Deep Latent Variable Model for Learning Longitudinal Multi-view Data

Lin Qiu [1] [2]   Vernon M. Chinchilli [1] [2]   Lin Lin [2]

Abstract

In many scientific problems such as video surveillance, modern genomics, and finance, data often are collected from diverse domains across time that exhibit time-dependent heterogeneous properties. It is important to not only integrate data from multiple sources (called multi-view data), but also to incorporate time dependency for deep understanding of the underlying system. Latent factor models are popular tools for exploring multi-view data. However, it is frequently observed that these models do not perform well for complex systems and they are not applicable to longitudinal data. Therefore, we propose a generative model based on variational autoencoder and recurrent neural network to infer the latent dynamic factors for multi-view longitudinal data. This approach allows us to identify the disentangled latent embeddings across multiple modalities while accounting for the time factor. We invoke our proposed model for analyzing three datasets on which we demonstrate the effectiveness and the interpretability of the model.

1. Introduction

Multi-view learning is an emerging problem in machine learning research, as multi-view data become increasingly common in many real world applications. Examples include multi-omics data where different biological layers such as genomics, epigenomics, transcriptomics and proteomics can be obtained from the same set of objects. In finance, the performance of each company may be better described in terms of different asset classes, such as stocks and bonds. In computer vision, a scene is typically represented by a series of audio and image frames. In those situations, the same set of objects typically have different features (views) collected from different measuring methods or domains,

where any particular single-view data is potentially inadequate to comprehensively describe the information of all the objects. Hence, one major goal of multi-view learning is to determine a lower-dimensional latent representation that can explain multiple views of the data and capture the shared variations among all views.

Many popular multi-view learning methods have been developed based on group factor analysis (Sridharan & Kakade, 2008; Klami et al., 2015; Zhao et al., 2016; Leppaaho et al., 2017), which generates a common linear mapping between the latent and observed groups of variables (multiple views). In order to further extract interpretable information, most of the methods exploit the idea of using sparse linear factor models. In particular, the resulting latent factor is restricted to contribute to variation in only a subset of the observed features. For example, sparse factor loadings in gene expression data analysis can be interpreted as non-disjoint clusters of co-regulated genes (Pournara & Wernisch, 2007; Lucas et al., 2010; Gao et al., 2013). In multi-omics studies, the sparse group factor analysis model helps to infer a set of hidden factors that capture both the biological and technical sources of variability. This information is important for downstream analysis, such as identification of sample subgroups, data imputation, and outlier sample detection (Argelaguet et al., 2018).

Despite the wide applications of group factor analysis, existing modeling approaches are severely challenged by complex data encountered in many research areas. For example, in various disease studies, the ability to integrate different data types (e.g., genomic, transcriptomic, epigenomic, and proteomic data) obtained from both healthy and disease individuals across different time point is crucial for the understanding of disease progression which can further contribute towards translational research for personalized medicine. Existing group factor analysis methods can not handle these multi-modal time-dependent complex structures efficiently (Ainsworth et al., 2018).

Latent variable models, on the other hand, are widely used in modeling high-dimensional longitudinal data. The basic idea is to use the low-dimensional latent variables to automatically induce dependency among the observed data space. Among them, the linear dynamical system (Rabiner & Juang, 1986) and hidden Markov models (Kalman, 1963)

are widely studied. However, those methods do not scale well with complex non-linear dynamics. Recently, recurrent neural networks(RNNs) (Martens & Sutskever, 2011; Hermans & Schrauwen, 2013; Pascanu et al., 2013; Graves, 2013) have displayed good performance in modeling sequence data, where the latent random variables in the RNN function serve as "memory" of the past sequence. RNN can be further extended to integrate the dependencies between the latent random variables at neighboring timesteps, called variational recurrent neural network (VRNN) (Chung et al., 2015), which can handle complex nonlinear highly structured sequential data, in the context of variational autoencoder (Kingma & Welling, 2014).

Our motivation lies in the study of high-dimensional longitudinal multi-view data. We seek to infer trajectories of latent variables that provide insight into the latent, lower-dimensional structure derived from the dynamics of the observed data space. Motivated by the success of VRNN for modeling temporal sequence data, we propose a new modeling strategy that integrates VRNN into sparse group factor analysis, in which the resulting model serves as a nonlinear factor model for multi-view data observed across time. Further, the model interpretability is achieved through the use of sparse priors on the latent-to-observed mappings. Specifically, each view has a generator neural network and only a small number of correlated views will be affected by each latent dimension. We label this model as the deep latent variable model for longitudinal group factor analysis (DLGFA).

2. Background

Generative Model: In generative models as shown in Figure 1, the class of variatonal autoencoders (VAEs) are popular for efficient approximate inference and learning (Kingma & Welling, 2014). VAE approximates intractable posterior distributions over latent representations that are parameterized by a deep neural network, which maps observations to a distribution over latent variables.

For non-sequential data, VAE has become one of the most popular approaches for efficiently recovering complex multimodal distributions. Recently, VAE has been extended to dynamic systems (Archer et al., 2015). Briefly, VAE provides a mapping from the observations to a distribution on their latent representation. The resulting simpler latent subspace can be used to describe the underlying complex system. Mathematically, let $\mathbf{x} \in \mathcal{R}^d$ denote a d-dimensional observation and $\mathbf{z} \in \mathcal{R}^K$ denote a vector of *latent random variables* of fixed dimension K with $K < d$. The generative process of VAE can be represented as:

$$\mathbf{z} \sim \mathcal{N}(0, I), \mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, D). \tag{1}$$

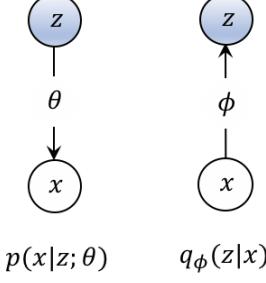where I is the identify matrix, D is a $d \times d$ diagonal matrix

Figure 1. Generative Model. Left: The conditional probability $p(x|z; \theta)$ parameterized by a non linear deep neural network through the latent variable z. Right: The inference network $q_\phi(z|x)$.

whose diagonals are the marginal variances of each component of $\mathbf{x}$, $\boldsymbol{\mu}_{\mathbf{x}}$ is the mean of the Gaussian likelihood and is produced by a neural network with parameters θ taking $\mathbf{z}$ as an input. The joint distribution is then defined as:

$$p(\mathbf{x}, \mathbf{z}; \theta) = p(\mathbf{x} \mid \mathbf{z}; \theta)p(\mathbf{z}). \tag{2}$$

Learning Inference: One unique feature of the VAE is that it allows the conditional $p(\mathbf{x} \mid \mathbf{z})$ being a potentially highly nonlinear mapping from $\mathbf{z}$ to $\mathbf{x}$.

