BasisDeVAE: Interpretable Simultaneous Dimensionality Reduction and Feature-Level Clustering with Derivative-Based Variational Autoencoders

Dominic Danks [1 2] **Christopher Yau** [2 3 4]

Abstract

The Variational Autoencoder (VAE) performs effective nonlinear dimensionality reduction in a variety of problem settings. However, the black-box neural network decoder function typically employed limits the ability of the decoder function to be constrained and interpreted, making the use of VAEs problematic in settings where prior knowledge should be embedded within the decoder. We present DeVAE, a novel VAE-based model with a derivative-based forward mapping, allowing for greater control over decoder behaviour via specification of the decoder function in derivative space. Additionally, we show how DeVAE can be paired with a sparse clustering prior to create BasisDeVAE and perform interpretable simultaneous dimensionality reduction and feature-level clustering. We demonstrate the performance and scalability of the DeVAE and BasisDeVAE models on synthetic and real-world data and present how the derivative-based approach allows for expressive yet interpretable forward models which respect prior knowledge.

1. Introduction

Variational Autoencoders (VAEs) (Kingma & Welling, 2014) have become ubiquitous in modern machine learning and are applied in a wide range of settings, including the generation and disentangled latent coding of images (Gregor et al., 2015; Higgins et al., 2017; Kumar et al., 2018), text generation (Bowman et al., 2016a; Xu et al., 2020) and semi-supervised learning (Kingma et al., 2014; Maaløe et al., 2016). They have also played an important role in interpreting high-dimensional biological data, such as that produced in genomics settings (Lopez et al., 2018; Simidjievski et al., 2019; Qiu et al., 2020).

1.1. Extensions of VAEs

The standard VAE integrates Bayesian latent variable modelling, deep neural networks (DNNs) and variational inference, the combination of which results in a probabilistic *encoder* which maps a (high-dimensional) input onto a (low-dimensional) latent space and a corresponding probabilistic *decoder* which maps each position in the latent space onto a distribution in the observation space.

Many variants of the standard VAE have been developed, including the conditional VAE (cVAE) (Sohn et al., 2015) which conditions the behaviour of the encoder and decoder on additional fixed inputs, the neural decomposition VAE (Märtens & Yau, 2020) which imposes a functional ANOVA structure on the decoder, and the beta-VAE (Higgins et al., 2017) which incorporates additional regularisation to encourage latent variable disentanglement. Recently, an approach that combines dimensionality reduction and feature-level clustering within a VAE framework (BasisVAE) has been developed (Märtens & Yau, 2020). This type of model is of particular utility on *tabular* datasets where each feature may have a distinct standalone meaning (e.g. it represents a protein, gene or age). BasisVAE groups subsets of features together whose behaviour is similar over the latent dimensions. Figure 1 demonstrates the overall pipeline of such a simultaneous dimensionality reduction and feature-level clustering method. It can be seen that the application of dimensionality reduction via a standard VAE uncovers structure only in the *rows* (samples) of tabular data $X \in \mathbb{R}^{N \times d}$, whereas simultaneous dimensionality reduction and feature-level clustering aims to uncover structure in both the rows (samples) *and* columns (features).

These developments recognise the benefit of introducing additional structure into the flexible VAE framework in order to encourage desirable model behaviour. However, specifying particular functional characteristics with current VAE decoders remains an open challenge. For example, BasisVAE allows us to identify features which share exactly the same functional shape or dynamics, but it does not provide

[1]Institute of Cancer and Genomic Sciences, University of Birmingham, Birmingham, UK [2]The Alan Turing Institute, London, UK [3]Division of Informatics, Imaging & Data Sciences, Unversity of Manchester, Manchester, UK [4]Health Data Research UK, London, UK. Correspondence to: Dominic Danks <ddanks@turing.ac.uk>.

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

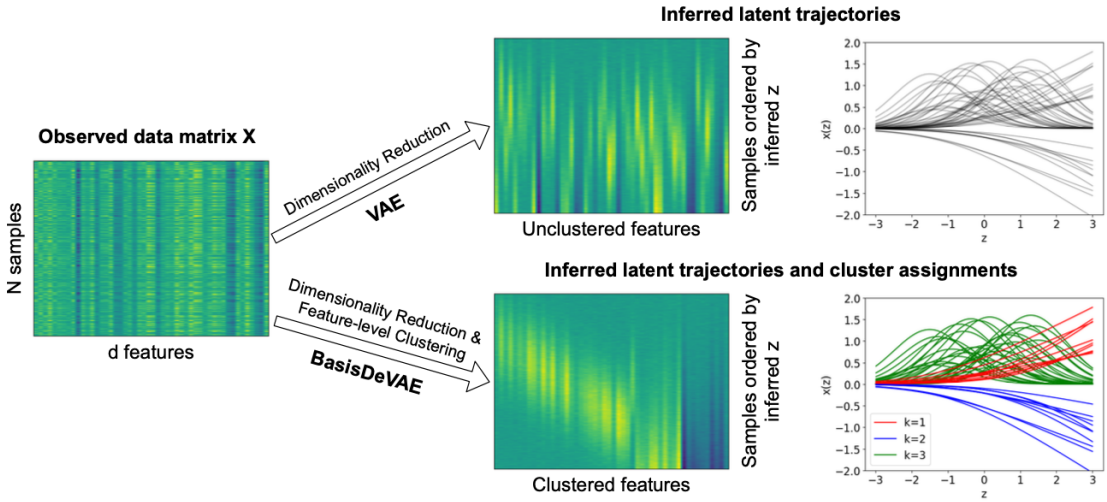

Figure 1. **Simultaneous dimensionality reduction and feature-level clustering.** Given an observed data matrix $X \in \mathbb{R}^{N \times d}$, we would like to i) learn a low-dimensional representation $\mathbf{z}$ (dimensionality reduction) and ii) cluster features according to their behaviour as a function of $\mathbf{z}$ (feature-level clustering). The standard VAE framework allows us to perform only i). With our proposed BasisDeVAE model we can perform both tasks simultaneously and guarantee *interpretability* of the inferred cluster assignments.

a mechanism to define a broader group of shared patterns among features, e.g. to cluster together all monotonically increasing functions. The ability to characterise certain functional behaviours is important in applications where *a priori* knowledge or physical laws constrain feature dynamics.

