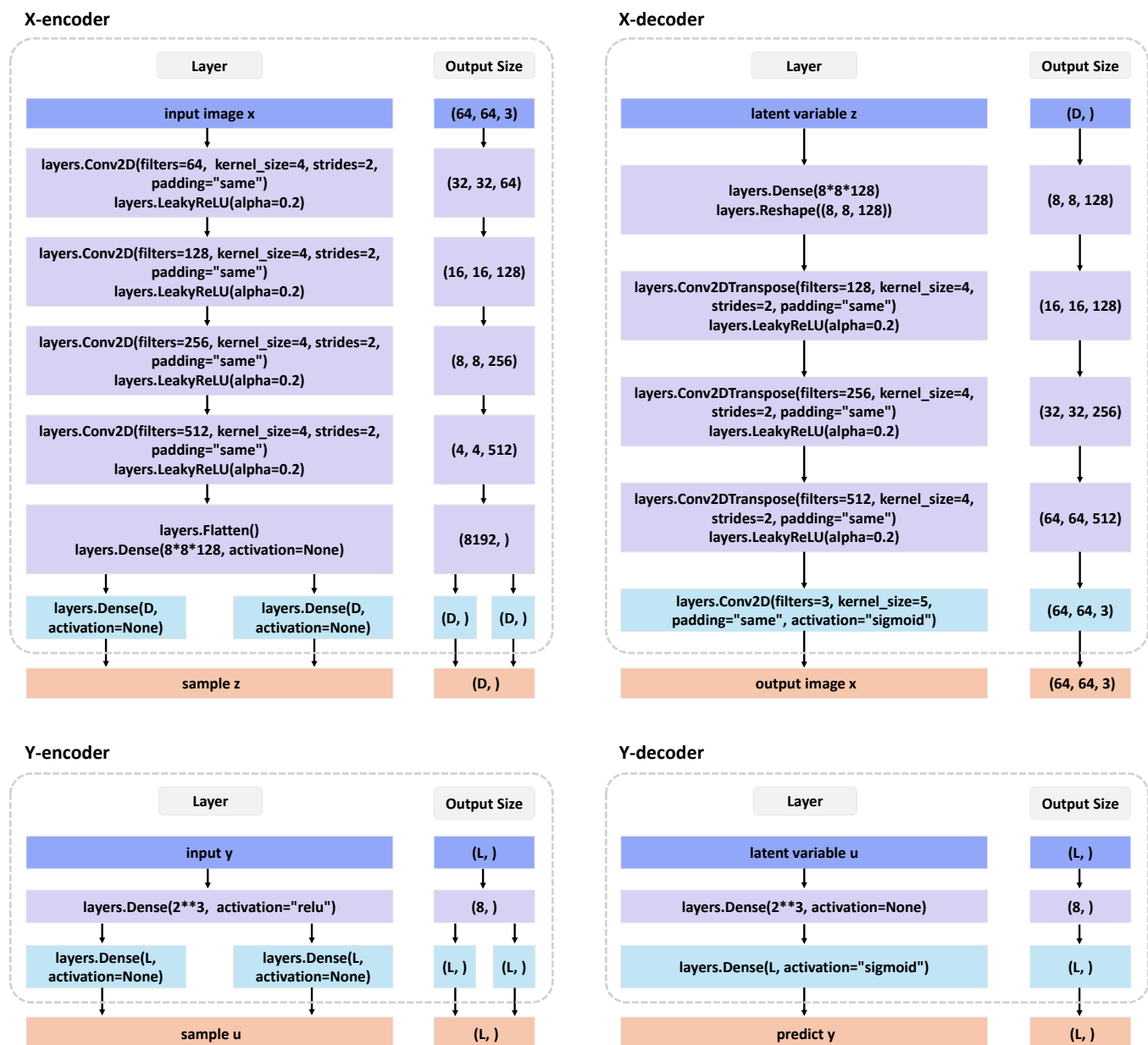


Figure A4. Architectures of encoder and decoder networks used in the experiments for CelebA data. The network layers are represented by their corresponding Python code.