The likelihood is then parameterized with a generative network (called decoder). VAE uses $q(\mathbf{z}|\mathbf{x})$ with an inference network (called encoder) to approximate the posterior distribution of $\mathbf{z}$. For example, $q(\mathbf{z}|\mathbf{x})$ can be a Gaussian $\mathcal{N}(\mu, \sigma^2 I)$, where both μ and σ^2 are parameterized by a neural network: $[\mu, \log \sigma^2] = f_\phi(\mathbf{x})$, where f_ϕ is a neural network with parameters ϕ. The parameters for both generative and inference networks are learned through variational inference, Jensen's inequality yields the evidence lower bound (ELBO) on the marginal likelihood of the data:

$$\log p_\theta(\mathbf{x}) \geq \underbrace{\mathbb{E}_{q(\mathbf{z};\phi)}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - \mathrm{KL}(q(\mathbf{z};\phi) \mid\mid p(\mathbf{z}))}_{\mathcal{L}(x;\theta,\phi)}. \tag{3}$$

where $\mathrm{KL}(Q\|P)$ is Kullback-Leibler divergence between two distributions Q and P. $q(\mathbf{z}; \phi)$ is a tractable "variational" distribution meant to approximate the intractable posterior distribution $p(\mathbf{z} \mid \mathbf{x})$; it is controlled by some parameters ϕ. We want to choose ϕ that makes the bound in equation 3 as tight as possible, $\phi^* \triangleq \arg\max_\phi \mathcal{L}(x; \theta, \phi)$.

One can train a feedforward *inference network* to find good variational parameters $\phi(x)$ for a given x, where $\phi(x)$ is the output of a neural network with parameters ϕ that are trained to maximize $\mathcal{L}(x; \theta, \phi(x))$ (Kingma & Welling, 2014).

3. Related Work

For complex structured data modeling, previous work including deep conditional generative models (CVAE) for output representation learning and structured prediction (Sohn et al., 2015). CVAE is a conditional directed graphical model whose input observations modulate the prior on Gaussian latent variables that generate the outputs.

DP-GP-LVM is a non-parametric Bayesian latent variable model aims to learn the dependency structures of multiple groups data by the Gaussian process priors for the generative mappings and Dirichlet process priors (Lawrence et al., 2019). Compared to these methods, our proposed model can achieve interpretable results for the time dynamic dependency structure. Output interpretable VAE (oi-VAE) is designed for non-temporal grouped (multi-view) data with a structured VAE comprised of group-specific generators (Ainsworth et al., 2018).

The latent variable **z** is shared across all groups and assumed iid for each data point. We borrowed the idea from oi-VAE to introduce the column-wise sparsity inducing prior on the latent-to-group matrices for interpretable results.

4. Our Contribution

Learning from longitudinal multi-view data: To our best knowledge, DLGFA is the first nonlinear interpretable modeling framework to study the complex longitudinal multi-view data. The ability of DLGFA to perform sensible disentanglement is through the introduction of a group and time-specific transformation **W** (introduced in Section 5) which built on a sparsity inducing prior between the time-specific latent representation **z** and the group and time-specific neural generator. By penalizing interactions between the components of **z** and each of the groups, DLGFA effectively forces the model to arrive at a representation that minimizes correlation across the components of **z**, encouraging each component to capture the dynamic distinct modes of variation.

Learning dynamic dependency among groups: **z** captures the temporal variability shared among all groups. By associating each component of **z** with only a sparse subset of the groups, we are able to identify the dynamic disentangled representations for multi-view data. The relationship between the dimension of **z** and the groups can be used as an exploratory tool to indicate the dependency among different groups. This is an appealing feature of DLGFA because most of the complex systems depend on a temporal component, and such component contributes to the development of variable interactions gradually. The ability to access the dynamic relationship of groups of variables will help us gain insight for downstream analysis of the complex data.

5. Deep Latent Variable Model for Longitudinal Group Factor Analysis

We propose to develop a new VRNN framework for longitudinal multi-view data though the dynamic modeling of the latent representations for better interpretability, as in nonlinear group factor analysis.

5.1. Modeling Framework

Prior Given a temporal sequence of vectors $\mathbf{x}_{1:T} = (\mathbf{x}_1, ..., \mathbf{x}_T), \mathbf{x}_t \in \mathcal{R}^d$, the conventional VAEs assumes independent latent variable **z** for each timestep t: $\mathbf{z} \sim \mathcal{N}(0, I)$. To encode temporal variability, we propose to allow the latent variable z_t at timestep t to depend on the state variable $\mathbf{h}_{t-1}$ of an RNN though the following distribution:

$$\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{0,t}, \text{diag}(\boldsymbol{\sigma}_{0,t}^2)), \quad [\boldsymbol{\mu}_{0,t}, \boldsymbol{\sigma}_{0,t}] = \varphi_\tau^{\text{prior}}(\mathbf{h}_{t-1}),$$
$$(4)$$

where both $\boldsymbol{\mu}_{0,t}$ and $\boldsymbol{\sigma}_{0,t}$ are produced by a distinct neural network that approximates the time dependent prior distribution (Chung et al., 2015). More specifically, $[\boldsymbol{\mu}_{0,t}, \boldsymbol{\sigma}_{0,t}^2] = \varphi_\tau^{\text{prior}}(\mathbf{h}_{t-1})$, and $\varphi_\tau^{\text{prior}}(\mathbf{h}_{t-1})$ denote a neural network taking the previous hidden state $\mathbf{h}_{t-1}$ as input.

Encoder Similar to the VAEs, we need to define an approximate posterior $q(\mathbf{z}|\mathbf{x})$. We propose to let z_t capture the shared variability among groups at each timestep by allowing $q(\mathbf{z}|\mathbf{x})$ as a function of both $\mathbf{x}_t = [\mathbf{x}_t^{(1)}, ..., \mathbf{x}_t^{(G)}]$ and $\mathbf{h}_{t-1}$ as:

$$\mathbf{z}_t|\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z},t}, \boldsymbol{D}_{\mathbf{z},t}), \qquad (5)$$
$$[\boldsymbol{\mu}_{\mathbf{z},t}, \text{diag}(\boldsymbol{D}_{\mathbf{z},t})] = \varphi_\tau^{\text{enc}}(\varphi_\tau^{\mathbf{x}}(\mathbf{x}_t), \mathbf{h}_{t-1}). \qquad (6)$$

Decoder Rewrite the sequence data to incorporate the group information as $\mathbf{x}_{1:T} = (\mathbf{x}_1, ..., \mathbf{x}_T), \mathbf{x}_t = [\mathbf{x}_t^{(1)}, ..., \mathbf{x}_t^{(G)}] \in \mathcal{R}^{d \times G}$, where $\mathbf{x}_t^{(g)}$ denote the data from group g at time t. The generation of $\mathbf{x}_t$ will depend on both $\mathbf{z}_t$ and $\mathbf{h}_{t-1}$. In addition, we propose to model different views of data independently while allowing the latent variable $\mathbf{z}_t$ to be shared across G views at timestep t. The corresponding generative distribution will be:

$$\mathbf{x}_t^{(g)} \mid \mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x},t}^{(g)}, \boldsymbol{D}_{\mathbf{x},t}^{(g)}). \qquad (7)$$

where $\mathbf{D}_{\mathbf{x},t}^{(g)}$ is a diagonal matrix. We propose to introduce a sequence of latent matrices $\mathbf{W}_t^{(g)} \in \mathcal{R}^{p \times K}$, for $t = 1 : T, g = 1 : G$. Both parameters $\boldsymbol{\mu}_{\mathbf{x},t}^{(g)}$ and $\mathbf{D}_{\mathbf{x},t}^{(g)}$ will be conditioned on $\mathbf{W}_t^{(g)}, \mathbf{z}_t$ and $\mathbf{h}_{t-1}$ through:

$$[\boldsymbol{\mu}_{\mathbf{x},t}^{(g)}, \text{diag}(\boldsymbol{D}_{\mathbf{x},t}^{(g)})] = \varphi_{\theta_{t,g}}^{\text{dec}}(\mathbf{W}_t^{(g)}\mathbf{z}_t, \mathbf{h}_{t-1}). \qquad (8)$$