1.2. Contribution

In this work, we present two novel VAE models, namely DeVAE and BasisDeVAE, which model feature-level behaviour $\mathbf{x}$ in terms of *derivatives* with respect to the latent dimensions $\mathbf{z}$, i.e. $\frac{\partial \mathbf{x}}{\partial z_j} = f_\theta^{(j)}(z_j)$, where $f_\theta^{(j)}$ is a DNN. We show how this allows for greater control over decoder behaviour in settings where certain forms of *a priori* knowledge and/or physical constraints, such as monotonicity and transience, are desirable without having to explicitly define parametric functional forms.

Our work is particularly motivated by modelling *disease* or *biological progression* from cross-sectional data. In this problem, a cross-sectional collection of input samples is projected on to a one-dimensional latent space. If the dominant latent source of variation in the cross-sectional data is temporal, the positioning of samples in the latent space then corresponds to relative ordering in time or *pseudotime*. This type of analysis has been used to determine temporal patterns associated with biological processes, such as cellular differentiation and cancer progression, where longitudinal time series data may be difficult or impossible to collect directly. Figure 2 provides a pictorial representation of this modelling context.

We demonstrate the real-world utility of DeVAE and BasisDeVAE by applying them to synthetic data, image-derived brain pathology biomarkers from the OASIS-3 dataset (LaMontagne et al., 2019) and large-scale single-cell RNA sequencing data (Ernst et al., 2019).

2. Background

2.1. Variational Autoencoders

The original VAE framework (Kingma & Welling, 2014) is constructed from a latent variable model with the generative process

$$\begin{aligned} \mathbf{z}_i &\sim p(\mathbf{z}_i),\ i = 1, \ldots, n \\ \mathbf{x}_i|\mathbf{z}_i &\sim p_{\tilde{\theta}}(\mathbf{x}_i|\mathbf{z}_i), \end{aligned}$$

where $\mathbf{z}_i \in \mathbb{R}^p$ is the latent variable corresponding to the i-th data point $\mathbf{x}_i \in \mathbb{R}^d$ in the dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1,\ldots,N}$. The conditional likelihood $p_{\tilde{\theta}}(\mathbf{x}|\mathbf{z})$ is modelled via a DNN f_θ (the *decoder*), typically such that $\mathbb{E}[\mathbf{x}|\mathbf{z}] = f_\theta(\mathbf{z})$. For example, in the case of a Gaussian conditional likelihood, $p_{\tilde{\theta}}(x_j|\mathbf{z}) = \mathcal{N}\left(x_j|f_\theta^{(j)}(\mathbf{z}), \sigma_j^2\right)$, with j denoting output dimension, θ representing neural network parameters and $\tilde{\theta} = \left(\theta, \{\sigma_j^2\}\right)$. The prior $p(\mathbf{z})$ can be chosen to match the problem of interest but is generally restricted to distributions which are easy to sample from. In this work, we set $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_p)$ throughout.

Parameter inference for $\tilde{\theta}$ is based on maximum marginal

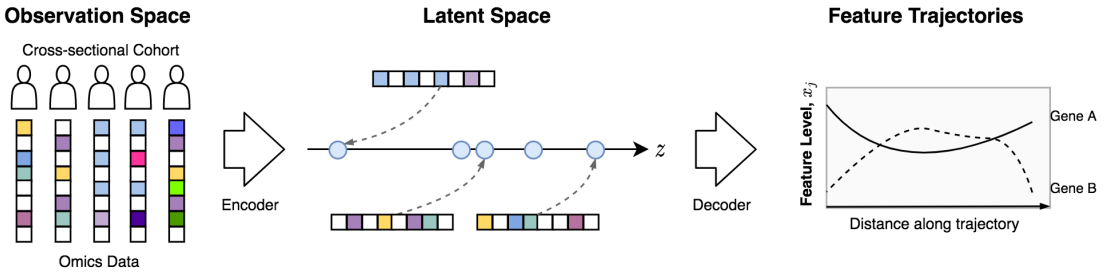

Figure 2. **Modelling biological progression.** If the dominant source of variation in a cross-sectional dataset is associated with biological or disease progression, we can use a VAE to encode and map the samples onto a one-dimensional latent space (pseudotime) and decode to understand the feature-level variability with pseudotime.

likelihood estimation, however the log-marginal likelihood

$$\log p_{\tilde{\theta}}(\mathbf{x}) = \log \int p_{\tilde{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z} \tag{1}$$

is in general intractable and cannot be optimised directly, and *amortised variational inference* is adopted. This involves introducing a variational posterior on $\mathbf{z}_i$ with distributional parameters (e.g. mean and covariance for a Gaussian) given by a neural network mapping $h_\phi(\mathbf{x}_i)$ referred to as the *encoder*. We denote this variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$. By applying Jensen's inequality to (1), a lower bound on the log-marginal likelihood of the form

$$\begin{aligned}\mathrm{ELBO}(\tilde{\theta}, \phi; \mathcal{X}) = &\sum_{i=1}^{N} \mathop{\mathbb{E}}_{\mathbf{z}_i \sim q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}_i)} \left[\log p_{\tilde{\theta}}(\mathbf{x}_i|\mathbf{z}_i)\right] \\ &- \sum_{i=1}^{N} \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}_i) \| p(\mathbf{z}_i)\right] \end{aligned} \tag{2}$$

is obtained which is typically referred to as the ELBO (see e.g. Blei et al. (2017); Jordan et al. (1999) for a detailed discussion of variational inference methodology and bound derivations). Inference in the VAE framework therefore reduces to minimisation of the loss

$$\mathcal{L}(\tilde{\theta}, \phi; \mathcal{X}) = -\mathrm{ELBO}(\tilde{\theta}, \phi; \mathcal{X}),$$

with respect to the decoder parameters $\tilde{\theta}$ and encoder parameters ϕ. This minimisation is typically performed jointly over $(\tilde{\theta}, \phi)$ using variants of stochastic gradient descent and the reparameterisation trick (Kingma & Welling, 2014; Rezende et al., 2014) with a Gaussian variational posterior such that $q_\phi\left(\mathbf{z}_i|\mathbf{x}_i\right) = \mathcal{N}\left(\mathbf{z}_i|\mu_\phi\left(\mathbf{x}_i\right), \sigma^2_\phi\left(\mathbf{x}_i\right)\right)$ in order to facilitate the calculation of ϕ gradients.