Appendix E. Validation Results

The 5-fold cross-validation was conducted to decide the number of epochs required when implementing the SGVB algorithms for the deep learning methods VaDA, VaDE and VAE (Figures A5 and A6). For the four synthetic scenarios, we used 500 epochs in the algorithms, where the averaged validation loss was at a relatively stable state in each case. And for the CelebA longitudinal data, the setting of 100 epochs was appropriate.

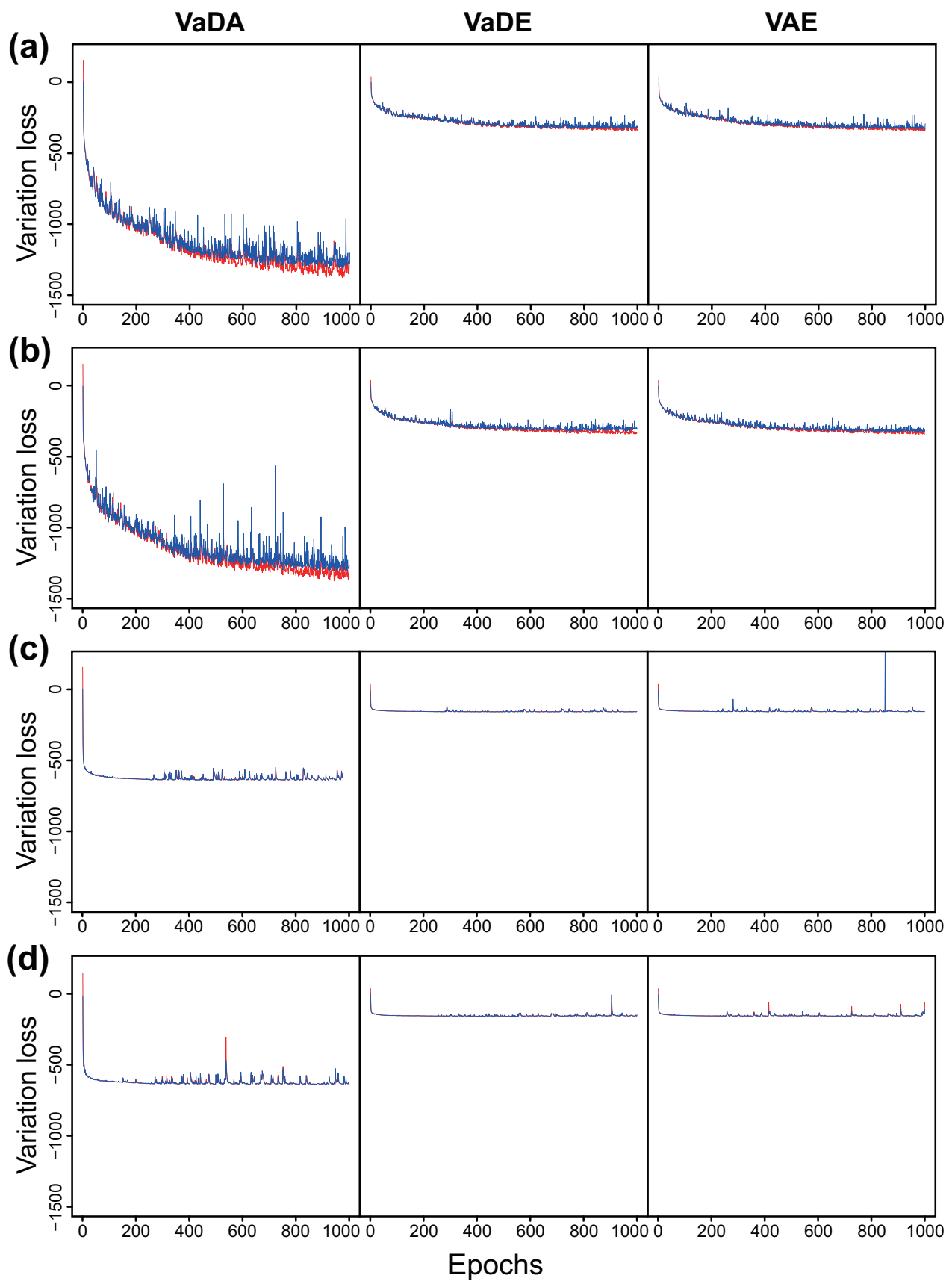


Figure A5. Trends of training loss (red line) and validation loss (blue line) in the synthetic scenarios 11 (a), 01 (b), 10 (c), and 00 (d).

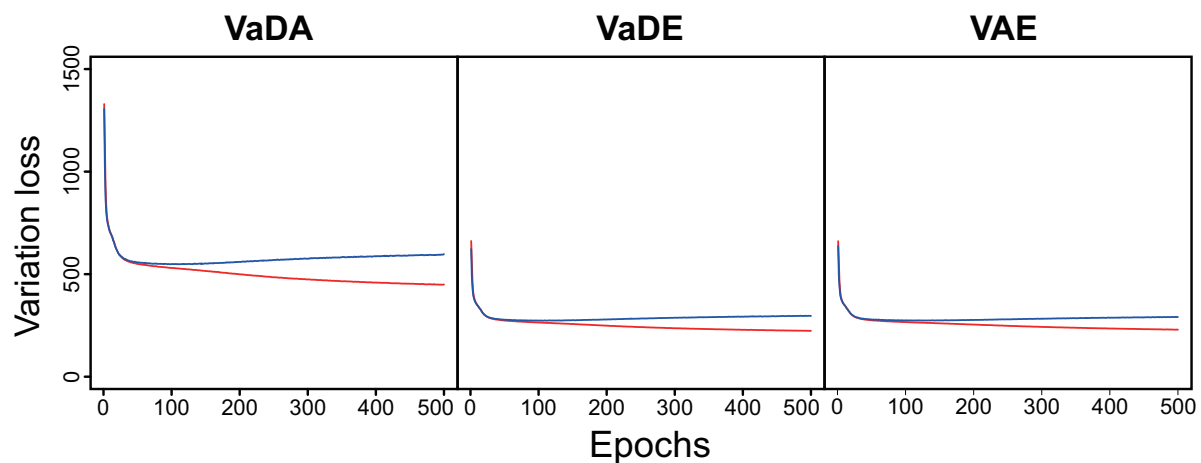


Figure A6. Trends of training loss (red line) and validation loss (blue line) for the CelebA longitudinal dataset.

Appendix F. Influence of MC Sample Size

In the synthetic study and the real-world experiment, we have set the MC sample size for SGVB estimator as $M = 1$ following the convention in papers of VAE [4], VaDE [26] and VaDeSC [27]. Here, a simple sensitivity analysis is conducted to test the influence of MC sample size on the performance of VaDA. Tables A2 and A3 show the clustering and prediction results (on testing datasets) under $M \in \{1, 5, 10\}$ in Scenario 11 based on ten independent runs. As can be seen, the performance of VaDA is not very sensitive to the change of M . There is no apparent performance enhancement by increasing the MC sample size, which could be a result of sufficiently large batch size as mentioned in [4].

Table A2. Performance of VaDA evaluated by adjusted Rand index for clustering results with complete data (ARI-YX) and incomplete data (ARI-X) and by mean square error for prediction accuracy (MSE) under different MC sample size M with varied settings of latent dimensions D when the number of clusters $K = 3$.