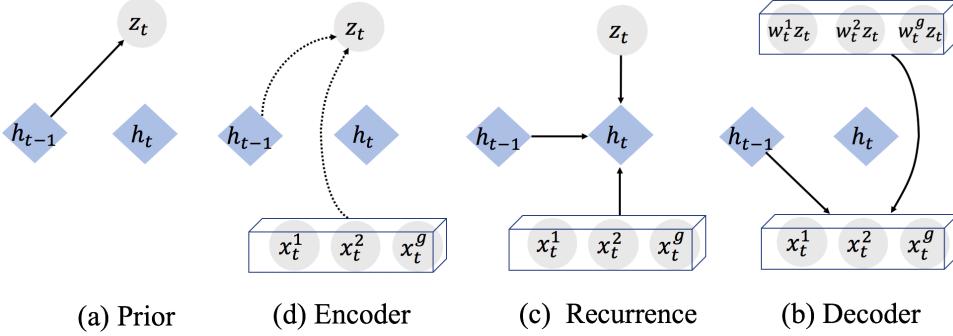(a) Prior   (d) Encoder   (c) Recurrence   (b) Decoder

Figure 2. Graphical illustration of each operation in DLGFA model, light blue color represents the hidden state and grey color represents data (in the box) and latent variable: (a) Computing the conditional prior using Eq. (4); (b) Inference of the approximate posterior using Eq. (5); (c) Updating RNN hidden states using Eq. (9), and (d) Generating function using Eqs. (7) and (8).

where $\varphi_{\theta_{t,g}}^{\text{dec}}$ denotes a neural network with parameters $\theta_{t,g}$, and $\text{diag}(\mathbf{D})$ denotes the diagonal elements of the matrix $\mathbf{D}$.

Recurrence The hidden state $\mathbf{h}_t$ is updating the conditioning on $\mathbf{z}_t$ in a recurrent way: $\mathbf{h}_t = S_\theta(\mathbf{x}_t, \mathbf{z}_t, \mathbf{h}_{t-1})$, where S is the transition function which can be implemented with gated activation functions such as long short-term memory or gated recurrent unit (Cho et al., 2014; Hochreiter & Schmidhuber, 1997). VRNN demonstrates that by including feature extractors in the recurrent equation are important for learning complex data:

$$\mathbf{h}_t = S_\theta\left(\varphi_\tau^{\mathbf{x}}(\mathbf{x}_t), \varphi_\tau^{\mathbf{z}}(\mathbf{z}_t), \mathbf{h}_{t-1}\right). \tag{9}$$

where $\varphi_\tau^{\mathbf{x}}$ and $\varphi_\tau^{\mathbf{z}}$ are two neural networks for feature extraction from $\mathbf{x}_t$ and $\mathbf{z}_t$, respectively. By the above model specifications, the generative distribution can be factorized as:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = \prod_{t=1}^{T}\left[\prod_{g=1}^{G} p(\mathbf{x}_t^{(g)}|\mathbf{z}_{\leq t}, \mathbf{x}_{\leq t}^{(g)})\right]p(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{\leq t}). \tag{10}$$

The model structure is depicted in Figure 2.

Model Interpretability We place a column-wise sparsity prior for $\mathbf{W}_t^{(g)}$ to ensure the interpretability of the model (Kyung et al., 2010):

$$\gamma_{gjt}^2 \sim \text{Gamma}\left(\frac{p+1}{2}, \lambda^2/2\right), \tag{11}$$

$$\mathbf{W}_{t:,j}^{(g)} \sim \mathcal{N}(0, \gamma_{gjt}^2 I). \tag{12}$$

p denotes the number of rows in each $\mathbf{W}_{t:,j}^{(g)}$, the parameter λ will control the model sparsity, larger λ will imply more

column-wise sparsity in $\mathbf{W}_{t:,j}^{(g)}$. Marginalizing over γ_{gjt}^2 induces group sparsity over the columns of $\mathbf{W}_{t:,j}^{(g)}$. Hence, the model automatically tracks the sparse features among groups through time.

5.2. Timestep-wise Learning

The traditional VAEs are learned according to optimization of the ELBO by stochastic gradient methods. We are more interested in the sparsity of learned $\mathbf{W}_{t:,j}^{(g)}$ for model interpretability. The sparsity inducing prior on $\mathbf{W}_{t:,j}^{(g)}$ is marginally equivalent to the convex group lasso penalty. Hence, we propose to adapt the idea of collapsed variational inference (Ainsworth et al., 2018) to obtain true sparsity of the columns $\mathbf{W}_{t:,j}^{(g)}$ and apply the timestep-wise variational lower bound.

Theorem 1 *Let* $\mathcal{W} = \left(\mathbf{W}_{1:T}^{(1)}, \cdots, \mathbf{W}_{1:T}^{(G)}\right)$, $\gamma^2 = \left(\gamma_{1:G,1:K,1:T}^2\right)$, $\mathbf{x} = \mathbf{x}_{1:T}$, *and* $\mathbf{z} = \mathbf{z}_{1:T}$. *Thus, we can compute* $\log p(\mathbf{x})$ *by marginalizing out all* γ_{gjt}^2's:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\left[\sum_{t=1}^{T} -\text{KL}(q_\phi(\mathbf{z}_t \mid \mathbf{x}_{\leq t}, \mathbf{z}_{<t})\|\right.$$

$$\left. p(\mathbf{z}_t \mid \mathbf{x}_{<t}, \mathbf{z}_{<t})) + \log p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{x}_{<t})\right]$$

$$+ \log p(\theta_t) - \lambda \sum_{t=1}^{T}\sum_{g,j}\|\mathbf{W}_{t:,j}^{(g)}\|_2$$

Where $\gamma_{gjt}^2 \sim \text{Gamma}\left(\frac{p+1}{2}, \lambda^2/2\right)$, $\mathbf{W}_{t:,j}^{(g)} \sim \mathcal{N}(0, \gamma_{gjt}^2 I)$, *and* ϕ, θ *are neural network parameters.*

Proof: See Supplementary A.

Algorithm 1 Collapsed VI for DLGFA

Input: data $\mathbf{x}^{(i)}$, sparsity parameter λ
Let $\tilde{\mathcal{L}}_t$ be $\mathcal{L}(\phi_t, \theta_t, \mathcal{W}_t)$ but without $-\lambda \sum_{g,j} ||\mathbf{W}_{t:,j}^{(g)}||_2$.
repeat
    For each time point t
    Calculate $\nabla_{\phi_t} \tilde{\mathcal{L}}_t$, $\nabla_{\theta_t} \tilde{\mathcal{L}}_t$, and $\nabla_{\mathcal{W}_t} \tilde{\mathcal{L}}_t$.
    Update ϕ_t and θ_t with Adam optimizer.
    Let $\mathcal{W}_t t + 1 = \mathcal{W}_t t - \eta \nabla_{\mathcal{W}} \tilde{\mathcal{L}}_t$.
    for all groups g, $j = 1$ **to** K **do**
        Set $\mathbf{W}_{t:,j}^{(g)} \leftarrow \frac{\mathbf{W}_{t:,j}^{(g)}}{||\mathbf{W}_{t:,j}^{(g)}||_2} \left(||\mathbf{W}_{t:,j}^{(g)}||_2 - \eta\lambda \right)_+$
    end for
until convergence in both $\sum_{t=1}^{T} \hat{\mathcal{L}}_t$ and $-\lambda \sum_{t=1}^{T} \sum_{g,j} ||\mathbf{W}_{t:,j}^{(g)}||_2$