2.2. BasisVAE

BasisVAE (Märtens & Yau, 2020) enables simultaneous dimensionality reduction and feature-level clustering within the VAE framework. It achieves this by modifying the form of the decoder function, introducing additional cluster-associating latent variables into the generative model and devising an efficient collapsed variational inference scheme for learning.

The generative model of BasisVAE, in the case of a Gaussian conditional likelihood, takes the form

$$\begin{aligned}\mathbf{z}_i &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \boldsymbol{\pi}|\boldsymbol{\alpha} &\sim \mathrm{Dirichlet}(\boldsymbol{\alpha}) \\ \left(w^{j1}, \ldots, w^{jK}\right)|\boldsymbol{\pi} &\sim \mathrm{Categorical}\left(\pi_1, \ldots, \pi_K\right) \\ x_i^{(j)}|\mathbf{w}^j, \mathbf{z}_i, \tilde{\theta} &\sim \mathcal{N}\left(\sum_{k=1}^{K} w^{jk}\lambda_{jk} f^{(k)}_{\mathrm{basis}}\left(\mathbf{z}_i + \delta_{jk}\right), \sigma_j^2\right),\end{aligned}$$

where $\tilde{\theta} = (\theta, \{\lambda_{jk}\}, \{\delta_{jk}\}, \{\sigma_j\})$. Here δ_{jk} and λ_{jk} are parameters representing the amount of translation and scaling respectively associated with feature j and component k, and a Dirichlet prior is introduced on $\boldsymbol{\pi}$ to induce sparse cluster assignments. It is therefore a mixture model where each feature is assigned to one of K basis functions $f^{(k)}_{\mathrm{basis}}$.

Inference is carried out on this model using *collapsed variational inference* (Hensman et al., 2012; Hensman et al., 2015) by introducing a categorical variational posterior

$$q_{\boldsymbol{\xi}}(\mathbf{w}) = \prod_{j=1}^{d} q_{\boldsymbol{\xi}}(\mathbf{w}^j) = \prod_{j=1}^{d} \mathrm{Categorical}\left(\xi_{j1}, \ldots, \xi_{jK}\right)$$

over cluster assignments, and a variational lower bound on the marginal log-likelihood is derived by repeated application of Jensen's inequality and by additionally marginalising out $\boldsymbol{\pi}$ (see Appendix A of Märtens & Yau (2020) for details).

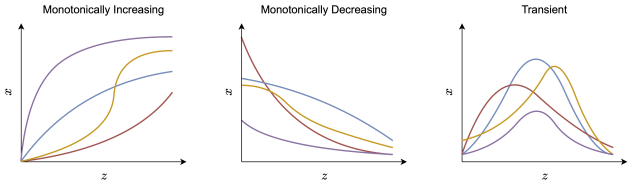

Figure 3. **Feature behaviours.** Schematic representations of the monotonically increasing, monotonically decreasing and transient feature behaviours often present within biological data that BasisDeVAE aims to capture.

The resulting loss function is

$$\mathcal{L}(\tilde{\theta}, \phi, \boldsymbol{\xi}) = -\sum_{i=1}^{N} \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathbf{y}_i)} \mathbb{E}_{q_{\boldsymbol{\xi}}(\mathbf{w})} \log p_{\tilde{\theta}}\left(\mathbf{x}_i|\mathbf{z}_i, \mathbf{w}\right) \quad (3)$$

$$- \gamma(\log \mathrm{B}(\mathbf{n} + \boldsymbol{\alpha}) - \log \mathrm{B}(\boldsymbol{\alpha})) \quad (4)$$

$$+ \gamma \mathbb{E}_{q_{\boldsymbol{\xi}}(\mathbf{w})} \log q_{\boldsymbol{\xi}}(\mathbf{w}) \quad (5)$$

$$+ \beta \sum_{i=1}^{N} \mathrm{KL}\left[q_\phi\left(\mathbf{z}_i|\mathbf{y}_i\right) \| p\left(\mathbf{z}_i\right)\right] \quad (6)$$

$$- \sum_{j=1}^{d} \sum_{k=1}^{K} \left[\log\left(\mathcal{N}\left(\delta_{jk}|0,1\right) \Gamma\left(\lambda_{jk}|0,1\right)\right)\right] \quad (7)$$

where

$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}$$

is the multivariate beta function. Term (7) arises from priors on the translation and scale parameters (these are estimated via MAP), and β and γ are hyperparameters allowing the relative importance of the prior $p(\mathbf{z})$ and sparse clustering prior to be edited in the case of large d and/or N (Higgins et al., 2017). Inference in the BasisVAE setting therefore equates to minimising $\mathcal{L}$ jointly over the decoder parameters $\tilde{\theta}$, posterior cluster assignments $\boldsymbol{\xi}$ and encoder parameters ϕ, which is performed via mini-batch gradient descent.

3. Derivative-based VAE (DeVAE)

The functional form of the decoder in a standard VAE is challenging to control due to its specification as a DNN and training being performed in the DNN's weight space. In our biological progression modelling applications, biomarkers often only exhibit a limited number of high-level behaviours, namely monotonic increases, monotonic decreases or a time-limited transient signal (see Figure 3). While a standard VAE could "learn" these behaviours given a sufficiently large number of low-noise samples, we would like to be able to robustly enforce such structures in low-sample or high-noise settings while still maintaining flexibility to model complex feature behaviours.

In order to do this, we introduce DeVAE. DeVAE specifies the decoder via its derivatives with respect to the latent variable $\mathbf{z}$. Formally, we model

$$\frac{\partial \mathbf{x}}{\partial z_p} = f_\theta^{(p)}(z_p), \quad (8)$$

with $f_\theta^{(p)}(z_p)$ a neural network-based function. Note that the *range* of values output by a neural network is easy to control. For example, setting the final activation of $f_\theta^{(p)}$ to be a softplus function ensures positivity of the derivative and hence monotonicity of $\mathbf{x}(z_p)$, whilst a sigmoid-based final activation of $f_\theta^{(p)}$ limits short-scale variation of $\mathbf{x}(z_p)$.