Metric	Samples	$D = 2$	$D = 5$	$D = 10$	$D = 20$	$D = 100$
ARI-YX	$M = 1$	0.963 ± 0.110	1.000 ± 0.000	0.956 ± 0.132	1.000 ± 0.000	0.873 ± 0.195
	$M = 5$	1.000 ± 0.000	0.958 ± 0.125	0.956 ± 0.133	0.912 ± 0.175	0.959 ± 0.122
	$M = 10$	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.871 ± 0.198
ARI-X	$M = 1$	0.963 ± 0.110	1.000 ± 0.000	0.956 ± 0.132	1.000 ± 0.000	0.873 ± 0.195
	$M = 5$	1.000 ± 0.000	0.958 ± 0.125	0.956 ± 0.133	0.912 ± 0.175	0.959 ± 0.122
	$M = 10$	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.871 ± 0.198
MSE	$M = 1$	0.281 ± 0.102	0.101 ± 0.029	0.067 ± 0.009	0.057 ± 0.005	0.058 ± 0.006
	$M = 5$	0.292 ± 0.106	0.099 ± 0.029	0.063 ± 0.008	0.059 ± 0.006	0.058 ± 0.007
	$M = 10$	0.323 ± 0.154	0.099 ± 0.025	0.063 ± 0.007	0.056 ± 0.004	0.058 ± 0.003

Figure A7 exhibits the training loss and validation loss of VaDA at a single run with different MC sample size M under the setting of $K = 3$ and $D = 10$.

Table A4 compares the overall memory usage and time consumption between the algorithms of VAE, VaDE and VaDA under different experimental settings in Scenario 11 over ten independent runs. The computational environment was established on the LINUX64 operating system, utilizing an Intel64 CPU architecture. All the algorithms were implemented using dual cores. The memory usage and the time consumption have an apparently increasing tendency as the MC sample size gets larger. Meanwhile, VaDA has approximately doubled memory usage and time consumption compared with VaDE.

This is a result of modeling the data of outcomes, as VaDA also conducts the task of longitudinal prediction.

Table A3. Performance of VaDA evaluated by adjusted Rand index for clustering results with complete data (ARI-YX) and incomplete data (ARI-X) and by mean square error for prediction accuracy (MSE) under different MC sample size M with varied settings of the number of clusters K when latent dimensions $D = 10$.

Metric	Samples	$K = 2$	$K = 3$	$K = 4$
ARI-YX	$M = 1$	0.574 ± 0.025	0.956 ± 0.132	1.000 ± 0.000
	$M = 5$	0.573 ± 0.019	0.956 ± 0.133	1.000 ± 0.000
	$M = 10$	0.574 ± 0.020	1.000 ± 0.000	1.000 ± 0.000
ARI-X	$M = 1$	0.574 ± 0.025	0.956 ± 0.132	1.000 ± 0.000
	$M = 5$	0.573 ± 0.019	0.956 ± 0.133	1.000 ± 0.000
	$M = 10$	0.574 ± 0.020	1.000 ± 0.000	1.000 ± 0.000
MSE	$M = 1$	0.062 ± 0.006	0.067 ± 0.009	0.065 ± 0.002
	$M = 5$	0.064 ± 0.005	0.063 ± 0.008	0.063 ± 0.006
	$M = 10$	0.068 ± 0.006	0.063 ± 0.007	0.062 ± 0.004

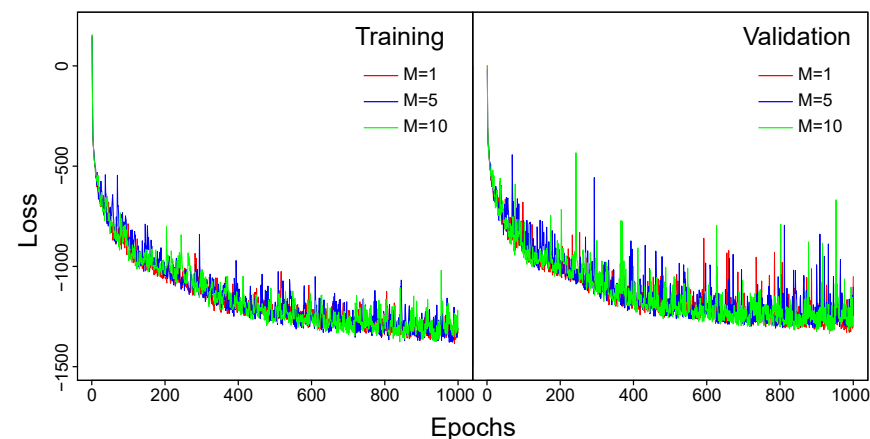


Figure A7. Comparison of the training loss and validation loss of VaDA with different MC sample size M under the setting of $K = 3$ and $D = 10$.