5.3. Optimization

The goal is to maximize this collapsed ELBO over ϕ, θ, and $\mathbf{W}$. The columns of the latent-to-group matrices $\mathbf{W}_{t:,j}^{(g)}$ appear in a $\mathcal{L}_2$-norm penalty in the new collapsed ELBO which is a group lasso penalty on the columns of $\mathbf{W}_{t:,j}^{(g)}$ and encourages the entire vector to be set to zero. We propose the use of efficient proximal gradient descent updates on the latent-to-group matrices $\mathbf{W}_{t:,j}^{(g)}$ (Parikh & Boyd, 2014). The proximal gradient method is

$$x^{k+1} = \text{prox}_{\lambda^k g}(x^k - \lambda^k \nabla f(x^k)). \tag{13}$$

where $\lambda^k > 0$ is a step size, $\text{prox}_f(x)$ is the proximal operator for the function f. Expanding the definition of $\text{prox}_{\lambda^k g}$, we can show that the proximal step corresponds to minimizing $g(x)$ plus a quadratic approximation to $g(x)$ centered on x^k.

Lemma 2 *When $f : \boldsymbol{R}^n \to \boldsymbol{R}$ and $g : \boldsymbol{R}^n \to \boldsymbol{R} \cup \{+\infty\}$ are closed proper convex and f is differentiable. For $g(\mathbf{W}_{t:,j}^{(g)}) = \eta||\mathbf{W}_{t:,j}^{(g)}||_2$, the proximal operator is given by*

$$prox_{\lambda^k g}(\mathbf{W}_{t:,j}^{(g)}) = \frac{\mathbf{W}_{t:,j}^{(g)}}{||\mathbf{W}_{t:,j}^{(g)}||_2} \left(||\mathbf{W}_{t:,j}^{(g)}||_2 - \lambda^k\eta \right)_+ .$$

$(\psi)_+ \triangleq \max(0, \psi)$ *(Parikh & Boyd, 2014). This operator reduces the norm of $\mathbf{W}_{t:,j}^{(g)}$ by $\lambda^k\eta$, and shrink all $\mathbf{W}_{t:,j}^{(g)}$ to zero with $||\mathbf{W}_{t:,j}^{(g)}||_2 \leq \lambda^k\eta$.*

This operator is superior than other Bayesian shrinkage approaches that will result in small values but not zeros. We can use Adam (Kingma & Ba, 2015) for the remaining neural network parameters, θ and ϕ. See Alg.1 for DLGFA pseudocode.

6. Experiments

6.1. Methods Considered

In addition to the DLGFA, we also considered VAE (Kingma & Welling, 2014), adapted conditional VAE (CVAE) (Sohn et al., 2015), oi-VAE (Ainsworth et al., 2018), and group factor analysis (GFA) (Klami et al., 2015) for performance comparison. In addition, since the implementation of DP-GP-LVM (Lawrence et al., 2019) is not public available yet, we exclude the comparison with DP-GP-LVM in the following experiments.

- **VAE** and **oi-VAE**: We concatenate data across different time points and treat the concatenated data independent.

- **CVAE**: CVAE requires the label information as input to both the encoder and decoder networks. For our unsupervised problem, we assign different t for the data as its label information. I.e., the input for encoder is $[X_t, t]$ and the input for decoder is $[Z_t, t]$.

- **GFA**: we only compare DLGFA with GFA on the metabolomic data, since GFA will automatically select the latent dimension K which is a limitation for low dimensional dataset interpretation.
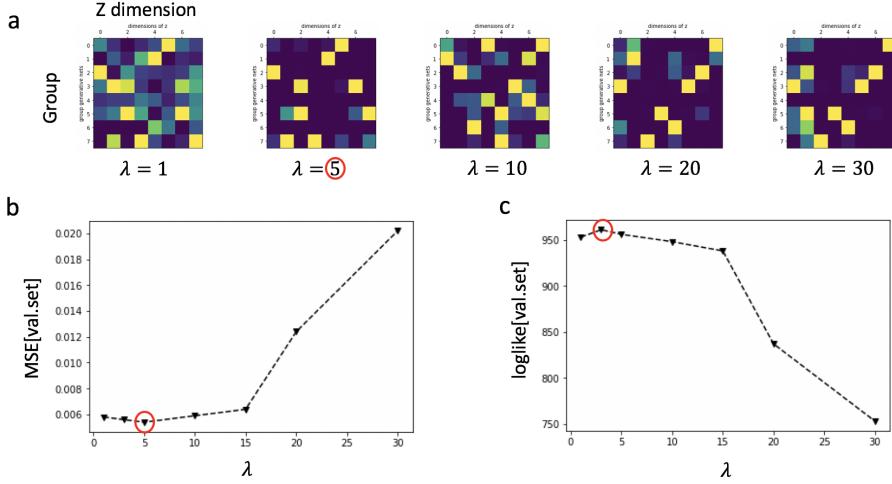
Evaluation metrics To check and validate how well the disentanglement is achieved, we propose to visualize the $\mathbf{W}_{t:,j}^{(g)}$ matrix at different time points t and quantitatively compare the MSE[test] (mean squared error on the test data) with alternative methods. For CVAE and DLGFA, we calculate the MSE[test] on the concatenated time points of the test data.

Selection on λ and K The parameter λ control the model sparsity, larger λ will imply more column-wise sparsity in $\mathbf{W}_{t:,j}^{(g)}$, we propose to select λ based on the learned $\mathbf{W}_{t:,j}^{(g)}$(Fig. 3a) to check the sparsity and the MSE[validation] (Fig. 3b). The latent dimension K is chosen based on interpretation purpose.

6.2. Artificial Data

Setup In order to visualize the performance, we generate one-bar images. The row position of the bar was taken as different time point labels, starting from the first row as time point $t = 1$ to the last row as time point $t = 8$. Thus, there are in total $T = 8$ time points.

Dataset Generation We generate 2000 8×8 one-bar images and we add normal random noises with mean 0 and standard deviation 0.05 to the entire image. We randomly select 80% (n=1600) of the image for training, 10% (n=200) for validation and 10% (n=200) for testing. For each batch we use batch size 64, so the data structure is of $8 \times 64 \times 64$

Figure 3. **Selection of** λ **on artificial data**. (a) The learned $\mathbf{W}_{t:,j}^{(g)}$ at time point $t = 8$ for different λ values; (b) The mean squared error on the validation set calculated on the concatenated time points on different λ values; (c) The loglikelihood value on the validation set on different λ values.

for DLGFA. In order to associate each dimension of $\mathbf{z}$ with a unique row in the image, we chose $K = 8$.