This derivative-based specification leads to the overall decoder output being expressed via the integral

$$\mathbf{x}(\mathbf{z}) = \mathbf{x}_0 + \int_{\mathbf{0}}^{\mathbf{z}} \mathbf{f}_\theta(\mathbf{z}') \cdot \mathrm{d}\mathbf{z}', \quad (9)$$

where $\mathbf{x}_0 = \mathbf{x}(\mathbf{0})$ and $[\mathbf{f}_\theta(\mathbf{z})]_p = f_\theta^{(p)}(z_p)$. The form of (8) implies that the integral in (9) decomposes into a sum of $\dim(\mathbf{z})$ one-dimensional integrals of the form

$$I_\theta(z) = \int_0^z f_\theta(z')\mathrm{d}z'.$$

which can be rewritten as

$$I_\theta(z) = \frac{z}{2} \int_{-1}^{1} f_\theta\left(\frac{z}{2}(u+1)\right) \mathrm{d}u$$

and evaluated using Gauss-Legendre (GL) quadrature. The explicit calculation of $I_\theta(z)$ using GL quadrature of order n (we use $n = 15$ throughout) is

$$I_\theta(z) = \frac{z}{2} \sum_{i=1}^{n} w_i f_\theta\left(\frac{z}{2}(u_i+1)\right),$$

with u_i, w_i the order n GL quadrature nodes and weights computed via Legendre polynomials (see e.g. Press et al. (2007) for details), implying that the overall computation of the decoder's output becomes

$$\mathbf{x}(\mathbf{z}) = \mathbf{x}_0 + \sum_{i,p} \frac{z_p}{2} w_i f_\theta^{(p)}\left(\frac{z_p}{2}(u_i+1)\right). \quad (10)$$

The final form of the decoder computation (10) therefore reduces to a weighted sum of neural network evaluations. Hence, backpropagation through the decoder is straightforward and the computation is fully parallelisable over data points, p, and i during optimisation using mini-batch gradient descent. We provide a GPU-aware PyTorch (Paszke et al., 2019) implementation of our approach at `https://github.com/djdanks/BasisDeVAE` and demonstrate its application to multiple settings in Section 5.

4. Basis Derivative-based VAE (BasisDeVAE)

We next embed DeVAE within the BasisVAE framework to perform simultaneous feature-level clustering and dimensionality reduction with control over the behaviour and meaning of the feature clusters. Let $g_{\tilde{\theta}}^{jk}(\mathbf{z})$ be the DeVAE-derived behaviour of feature $x_j(\mathbf{z})$ given cluster k. The generative model of BasisDeVAE is then

$$\begin{aligned}\mathbf{z}_i &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \boldsymbol{\pi}|\boldsymbol{\alpha} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\ \left(w^{j,1}, \ldots, w^{j,K}\right)|\boldsymbol{\pi} &\sim \text{Categorical}\left(\pi_1, \ldots, \pi_K\right) \\ x_i^{(j)}|\mathbf{w}^j, \mathbf{z}_i, \tilde{\theta} &\sim \mathcal{N}\left(\sum_{k=1}^{K} w^{j,k} g_{\tilde{\theta}}^{jk}(\mathbf{z}_i), \sigma_j^2\right),\end{aligned}$$

which is obtained by starting with the BasisVAE generative model (see Section 2.2) and inserting $g_{\tilde{\theta}}^{jk}(\mathbf{z}_i)$ in place of $f_{\text{basis}}^{(k)}(\mathbf{z}_i + \delta_{jk})$. Intuitively, this equates to removing the notion of feature j being obtained via the translation and scaling of one of K underlying basis functions and replacing it with the idea that feature j is described by a function from one of K families, each with interpretable properties specified via their derivatives.

In order to illustrate our approach in the context of the running example (Figure 3), let $k = 1$ and $k = 2$ correspond to monotonically increasing and decreasing behaviour respectively, and let $k = 3$ correspond to Gaussian-like transient behaviour. Additionally, let $f_\theta : \mathbb{R} \mapsto \mathbb{R}^{\dim(\mathbf{x}) \times K}$ be a neural network with a softplus output layer. Then $g_{\tilde{\theta}}^{j1}(z_i)$ and $g_{\tilde{\theta}}^{j2}(z_i)$ can be modelled as

$$\begin{aligned}g_{\tilde{\theta}}^{j1}(z_i) &= [\mathbf{x}_0]_j + \int_0^{z_i} [f_\theta(z)]_{j1}\, \mathrm{d}z \\ g_{\tilde{\theta}}^{j2}(z_i) &= [\mathbf{x}_0]_j - \int_0^{z_i} [f_\theta(z)]_{j2}\, \mathrm{d}z.\end{aligned}$$

Note that if $g_{\tilde{\theta}}^{j1}(z_i)$ or $g_{\tilde{\theta}}^{j2}(z_i)$ as defined above are mapped through a monotonically increasing function, their monotonicity properties are retained. One can therefore constrain the output range of these decoder constituents (e.g. by passing through a scaled sigmoid function) in addition to their monotonicity within this framework.

Next, for the transient component ($k = 3$), we define a Gaussian-like transient to be any function of the form $G_{t_0}(t) = A \exp\left(-[h_{t_0}(t)]^2\right)$ with $A > 0$, $h_{t_0}(t_0) = 0$, and $h_{t_0}(t)$ a monotonically increasing (or decreasing, but we restrict attention to the former without loss of generality) function. This ensures that G_{t_0} has only one stationary point, namely a unique maximum point at $t = t_0$ and that the function decays away from this point. These conditions are met by the derivative-based function

$$h_{t_0}(t) = (t - t_0) \times \text{softplus}\left(c + \int_0^t (\tau - t_0) f(\tau) \mathrm{d}\tau\right),$$

where $f(\tau)$ is any (locally) integrable function with positive range (see Appendix A for more details and a proof). Note that we use softplus for concreteness and due to its immediate availability within typical deep learning frameworks, but it could be replaced with any smooth monotonically increasing function $u : \mathbb{R} \mapsto \mathbb{R}_{>0}$ without invalidating the proof provided in Appendix A. We utilise this observation to define

$$h_{\tilde{\theta}}^{j3}(z_i) = (z_i - z_0)\text{softplus}\left(c + \int_0^{z_i} (z - z_0)\left[f_\theta(z)\right]_{j3} \mathrm{d}z\right)$$

and

$$g_{\tilde{\theta}}^{j3}(z_i) = A_j \exp\left(-\left[h_{\tilde{\theta}}^{j3}(z_i)\right]^2\right),$$

hence providing our Gaussian-like transient component. Samples of the Gaussian-like transient appearance of $g_{\tilde{\theta}}^{j3}(z_i)$ given different $h_{\tilde{\theta}}^{j3}(z_i)$ functions are provided in Appendix B.