Table A4. Comparison of overall memory usage and time consumption between the algorithms of VAE, VaDE and VaDA under different experimental settings over ten independent runs.

Method	K	D	$M = 1$			$M = 5$			$M = 10$		
			CPU	Memory (GiB)	Time (m)	CPU	Memory (GiB)	Time (m)	CPU	Memory (GiB)	Time (m)
VAE	–	2	2	4.56	64.27	2	6.31	109.83	2	8.70	177.33
	–	5	2	4.54	64.33	2	6.56	110.58	2	8.60	171.15
	–	10	2	4.56	63.85	2	6.52	110.23	2	8.71	169.33
	–	20	2	4.39	66.00	2	6.38	107.62	2	8.79	168.02
	–	100	2	4.48	64.98	2	6.53	120.65	2	8.71	205.22
VaDE	3	2	2	6.71	79.03	2	8.89	116.18	2	10.85	180.60
	3	5	2	6.96	75.13	2	8.80	122.15	2	10.93	184.60
	3	10	2	6.66	78.03	2	8.75	121.20	2	10.96	188.30
	3	20	2	6.96	78.82	2	8.73	126.58	2	10.86	189.65
	3	100	2	6.75	81.17	2	8.88	134.02	2	11.08	221.92
	2	10	2	6.43	69.42	2	8.48	113.25	2	10.44	193.63
	4	10	2	7.47	78.10	2	9.16	120.43	2	11.31	198.57

Table A4. Cont.

Method	K	D	M = 1			M = 5			M = 10		
			CPU	Memory (GiB)	Time (m)	CPU	Memory (GiB)	Time (m)	CPU	Memory (GiB)	Time (m)
VaDA	3	2	2	13.07	136.13	2	17.80	230.40	2	23.14	333.55
	3	5	2	13.26	145.47	2	17.98	227.88	2	23.31	320.53
	3	10	2	13.62	141.22	2	17.91	231.87	2	23.15	335.52
	3	20	2	13.18	145.25	2	17.70	235.70	2	23.07	342.47
	3	100	2	13.43	152.73	2	17.79	235.28	2	23.28	364.47
	2	10	2	12.16	132.18	2	16.44	219.83	2	22.08	319.65
	4	10	2	14.61	151.97	2	19.14	241.77	2	24.56	347.42

Appendix G. Instructions of PCA

When performing the method VaDE+MM, VAE+GMM+MM and GMM+MM, we encountered frequently the problem in marginal model that “the design matrix is rank deficiency and the GEE cannot proceed”. This is a common problem in linear regression analysis with high-dimensional data, where dimension reduction or feature transformation is required. As a simple and efficient treatment, we used the PCA to make a projection of the feature (representation) data in each obtained cluster into orthogonal principal components. In a cluster, there were generally two situations to be treated: the number of observations in that cluster was larger than the dimensions of feature data; the number of observations in that cluster equaled to or was smaller than the dimensions. In the first situation, the traditional PCA was applied and the original dimensions of data was kept. In the second situation, the traditional PCA could not proceed, we had to use the sparse PCA [31], where sparse loadings were obtained by imposing the 1-norm constraint on the regression coefficients from viewing the PCA as a regression-type optimization problem. The number of principal components was set as $N_k - 1$, where N_k was the number of observations in cluster k . Throughout the experimental study, the marginal models were fitted using the R package “geepack”. The remedies by sparse PCA were implemented using the R package “elasticnet”.