Results For DLGFA and CVAE, we calculate the MSE[test] on the concatenated time points of test data which is the same for the rest of the experiments. DLGFA and CVAE yield lower MSE[test] than oi-VAE and VAE, [VAE (0.0290 ± 0.0022), CVAE (0.0025 ± 0.0004), oi-VAE (0.0051 ± 0.0008), DLGFA (0.0028 ± 0.0002)](MSE$\pm$SD) in Fig. 4. DLGFA and CVAE have similar MSE[test], however, Fig. 4 a shows that CVAE removes all the background noises which is different from the truth. Meanwhile, we randomly select 64 images for each batch and replicate each image 20 times ($T = 20$) to represent the perfect time series structure, the data structure is of $20 \times 64 \times 64$, Fig. 4 c shows that DLGFA also can successfully disentangle each of the dimensions of $\mathbf{z}$ to correspond to exactly one row (group) of the image at each time point. We also tried different values of λ, and based on the results shown in Figs. 4 c and d, we chose $\lambda = 5$ for this experiment.
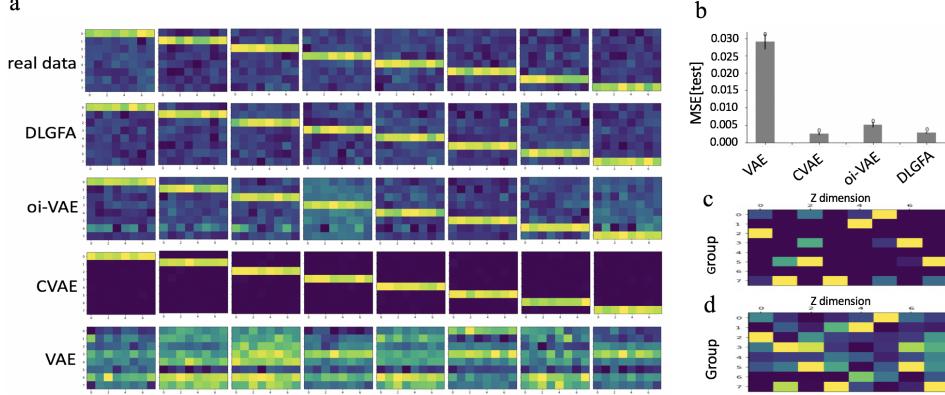
6.3. Motion Capture Data

Setup We consider the motion capture data obtained from CMU (http://mocap.cs.cmu.edu) to evaluate DLGFA's ability to handle complex longitudinal multivariate data. We use subject 7 data, which contains 11 trials of standard walking and one brisk walking recordings from the same person. For each trial, it contains different time frames of the person's moving skeleton, and it measures 59 joint angles split across 29 distinct joints. In this setting, we treat each distinct joint as a view (group), and each joint has 1 to 3 observed degrees

of freedom to represent the different group dimensions. The task is to evaluate the model's ability for sensible dynamic disentanglement, model interpretability and generalization ability.

Data For model training, we use the data from 1 to 10 trials. In total, the 10 trials training data have 3776 frames. For testing, we use the 11th trial data which has 315 frames. We set $T = 32$ to train the model with batch size 32. For each batch, the data structure is of $32 \times 32 \times 59$ for DLGFA.

Results To check different latent dimensional effects of $\mathbf{z}$, we train DLGFA on $K = 4, 8$, and 16. Fig. 5 shows the results for $K = 8$. Results for the other values of K are in the supplementary material. DLGFA displays lower MSE[test] than all the other competitors in Fig. 5 a: [VAE (0.0896 ± 0.0163), CVAE (0.0525 ± 0.0040), oi-VAE (0.0569 ± 0.0058), DLGFA (0.0326 ± 0.0056)](MSE$\pm$SD).

To evaluate the generative ability of DLGFA, we show the reconstructed images of trial 11 in Fig.5 b top row. The hidden dynamic information extracted from DLGFA generates very natural poses of human walking. In fact, there is clearly a moving pattern from the head to foot between neighboring timesteps. On the other hand, the results obtained from oi-VAE, which treats each time frame data independently, are very similar among each other, and there is no obvious trend in CVAE either. And also, We further compared the test-loglikehood on trial 11 and trial 12, which is the brisk walk data. Table 1 records the log-likelihood for both DLGFA and oi-VAE models on two testing trials with $K = 4, 8$. DLGFA has higher test log-likelihood and both methods

Figure 4. Results from experiments on artificial data with $K = 8$. (a) Reconstructed images; (b) Mean squared error on test set; (c) Learned DLGFA $\mathbf{W}_{t:,j}^{(g)}$ at time point 8 for $\lambda = 5$ and (d) $\lambda = 1$.

achieve higher test log-likelihood when the latent dimension K is larger. This indicates that DLGFA can achieve better generalization, because the brisk walking trial is very different from the training walking trials.

Fig.5 c shows that the factors change across different time points. For example, from time point 1 to 3, the first factor (first column of the left and middle images) changes from lfoot (left foot) to rfoot (right foot), factor 2 changes from rwrist (right wrist) to thorax, and factor 7 changes from rwrist to rtibia (right tibia). These changes are indeed reasonable because when we start to walk with the foot, the tibia and the thorax move accordingly (Versichele et al., 2012). The above observation demonstrates that the learned latent representation from DLGFA has an intuitive anatomical interpretation for different time points. We also provided a detailed list of the joints per latent variable dimension that are most strongly influenced by each factor in Fig.4 d. For example, factor 1 represents foot and lower back, factor 2 represents wrist, thorax and upper back, and factor 8 represents wrist, foot and hand. All these observations demonstrate that DLGFA can track the dynamic latent embeddings and provide meaningful interpretation.

6.4. Metabolomic Data

Setup In this section, we propose to analyze the data obtained from a longitudinal study (Jozefczuk et al., 2010), where one of the objectives is to compare metabolic changes of *E.coli* response to five different perturbations: cold, heat, oxidative stress, lactose diauxie, and stationary phase. The task is to evaluate the model on limited sample size studies which is common in the life sciences field.
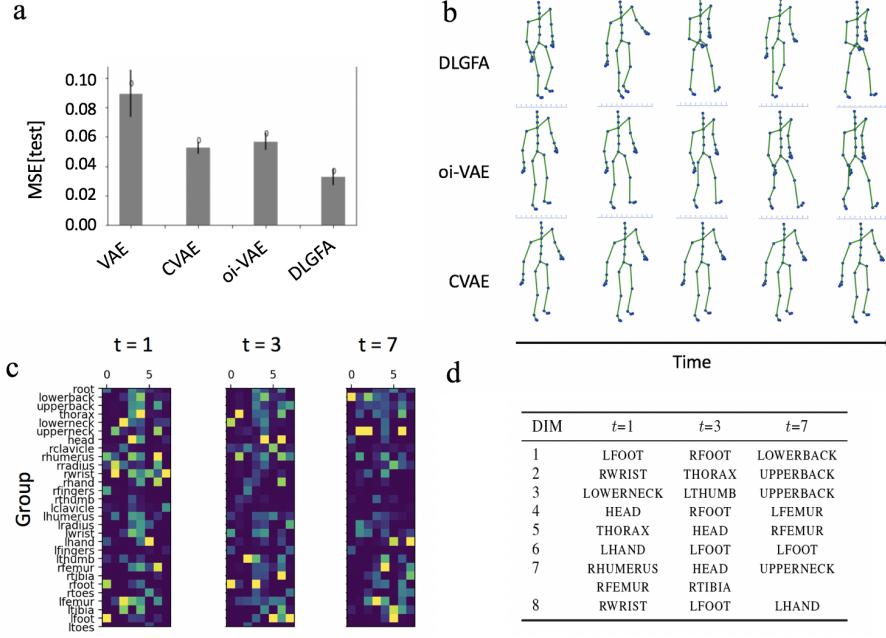
Data The dataset contains 196 metabolite expression values measured for 8 subjects at 12 different time points under five stress conditions. We treat each condition as a group

Table 1. Test log-likelihood for DLGFA and oi-VAE trained on the first 10 trials of walking data. Table includes results for a test walk (similar as training) and the brisk walk trial (different from training).