Inference in the BasisDeVAE model is carried out using the collapsed variational inference scheme applied to BasisVAE, noting the removal of the necessity to learn the translation and scale parameters δ and λ.

We conclude this section by emphasising that it is the derivative-based approach of DeVAE that has allowed us to specify monotonic components without having to adopt parametric constraints (e.g. linearity, sigmoid-like models) or neural network weight constraints. It has also allowed us to specify a general form of Gaussian-like transient component. The BasisDeVAE framework has then enabled the data-driven *learning* of cluster assignments, linking each feature with an *interpretable* behaviour cluster.

5. Experiments[1]

5.1. Synthetic data

We first illustrate the performance of BasisDeVAE via synthetic data experiments. We generate a dataset with $N = 500$ samples and $d = 50$ features. Thirty features are randomly translated and scaled Gaussians, and the other 20 are randomly translated and scaled monotonic functions specified using softplus functions, with 10 positive and 10 negative (see Figure 4, upper-left panel). Each feature is also corrupted with Gaussian noise ($\sigma = 0.1$). Ground-truth pseudotimes were drawn uniformly to produce the

[1] All computations were performed on a Linux (Ubuntu) desktop with an Intel i7-4790K 4GHz CPU and NVIDIA GTX 980 GPU (4GB VRAM).

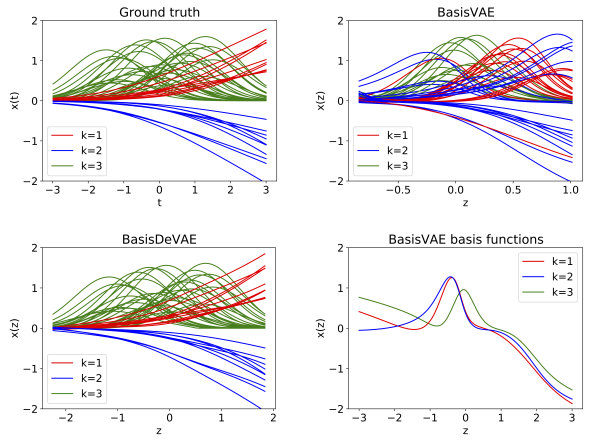

Figure 4. **Synthetic data.** BasisVAE and BasisDeVAE are applied to synthetic data with the ground-truth trajectories and clusterings shown in the upper-left panel. BasisDeVAE infers both feature behaviour and cluster assignments correctly (bottom left), whereas BasisVAE learns degenerate basis functions (bottom right), frustrating clustering performance (top right).

data but withheld from analyses. Note that both models are well-specified with respect to this data.

Figure 4 shows a comparative analysis of BasisVAE and BasisDeVAE fits to the data. We see that BasisDeVAE faithfully recovers both the pseudotemporal trajectories and cluster assignments. In contrast, while BasisVAE captures the overall pseudotemporal behaviour, it loses accuracy at the extremities of the latent dimension and produces erroneous clustering. This behaviour is explained by examining the basis functions learnt by BasisVAE (Figure 4, bottom-right panel), which do not appear to properly distinguish the monotonic and transient characteristics, instead merging the behaviours in each of the clusters.

Table 1. **Synthetic data clustering performance.** Adjusted Rand Index (ARI) values associated with the clusterings learnt from the "low z" ($\mathcal{C}_\text{l}$) and "high z" ($\mathcal{C}_\text{h}$) datasets (see Section 5.1 text). GT represents the ground-truth clustering.

	BASISVAE	**BASISDEVAE (OURS)**
$\text{ARI}(\text{GT}, \mathcal{C}_\text{L})$	0.381	**0.524**
$\text{ARI}(\text{GT}, \mathcal{C}_\text{H})$	0.102	**0.455**
$\text{ARI}(\mathcal{C}_\text{L}, \mathcal{C}_\text{H})$	0.258	**0.280**

In real-world settings, one may not observe a set of samples which spans the entire range of pseudotime. Furthermore, the observed region may vary between experiments. It is therefore of interest to test whether clustering performance agrees across different pseudotemporal segments. As an empirical example, we segment the synthetic data into two (overlapping) datasets. The first, referred to as "low z", contains only $\mathbf{x}$ values associated with $t < 1$ in the ground-truth dataset (plotted in the upper-left panel of Figure 4). The second, referred to as "high z", contains only $\mathbf{x}$ values associated with $t > -1$. Each dataset therefore totally excludes one-third of the complete trajectory. Table 1 shows the Adjusted Rand Index (ARI) values associated with the feature clusterings obtained by BasisVAE and BasisDeVAE on the "low z" and "high z" datasets, with GT denoting ground truth. It can be seen in Table 1 that the clusterings inferred by BasisDeVAE from the restricted data are more consistent with the ground truth than those of BasisVAE (rows 1 and 2), and that there is more consistency between each dataset's clusterings within the BasisDeVAE framework (row 3). These observations suggest that the presence of an underlying structured decoder model can improve extrapolation capability compared to the purely data-driven decoder approach of BasisVAE.

5.2. OASIS

The Open Access Series of Imaging Studies (OASIS) is a project led by the Knight Alzheimer Disease Research Center of Washington University to collect and openly release anonymised patient data originating from a number of studies carried out there over the past 30 years. We use OASIS-3 (LaMontagne et al., 2019) which is the latest iteration of released data and contains entries from over 2,000 MRI sessions of patients at various stages of cognitive decline.

It is well known that cognitive decline is associated with the reduction of regional brain volumes (Fox & Schott, 2004). We would therefore like the decoder of a VAE-based pseudotime model applied to tabular regional brain volume data to be monotonically decreasing, as can be easily defined within our DeVAE framework.