References

- Fahrmeir, L.; Tutz, G. *Multivariate Statistical Modelling Based on Generalized Linear Models*, 2nd ed.; Springer Series in Statistics; Springer: New York, NY, USA, 2001. [\[CrossRef\]](#)
- Liang, K.Y.; Zeger, S.L. Longitudinal data analysis using generalized linear models. *Biometrika* **1986**, *73*, 13–22. [\[CrossRef\]](#)
- Zhu, R.; Guo, D.; Qi, D.; Chu, Z.; Yu, X.; Li, S. A survey of trustworthy representation learning across domains. *ACM Trans. Knowl. Discov. Data* **2024**, *18*, 173. [\[CrossRef\]](#)
- Kingma, D.P.; Welling, M. Auto-encoding variational bayes. In Proceedings of the 2nd International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
- Hastie, T.; Tibshirani, R.; Wainwright, M.J. *Statistical Learning with Sparsity: The Lasso and Generalizations*; CRC Press: Boca Raton, FL, USA, 2015.
- Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep learning face attributes in the wild. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
- Clogg, C.C. Some models for the analysis of association in multiway cross-classifications having ordered categories. *J. Am. Stat. Assoc.* **1982**, *77*, 803–815. [\[CrossRef\]](#)
- Qu George, Y.; Williams, G.W.; Beck, G.J.; Goormastic, M. A generalized model of logistic regression for clustered data. *Commun. Stat. Theory Methods* **1987**, *16*, 3447–3476. [\[CrossRef\]](#)
- Longford, N.T. *Random Coefficient Models*; Oxford University Press: Oxford, UK, 1994. [\[CrossRef\]](#)
- Lin, X.; Breslow, N.E. Bias correction in generalized linear mixed models with multiple components of dispersion. *J. Am. Stat. Assoc.* **1996**, *91*, 1007–1016. [\[CrossRef\]](#)