	STANDARD WALK	BRISK WALK
DLGFA(K=4)	$-93,221$	$-30,056$
OI-VAE(K=4)	-1,006,120	-598,660
DLGFA(K=8)	$-17,667$	$-36,299$
OI-VAE(K=8)	-998,849	-492,411
DLGFA(K=16)	73.3 ± 0.9	69.7 ± 1.0
OI-VAE(K=16)	67.1 ± 0.6	76.5 ± 0.5

and randomly select 6 subjects as the training set and the remaining 2 subjects as the test set. We use batch size $n = 2$, and for each batch, the data structure is of $12 \times 2 \times 980$ for DLGFA. The GFA algorithm automatically chooses $K = 30$ as the optimum dimension. We, therefore, use $K = 30$ for DLGFA in order to obtain proper comparison.

Results DLGFA has the lowest MSE[test] in Fig. 6 a. [VAE (0.0717 ± 0.0475), CVAE (0.0712 ± 0.0464), oi-VAE (0.0626 ± 0.0272), DLGFA (0.0259 ± 0.0067)](MSE±SD). The learned group-weights $\mathbf{W}_{t:,j}^{(g)}$ from DLGFA and GFA (Klami et al., 2015) are shown in Fig. 6 b. For DLGFA, it is clear that at time $t = 3$, most of the factors' variations are explained by cold and heat groups, at time $t = 5$, cold and lactose group explain most of the variations and at time $t = 8$, lactose dominates most of the variation. These results are consistent with the findings in the original paper (Jozefczuk et al., 2010). The growth curve of *E.coli* under lactose shift after perturbation is much sharper during the late time stage compared to all other conditions in their paper, which is also quite obvious at $t = 8$ in our results. On the other hand, group factor analysis results in different findings. In

Figure 5. Results for motion capture data with $K = 8$. (a) Mean squared error on trial 11; (b) Reconstructed images. All the models are trained on the first 10 trial walking data. The generated images are from trial 11 (different from training) at time point $t = 1, 7, 15, 25, 32$; (c) Learned DLGFA $\mathbf{W}_{t:,j}^{(g)}$ at time points $t = 1, 3, 7$. Each row corresponds to each group of the joints, columns represent different latent dimensions. Specifically, the values of latent dimensions are color-coded from dark blue (zero) to yellow (maximum non-zero value) to indicate the strength of the latent-to-group mappings $\mathbf{W}_{t:,j}^{(g)}$; (d) Top joints corresponding to each latent dimension determined by $\mathbf{W}_{t:,j}^{(g)}$.
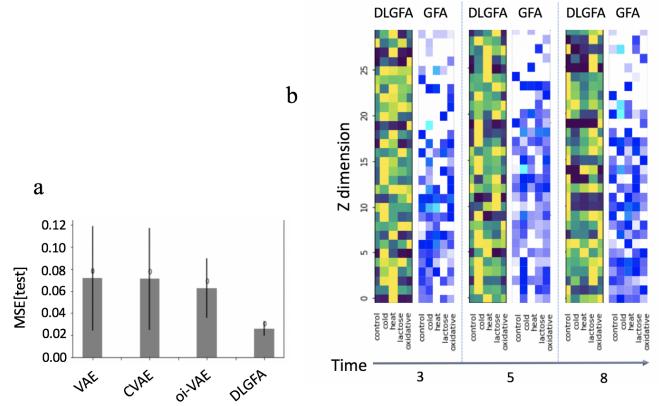
particular, at $t = 3$, oxidative condition explains the most variations and heat is the least. At $t = 5$, there is little difference among cold and oxidative stress, and at $t = 8$, there is little difference among control, heat and lactose.

Those results indicate that the generic GFA is not capable of capturing the true variations shared by groups under a complex time-series system.

7. Discussion

We have developed a nonlinear framework for longitudinal group factor analysis, namely DLGFA, with the goal of disentangling the dynamically shared latent embeddings for multiple groups (views) of data. One key feature of DLGFA is its ability to integrate the VRNN to the shared latent variables among different groups in order to model the complex sequence data and extract the dependency relationships.

Our empirical analysis on both motion capture and metabolomics data demonstrates that DLGFA successfully can extract the hidden time dependence structures. More importantly, the achieved model efficiency and interpretability does not occur at the cost of model generalization. Because DLGFA can model complex longitudinal multi-view data and result in interpretable results, we believe DLGFA

Figure 6. Results on metabolomic data with $K = 30$. (a) Mean squared error on test set; (b) Learned $\mathbf{W}_{t:,j}^{(g)}$ at time point $t = 3, 5, 8$ under control, cold, heat, lactose shift and oxidative stress from DLGFA (yellow represents dominant factor) and GFA (blue represents dominant factor).

will have wide applications in different fields ranging from computer vision, disease studies, and finance.

References

Ainsworth, S. K., Foti, N. J., Lee, A. K. C., and Fox, E. B. oi-vae: Output interpretable vaes for nonlinear group

factor analysis. In *Proceedings of the 35th International Conference on Machine Learning (ICML 18)*, 2018.

Archer, E., Park, I. M., L., Cunningham, J., and Paninski, L. Black box variational inference for state space models. In *arXiv:1511.07367*, 2015.

Argelaguet, R., Velten, B., Arnol, D., Dietrich, S., Zenz, T., Marioni, J. C., Buettner, F., Huber, W., and Stegle., O. Multi-omics factor analysis-a framework for unsupervised integration of multi-omics data sets. *Molecular Systems Biology*, 14:e8124, 2018.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179.

Chung, J., Kastner, K., Dinh, L., Goel, K., and Courville, A. A recurrent latent variable model for sequential data. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS 15)*, pp. 2980–2988, 2015.

Gao, C., Brown, C. D., and Engelhardt, B. E. A latent factor model with a mixture of sparse and dense factors to model gene expression data with confounding effects. *arXiv:1310.4792*, 2013.

Graves, A. Generating sequences with recurrent neural networks. *arXiv:1308.0850*, 2013.

Hermans, M. and Schrauwen, B. Training and analyzing deep recurrent neural networks. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 13)*, volume 1, pp. 190–198, 2013.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Jozefczuk, S., Klie, S., Catchpole, G., Szymanski, J., Cuadros-Inostroza, A., Steinhauser, D., Selbig, J., and Willmitzer, L. Metabolomic and transcriptomic stress response of escherichia coli. *Molecular Systems Biology*, 6, 2010.

Kalman, R. Mathematical description of linear dynamical systems. *J. the Society for Industrial and Applied Mathematics*, Series A, 1963.

Kingma, D. P. and Ba, J. L. Adam: A method for stochastic optimization. *arxiv:1412.6980v9*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv:1312.6114*, 2014.

Klami, A., Virtanen, S., Leppaaho, E., and Kaski, S. Group factor analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 26:2136–2147, 2015.

Kyung, M., Gill, J., and Casella, G. Penalized regression, standard errors, and bayesian lassos. *Bayesian Analysis*, 5(2):369–412, 2010.

Lawrence, A. R., Ek, C. H., and Campbell, N. D. Dp-gp-lvm: A bayesian non-parametric model for learning multivariate dependency structures. In *Proceedings of the 36th International Conference on Machine Learning (ICML 19)*, 2019.