To test whether the explicit specification of negative monotonicity aids performance, we extract the MRI sessions with associated FreeSurfer volume segmentations to create a tabular dataset of size $N = 2047$, $d = 13$ (see Appendix C for details on the preparation of the data and meaning of each feature) and train: i) a linear VAE, ii) a standard VAE and iii) a monotonically decreasing DeVAE. Each model is trained for 50 epochs using Adam (Kingma & Ba, 2015) with a 5×10^{-3} learning rate and employs a Gaussian conditional log-likelihood in the decoder. The neural networks within the VAE and DeVAE have the same architecture apart from the final layer of the DeVAE network applying a negative softplus to enforce negative monotonicity. The pseudotemporal trajectories inferred by DeVAE for features associated with atrophy in the hippocampus, caudate, frontal lobe and thalamus are shown in Figure 5.

We quantitatively evaluate the performance of each model by performing 10 train-evaluate iterations. Each iteration con-

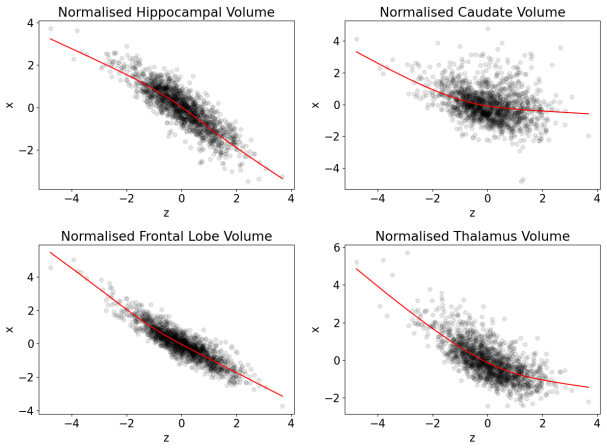

Figure 5. **OASIS regional brain volumes.** Inferred pseudotemporal profiles of 4 regional brain volumes. DeVAE extracts the monotonically decreasing pseudotemporal trajectories associated with cognitive decline.

Table 2. **OASIS performance metrics.** Test set predictive log-likelihood and ELBO values for the OASIS-3 data.

METHOD	$\log p(\mathcal{X}\|\mathcal{Z},\theta^*)$	ELBO
LINEAR VAE	-4929.1	-5798.9
VAE	-4903.7	-5779.9
DEVAE (OURS)	$\mathbf{-4879.6}$	$\mathbf{-5749.6}$

sists of i) randomly partitioning the data into an 80%/20% train/test split, ii) training on the training data and iii) evaluating two metrics on the test data. The first metric is the conditional log-likelihood $\log p(\mathcal{X}|\mathcal{Z},\theta^*)$, where $\mathcal{X}$ is the test data, θ^* are the learnt decoder parameters and $\mathcal{Z} = \{z_i\}$ with z_i the posterior mean latent variable value associated with test point $\mathbf{x}_i$. The second metric is the ELBO, i.e. the quantity given in (2), evaluated over the test data, which acts as a lower bound on the marginal log-likelihood $\log p(\mathcal{X}|\theta^*)$. We report the mean value of each metric calculated across the 10 runs in Table 2, where it can be seen that DeVAE outperforms the linear VAE and VAE with respect to both metrics. DeVAE also had the highest metric on each of the 10 runs.

5.3. Single-cell expression analysis

We next demonstrate the utility and scalability of our BasisDeVAE model by analysing a single-cell RNA sequencing (scRNA-seq) mouse spermatogenesis dataset (Ernst et al., 2019) that was also used for testing BasisVAE in Märtens & Yau (2020). The data consists of the expression values of $d = 5,216$ genes measured across $N = 8,509$ cells. The analysis task on such data is to i) recreate a representation of the temporal variable via a one-dimensional latent pseudotime z and learn the gene expression profiles with respect to z, and ii) to group features with similar expression profiles into interpretable clusters (Figure 1). These two tasks can be performed simultaneously within the BasisVAE and BasisDeVAE frameworks.

As is common in scRNA-seq analysis, we utilise a zero-inflated negative binomial conditional likelihood model in the BasisVAE and BasisDeVAE decoder (Risso et al., 2018; Lopez et al., 2018). For BasisVAE, we utilise the loss as described by (3)–(7) with $K = 3$. For BasisDeVAE, we replace the δ, λ in term (7) with the generalised Gaussians' z_0s and scale factors respectively but otherwise use the same loss. In both models we use $\beta = 10, \gamma = 1, \boldsymbol{\alpha} = \mathbf{0.1}$ and optimise using Adam with a 5×10^{-3} learning rate. We apply linear KL-annealing (Bowman et al., 2016b) to (β, γ) over the first 20% of 100 training epochs.

We have found that within the standard BasisVAE framework it is often necessary to place hand-tuned constraints on the values of the translation and scale parameters δ, λ in order to prevent collapse to a single basis function with regional behaviours. To demonstrate the effect of such constraints, we train two full BasisVAE models, one with the default constraint on λ specified within Märtens & Yau (2020)'s implementation, namely $\lambda_{jk} \in [0.25, 1.75]$, which we denote BasisVAE1, and another with a looser $\lambda_{jk} \in [0.1, 3]$ condition, which we denote BasisVAE2. We also train BasisVAE with a linear network using the same hyperparameters and default constraints to serve as a baseline. We train on a randomly sampled 90% portion of the data and reserve the remaining 10% for test-set evaluation.

Figure 6 visualises the pseudotemporal trajectories of 8 genes highlighted in Ernst et al. (2019) according to (from top to bottom) the BasisVAE1, BasisVAE2 and BasisDeVAE models. BasisDeVAE clearly clusters the 8 genes into the three distinct behaviours: monotonically increasing (red), transient (green) and monotonically decreasing (blue). It is less clear that the clusters identified by BasisVAE are necessarily plausible. For example, *Dmrtb1* is clustered with *Tex101* and *Ly6k* by BasisVAE1 but with *Stra8* and *Sohlh1* by BasisVAE2.