11. Lindstrom, M.J.; Bates, D.M. Nonlinear mixed effects models for repeated measures data. *Biometrics* **1990**, *46*, 673–687. [[CrossRef](#)] [[PubMed](#)]
12. Davidian, M.; Giltinan, D.M. Nonlinear models for repeated measurement data: An overview and update. *J. Agric. Biol. Environ. Stat.* **2003**, *8*, 387–419. [[CrossRef](#)]
13. Sohn, K.; Yan, X.; Lee, H. Learning structured output representation using deep conditional generative models. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, Canada, 7–12 December 2015; Volume 28.
14. Casale, F.P.; Dalca, A.V.; Saglietti, L.; Listgarten, J.; Fusi, N. Gaussian process prior variational autoencoders. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, Canada, 3–8 December 2018; Volume 31.
15. Wilson, A.G.; Adams, R.P. Gaussian process kernels for pattern discovery and extrapolation. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; Volume 28.
16. Ramchandran, S.; Tikhonov, G.; Kujanpää, K.; Koskinen, M.; Lähdesmäki, H. Longitudinal variational autoencoder. In Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, Virtual Conference, 13–15 April 2021; Volume 130.
17. Öğretir, M.; Ramchandran, S.; Papatheodorou, D.; Lähdesmäki, H. A variational autoencoder for heterogeneous temporal and longitudinal data. In Proceedings of the 21st IEEE International Conference on Machine Learning and Applications, Nassau, Bahamas, 12–14 December 2022.
18. Ramchandran, S.; Tikhonov, G.; Lönnroth, O.; Tiikkainen, P.; Lähdesmäki, H. Learning conditional variational autoencoders with missing covariates. *Pattern Recognit.* **2024**, *147*, 110113. [[CrossRef](#)]
19. Gootjes-Dreesbach, L.; Sood, M.; Sahay, A.; Hofmann-Apitius, M.; Fröhlich, H. Variational autoencoder modular Bayesian networks for simulation of heterogeneous clinical study data. *Front. Big Data* **2020**, *3*, 16. [[CrossRef](#)] [[PubMed](#)]
20. Krishnan, R.G.; Shalit, U.; Sontag, D. Deep Kalman Filters. In Proceedings of the NIPS 2015 Deep Learning Workshop, Montréal, QC, Canada, 11–12 December 2015.
21. Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A.C.; Bengio, Y. A recurrent latent variable model for sequential data. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 7–12 December 2015; Volume 28.
22. Johnson, M.J.; Duvenaud, D.; Wiltchko, A.B.; Datta, S.R.; Adams, R.P. Composing graphical models with neural networks for structured representations and fast inference. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 7–12 December 2016; Volume 29.
23. Mandel, F.; Ghosh, R.P.; Barnett, I. Neural networks for clustered and longitudinal data using mixed effects models. *Biometrics* **2023**, *79*, 711–721. [[CrossRef](#)] [[PubMed](#)]
24. Rosen, O.; Jiang, W.; Tanner, M.A. Mixture of marginal models. *Biometrika* **2000**, *87*, 391–404. [[CrossRef](#)]
25. Ito, T.; Sugawara, S. Grouped generalized estimating equations for longitudinal data analysis. *Biometrics* **2023**, *79*, 1868–1879. [[CrossRef](#)] [[PubMed](#)]
26. Jiang, Z.; Zheng, Y.; Tan, H.; Tang, B.; Zhou, H. Variational deep embedding: An unsupervised and generative approach to clustering. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017.
27. Manduchi, L.; Marcinkevičs, R.; Vogt, J.E. A deep variational approach to clustering survival data. In Proceedings of the AI for Public Health Workshop at ICLR 2021, Virtual Conference, 8–13 May 2021.
28. Qiu, J.; Hu, Y.; Li, L.; Erzurumluoglu, A.M.; Braenne, I.; Whitehurst, C.; Schmitz, J.; Arora, J.; Bartholdy, B.A.; Gandhi, S.; et al. Deep representation learning for clustering longitudinal survival data from electronic health records. *Nat. Commun.* **2025**, *16*, 2534. [[CrossRef](#)] [[PubMed](#)]
29. Højsgaard, S.; Halekoh, U.; Yan, J. The R package geepack for generalized estimating equations. *J. Stat. Soft.* **2005**, *15*, 1–11. [[CrossRef](#)]
30. Venables, W.N.; Ripley, B.D. *Modern Applied Statistics with S*; Springer: Berlin/Heidelberg, Germany, 2002.
31. Zou, H.; Hastie, T.; Tibshirani, R. Sparse principal component analysis. *J. Comput. Graph. Stat.* **2006**, *15*, 265–286. [[CrossRef](#)]
32. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016.
33. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
34. Rodríguez, C.E.; Walker, S.G. Label switching in Bayesian mixture models: Deterministic relabeling strategies. *J. Comput. Graphical Stat.* **2014**, *23*, 25–45. [[CrossRef](#)]
35. Hou, X.; Sun, K.; Shen, L.; Qiu, G. Feature perceptual loss for variational autoencoder. *arXiv* **2024**, arXiv:1610.00291v2.
36. Bartlett, P.L.; Jordan, M.I.; McAuliffe, J.D. Convexity, classification, and risk bounds. *J. Am. Stat. Assoc.* **2006**, *101*, 138–156. [[CrossRef](#)]
37. François, C. *Deep Learning with Python*, 2nd ed.; Manning Publications: Shelter Island, NY, USA, 2021.

38. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; Volume 48.
39. Irina, H.; Loic, M.; Arka, P.; Christopher, B.; Xavier, G.; Matthew, B.; Shakir, M.; Alexander, L. β -VAE: Learning basic visual concepts with a constrained variational framework. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
40. Kivva, B.; Rajendran, G.; Ravikumar, P.; Aragam, B. Identifiability of deep generative models without auxiliary information. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; Volume 35.
41. Ibrahim, J.G.; Molenberghs, G. Missing data methods in longitudinal studies: A review. *Test* **2009**, *18*, 1–43. [[CrossRef](#)] [[PubMed](#)]
42. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.