Leppaaho, E., Ammad–ud–din, M., and Kaski, S. Gfa: Exploratory analysis of multiple data sources with group factor analysis. *Journal of Machine Learning Research*, 18:1–5, 2017.

Lucas, J. E., Kung, H., and Chi, J. A. Latent factor analysis to discover pathway-associated putative segmental aneuploidies in human cancers. *PLoS Computational Biology*, 6(9):e1000920, 2010.

Martens, J. and Sutskever, I. Learning recurrent neural networks with hessian-free optimization. In *Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML 11)*, pp. 1033–1040, 2011.

Parikh, N. and Boyd, S. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning (ICML 13)*, volume 28, pp. 1310–1318, 2013.

Pournara, I. and Wernisch, L. Factor analysis for gene regulatory networks and transcription factor activities profiles. *BMC Bioinformatics*, 8:61, 2007.

Rabiner, L. and Juang, B. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3:4–16, 1986.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. *In Advances in Neural Information Processing Systems*, pp. 3483–3491, 2015.

Sridharan, K. and Kakade, S. M. An information theoretic framework for multi-view learning. In *In Proceedings of COLT*, pp. 403–414, 2008.

Versichele, M., Neutens, T., Delafontaine, M., and Weghe, N. V. D. The use of bluetooth for analysing spatiotemporal dynamics of human movement at mass events: A case study of the ghent festivities. *Applied Geography*, 32:208–220, 2012.

Zhao, S., Gao, C., Mukherjee, S., and Engelhardt, B. E. Bayesian group factor analysis with structured sparsity. *Journal of Machine Learning Research*, 17(4):1–47, 2016.

A Supplementary Material for Deep Latent Variable Model for Learning Longitudinal Multi-view Data

A.1 Supplementary Methods

Derivation of the ELBO

We assume the following posterior distribution on latent variables

$$q(\mathbf{z} \le T | \mathbf{x} \le T) = \prod_{i=1}^{N} \prod_{t=1}^{T} q(\mathbf{z}_{it} \le T | \mathbf{x}_i \le t, \mathbf{z}_i \le t) \tag{14}$$

The evidence lower bound can be derived as follows:

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}|\mathbf{z}, \mathcal{W}, \theta) p(\mathbf{z}) p(\mathcal{W}|\gamma^2) p(\gamma^2) p(\theta) \, d\gamma^2 \, dz$$

$$= \log \int \left(\int p(\mathcal{W}, \gamma^2) \, d\gamma^2 \right) \frac{p(\mathbf{x}|\mathbf{z}, \mathcal{W}, \theta) p(\mathbf{z}) p(\theta)}{q_\phi(\mathbf{z}|\mathbf{x}) / q_\phi(\mathbf{z}|\mathbf{x})} \, dz$$

$$\ge \mathbb{E}_{q_\phi(\mathbf{z}_{\le T}|\mathbf{x}_{\le T})} \left[\sum_{t=1}^{T} -\mathrm{KL}(q_\phi(\mathbf{z}_t \mid \mathbf{x}_{\le t}, \mathbf{z}_{<t}) \| \right.$$

$$p(\mathbf{z}_t \mid \mathbf{x}_{<t}, \mathbf{z}_{<t})) + \log p(\mathbf{x}_t \mid \mathbf{z}_{\le t}, \mathbf{x}_{<t}) \bigg]$$

$$+ \log p(\theta_t) - \lambda \sum_{t=1}^{T} \sum_{g,j} ||\mathbf{W}_{t:,j}^{(g)}||_2 \tag{15}$$

Feature extraction: Since we introduced the random hidden state $\mathbf{h}_t$ for the recurrent neural network, we use neural networks $\varphi_\tau^{\mathbf{x}}$ and $\varphi_\tau^{\mathbf{z}}$ for feature extraction from $\mathbf{x}_t$ and $\mathbf{z}_t$, respectively.

- $\varphi_\tau^{\mathbf{x}}(\mathbf{x}_t) = \mathbf{W}_1 \mathbf{x}_t + b_1$

- $\varphi_\tau^{\mathbf{z}}(\mathbf{z}_t) = \mathbf{W}_3 \mathrm{relu}(\mathbf{W}_2 \mathbf{z} + b_2) + b_3$

After feature extraction from $\mathbf{x}_t$ and $\mathbf{z}_t$, then, we stack $\mathbf{x}_t$ and $\mathbf{z}_t$ with $\mathbf{h}_{t-1}$ together for the inference and generative model respectively.

Artificial data model structure:

- Encoder:
  - $\mu(\mathbf{x}_t + \mathbf{h}_{t-1}) = \mathbf{W}_1(\mathbf{x}_t + \mathbf{h}_{t-1}) + b_1$.
  - $\sigma(\mathbf{x}_t + \mathbf{h}_{t-1}) = \exp(\mathbf{W}_2(\mathbf{x}_t + \mathbf{h}_{t-1}) + b_2)$.

- Deconder:
  - $\mu(\mathbf{z}_t + \mathbf{h}_{t-1}) = \mathbf{W}_{3t}(\mathbf{z}_t + \mathbf{h}_{t-1}) + b_3$.
  - $\sigma(\mathbf{z}_t + \mathbf{h}_{t-1}) = \exp(b_4)$.

Motion capture data model structure:

- Encoder:
  - $\mu(\mathbf{x}_t + \mathbf{h}_{t-1}) = \mathbf{W}_2 \mathrm{relu}(\mathbf{W}_1(\mathbf{x}_t + \mathbf{h}_{t-1})) + b_2$.
  - $\sigma(\mathbf{x}_t + \mathbf{h}_{t-1}) = \exp(\mathbf{W}_3 \mathrm{relu}(\mathbf{W}_1(\mathbf{x}_t + \mathbf{h}_{t-1})) + b_3)$.

- Deconder:
  - $\mu(\mathbf{z}_t + \mathbf{h}_{t-1}) = \mathbf{W}_{3t} \tanh(\mathbf{z}_t + \mathbf{h}_{t-1}) + b_3$.
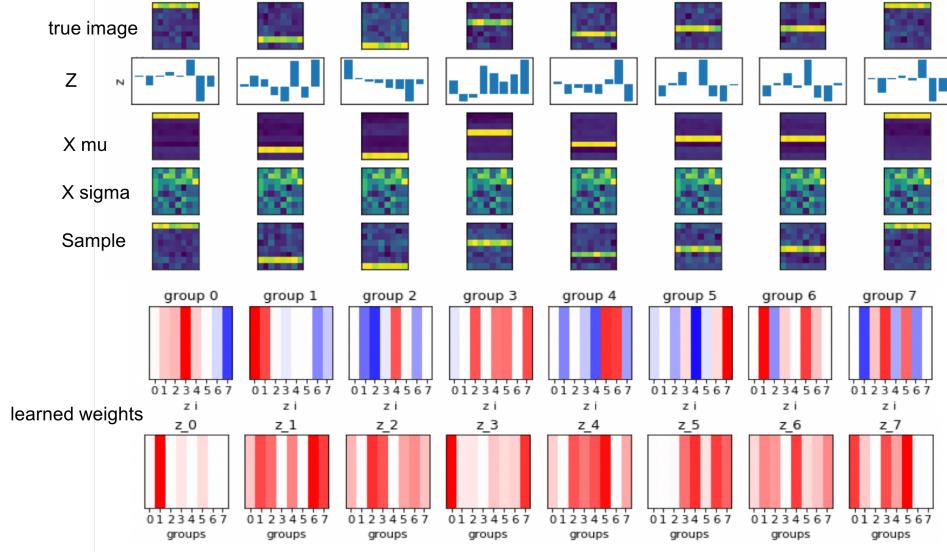 - $\sigma(\mathbf{z}_t + \mathbf{h}_{t-1}) = \exp(b_4)$.