In Table 3, we show the training time per epoch, predictive log-likelihood and ELBO for each of the VAE models. Our results indicate that BasisDeVAE is more computationally efficient than BasisVAE and is also able to achieve superior model fits in terms of log-likelihood and ELBO metrics. We also note that BasisDeVAE tends to converge more readily than BasisVAE, achieving BasisVAE-level metric scores after less than half of the 100 utilised training epochs. This is likely caused by the enforced behavioural separation of BasisDeVAE's clusters making the learning problem less difficult. The lower training time per epoch for BasisDeVAE can be explained by noting that in BasisVAE, the compu-

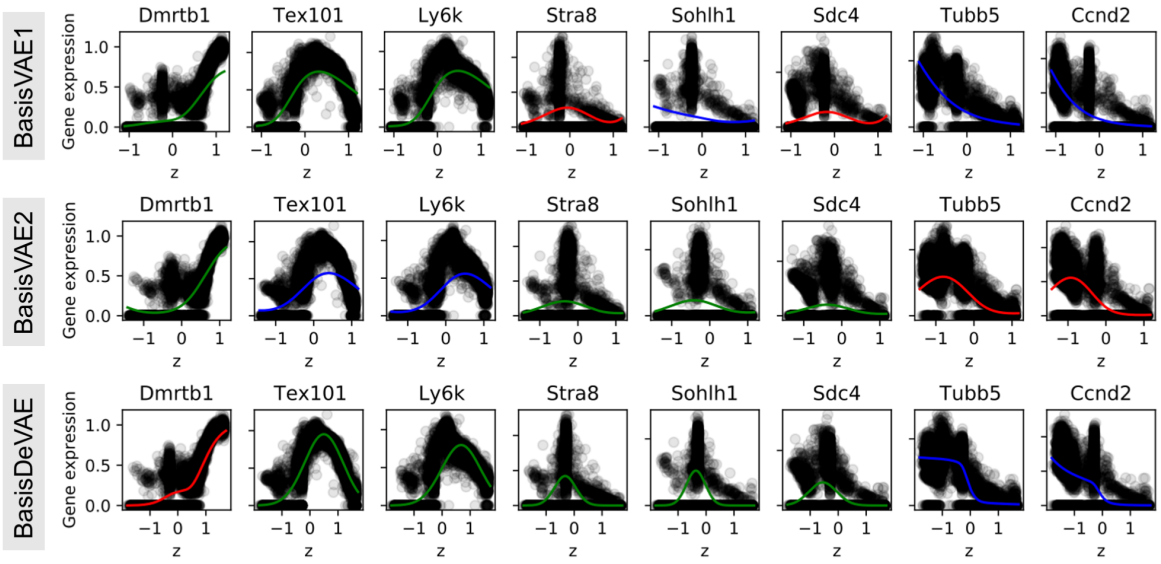

Figure 6. **Single-cell spermatogenesis data.** Inferred clustering and pseudotemporal gene expression trajectories of eight genes from Ernst et al. (2019) with BasisVAE1, BasisVAE2 and BasisDeVAE. BasisDeVAE provides a superior fit to observed data as well as greater cluster interpretability.

Table 3. **Single-cell data performance metrics.** Training time and model fit metrics for the simultaneous dimensionality reduction and feature-level clustering methods applied to the Ernst et al. (2019) scRNA-seq data.

METHOD	t_{TRAIN}/EPOCH (S)	$\log p(\mathcal{X}\|\mathcal{Z},\theta^*)$	ELBO
LINEAR	7.42 ± 0.19	-2.55×10^6	-6.86×10^7
BASISVAE1	9.24 ± 0.15	-2.54×10^6	-6.82×10^7
BASISVAE2	9.32 ± 0.13	-2.54×10^6	-6.81×10^7
BASISDEVAE (OURS)	$\mathbf{4.67 \pm 0.04}$	$\mathbf{-2.51 \times 10^6}$	$\mathbf{-6.71 \times 10^7}$

tation of $\lambda_{jk} f^{(k)}_{\text{basis}}(\mathbf{z}_i + \delta_{jk})$ for N datapoints, d features and K clusters requires NdK evaluations of a DNN with K outputs, compared with Nn evaluations of a dK-output DNN to compute the corresponding BasisDeVAE quantity.

6. Discussion

We introduced DeVAE and BasisDeVAE, two novel VAE models with decoders specified in terms of their derivatives with respect to the latent variable $\mathbf{z}$. We demonstrated that the derivative-based construction of the decoder employed in these models allows the specification of functional forms with both expressivity and interpretability, showing in particular how to specify monotonicity and transience in the context of pseudotemporal models of biological progression. We achieved state-of-the-art performance on both synthetic and real-world data examples for the problem of simultaneous dimensionality reduction and feature-level clustering.

Our work is complementary to a number of ongoing research areas. It has immediate links with other works which attempt to introduce additional structure into the VAE, such as the conditional VAE (Sohn et al., 2015), beta-VAE (Higgins et al., 2017) and functional ANOVA VAE (Märtens & Yau, 2020). Our demonstration of BasisDeVAE in the context of pseudotemporal analysis of single-cell data can be seen as a generalisation of Campbell & Yau (2018) capable of capturing any form of monotonic or Gaussian-like transient behaviour, not just sigmoidal or parametric Gaussian profiles. It also automatically assigns features to interpretable clusters without having to rely on pre-assigned genes as in Campbell & Yau (2018) or on a fully data-driven forward model as in Märtens & Yau (2020).

Our work can also be seen as another example of how the incorporation of derivative-based approaches into models can lead to improved utility and performance. The application of ODEs within machine learning has grown significantly since the widespread attention of the Neural ODE approach of Chen et al. (2018), particularly in the contexts of irregu-

larly sampled time-series modelling (Rubanova et al., 2019; Kidger et al., 2020) and, more similarly to this work, in scientific machine learning (Rackauckas et al., 2020). With significant volumes of related work continuing to emerge and increased attention on the development of associated software (Rackauckas & Nie, 2017; Bradbury et al., 2018; Chen et al., 2018), it is likely that this will remain an active area and may lead to further contributions related to ours in the context of VAEs.

The specification of functional constraints has typically been easier to adopt in a Gaussian Process (GP) framework due to the comparative ease with which one can specify functional properties relative to working in the (generally uninterpretable) weight space of a DNN. For example, Kazlauskaite et al. (2019) and (Ustyuzhaninov et al., 2020) perform constrained-warp unsupervised learning of sequence alignments in a GP framework. However, our work here shows that by operating in the derivative space we can impose certain such functional constraints with relative ease within a DNN framework.