Metabolomic data model structure:

- Encoder:
  - $\mu(\mathbf{x}_t + \mathbf{h}_{t-1}) = \mathbf{W}_2 \mathrm{relu}(\mathbf{W}_1(\mathbf{x}_t + \mathbf{h}_{t-1})) + b_2$.
  - $\sigma(\mathbf{x}_t + \mathbf{h}_{t-1}) = \exp(\mathbf{W}_3 \mathrm{relu}(\mathbf{W}_1(\mathbf{x}_t + \mathbf{h}_{t-1})) + b_3)$.

- Deconder:
  - $\mu(\mathbf{z}_t + \mathbf{h}_{t-1}) = \mathbf{W}_{3t} \tanh(\mathbf{z}_t + \mathbf{h}_{t-1}) + b_3$.
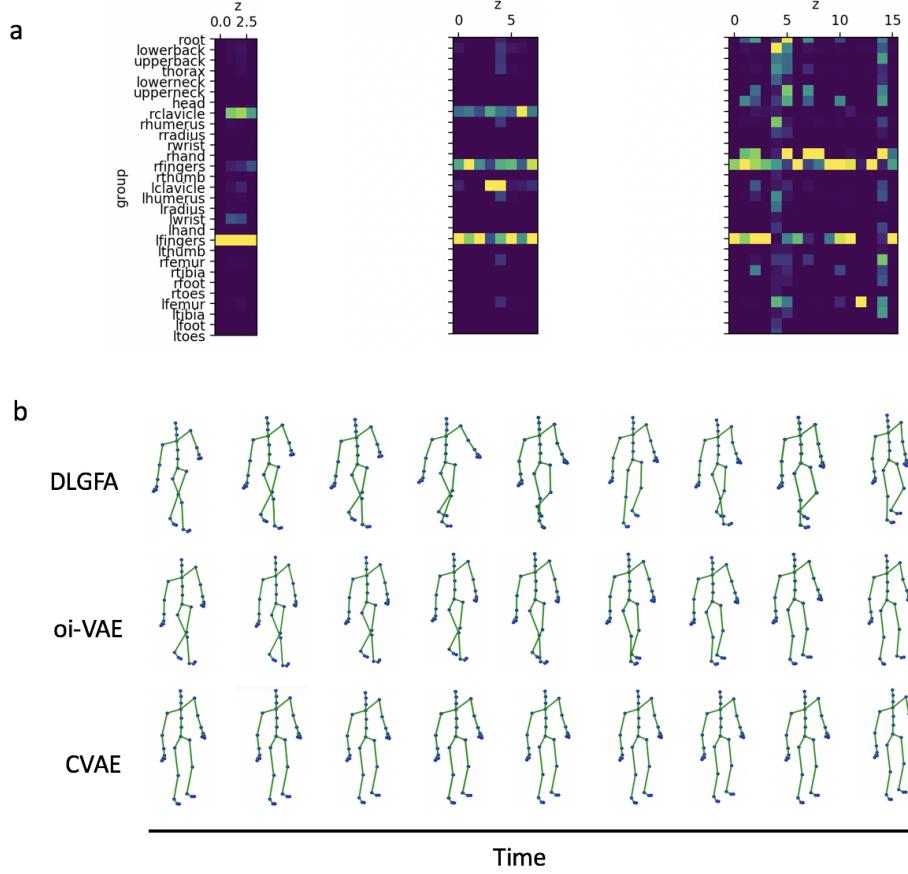 - $\sigma(\mathbf{z}_t + \mathbf{h}_{t-1}) = \exp(b_4)$.

A.2 Experimental Details

We ran Adam for the inference and generative net parameters optimization with learning rate $1e$-3. Proximal gradient descent was run on $\mathbf{W}_t$ with learning rate $1e$-4. **Artificial data**: We set $p = 1$ and chose $\lambda = 5$. For the first part of experiment, we want to select λ so we randomly selected 64 images at each iteration and replicate each image 20 times as one batch, we ran for 10,000 iterations. The data structure is $20 \times 64 \times 64$. For the second part of experiment, we assign the row position of bar as the time label, so in total we have $t = 8$ different types of images, the data structure for each batch is $8 \times 64 \times 64$. **Motion capture data**: We set $p = 8$ and chose $\lambda = 5$, and we used $T = 32$ frames and replicate each frame 32 times to stack as one batch ($32 \times 32 \times 59$) to train our model, optimization was run for 100 epochs. **Metabolomic data**: We set $p = 40$ and chose $\lambda = 10$, we randomly selected $n = 2$ as one batch, the data structure for each batch is $12 \times 2 \times 980$, we ran 10,000 epochs.

A.3 Supplementary Figures

Figure 7. **Additional results of DLGFA on artificial data**. $K = 8$, $\lambda = 5$, iteration = 10000, batch-size = 64, $t = 8$. Here, we didn't assign the t label to each image since we want to check the model sparsity induced by λ. **true image**: the training image, $\mathbf{Z}$: the sampled z values from encoder, **X mu**: the decoder mean, **X sigma**: the decoder sigma, **sample**: the reconstructed image from decoder mean and sigma, **learned weights**: the learned weights from the model.

Figure 8. **Additional results from motion capture data** (a) The learned $\mathbf{W}_{t:,j}^{(g)}$ at time point $t = 10$ for $k = 4$ (left), $k = 8$ (middle), and $k = 16$ (right) with $\lambda = 5$; (b) Reconstructed images from trial 1 (in the training set) at $t = 1, 3, 5, 7, 10, 15, 20, 25, 30$.
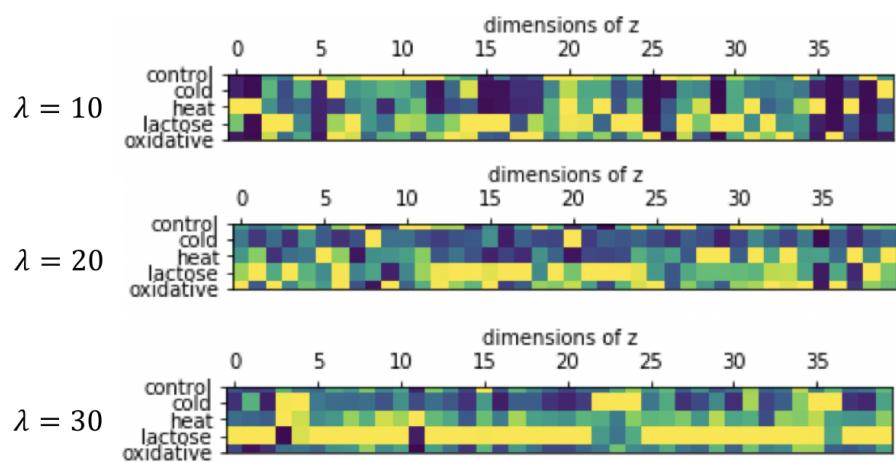
Figure 9. λ **effect on metabolomic data.** The learned $\mathbf{W}_{t:,j}^{(g)}$ at time point $t = 8$ for $\lambda = 10, 20, 30$.