Natural extensions of this work include formulating derivative-based decoder specifications involving higher order derivatives, as well as the consideration of the case in which $\frac{\partial \mathbf{x}}{\partial z_p}$ is allowed to depend on features $\mathbf{x}$.

Acknowledgements

We thank Kaspar Märtens for discussions and support in the development of this work. DD is supported by a Doctoral Studentship from the Alan Turing Institute (EPSRC Grant Ref: EP/N510129/1). CY is supported by an EPSRC Turing AI Acceleration Fellowship (Grant Ref: EP/V023233/1).

References

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773. URL https://doi.org/10.1080/01621459.2017.1285773.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, Berlin, Germany, August 2016a. Association for Computational Linguistics. doi: 10.18653/v1/K16-1002. URL https://www.aclweb.org/anthology/K16-1002.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space, 2016b.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Campbell, K. R. and Yau, C. A descriptive marker gene approach to single-cell pseudotime inference. *Bioinformatics*, 35(1):28–35, 06 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty498. URL https://doi.org/10.1093/bioinformatics/bty498.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 6571–6583. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

Ernst, C., Eling, N., Martinez-Jimenez, C. P., Marioni, J. C., and Odom, D. T. Staged developmental mapping and x chromosome transcriptional dynamics during mouse spermatogenesis. *Nature Communications*, 10(1):1251, 2019. doi: 10.1038/s41467-019-09182-1. URL https://doi.org/10.1038/s41467-019-09182-1.

Fox, N. C. and Schott, J. M. Imaging cerebral atrophy: normal ageing to alzheimer's disease. *The Lancet*, 363 (9406):392–394, Jan 2004. ISSN 0140-6736. doi: 10.1016/S0140-6736(04)15441-X. URL https://doi.org/10.1016/S0140-6736(04)15441-X.

Gregor, K., Danihelka, I., Graves, A., Rezende, D., and Wierstra, D. Draw: A recurrent neural network for image generation. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1462–1471, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/gregor15.html.

Hensman, J., Rattray, M., and Lawrence, N. Fast variational inference in the conjugate exponential family. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems*, volume 25, pp. 2888–2896. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/50905d7b2216bfeccb5b41016357176b-Paper.pdf.

Hensman, J., Rattray, M., and Lawrence, N. D. Fast nonparametric clustering of structured time-series. *IEEE Trans-*

actions on Pattern Analysis and Machine Intelligence, 37 (2):383–393, 2015. doi: 10.1109/TPAMI.2014.2318711.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, November 1999. ISSN 0885-6125. doi: 10.1023/A:1007665907178. URL https://doi.org/10.1023/A:1007665907178.

Kazlauskaite, I., Ek, C. H., and Campbell, N. Gaussian process latent variable alignment learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 748–757. PMLR, 2019.

Kidger, P., Morrill, J., Foster, J., and Lyons, T. Neural controlled differential equations for irregular time series, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. Semi-supervised learning with deep generative models. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems*, volume 27, pp. 3581–3589. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/d523773c6b194f37b938d340d5d02232-Paper.pdf.

Kumar, A., Sattigeri, P., and Balakrishnan, A. Variational Inference of Disentangled Latent Concepts From Unlabeled Observations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1kG7GZAW.

LaMontagne, P. J., Benzinger, T. L., Morris, J. C., Keefe, S., Hornbeck, R., Xiong, C., Grant, E., Hassenstab, J., Moulder, K., Vlassenko, A. G., Raichle, M. E., Cruchaga, C., and Marcus, D. Oasis-3: Longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease. *medRxiv*, 2019. doi: 10.1101/2019.12.13.19014902. URL https://www.medrxiv.org/content/early/2019/12/15/2019.12.13.19014902.

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, 2018. doi: 10.1038/s41592-018-0229-2. URL https://doi.org/10.1038/s41592-018-0229-2.

Maaløe, L., Sønderby, C. K., Sønderby, S. K., and Winther, O. Auxiliary deep generative models. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1445–1453, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/maaloe16.html.

Märtens, K. and Yau, C. Neural decomposition: Functional anova with variational autoencoders. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2917–2927. PMLR, 26–28 Aug 2020. URL http://proceedings.mlr.press/v108/martens20a.html.

Märtens, K. and Yau, C. BasisVAE: Translation-invariant feature-level clustering with Variational Autoencoders. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, USA, 3 edition, 2007. ISBN 0521880688.

Qiu, Y. L., Zheng, H., and Gevaert, O. Genomic data imputation with variational auto-encoders. *GigaScience*, 9(8), 08 2020. ISSN 2047-217X. doi: 10.1093/

gigascience/giaa082. URL https://doi.org/10.1093/gigascience/giaa082. giaa082.

Rackauckas, C. and Nie, Q. Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1), 2017.

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., and Edelman, A. Universal differential equations for scientific machine learning, 2020.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In Xing, E. P. and Jebara, T. (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. URL http://proceedings.mlr.press/v32/rezende14.html.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.-P. A general and flexible method for signal extraction from single-cell rna-seq data. *Nature Communications*, 9(1):284, Jan 2018. ISSN 2041-1723. doi: 10.1038/s41467-017-02554-5. URL https://doi.org/10.1038/s41467-017-02554-5.

Rubanova, Y., Chen, R. T. Q., and Duvenaud, D. K. Latent ordinary differential equations for irregularly-sampled time series. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 5320–5330. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/42a6845a557bef704ad8ac9cb4461d43-Paper.pdf.

Simidjievski, N., Bodnar, C., Tariq, I., Scherer, P., Andres Terre, H., Shams, Z., Jamnik, M., and Liò, P. Variational autoencoders for cancer data integration: Design principles and computational practice. *Frontiers in Genetics*, 10:1205, 2019. ISSN 1664-8021. doi: 10.3389/fgene.2019.01205. URL https://www.frontiersin.org/article/10.3389/fgene.2019.01205.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28, pp. 3483–3491. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf.

Ustyuzhaninov, I., Kazlauskaite, I., Ek, C. H., and Campbell, N. Monotonic gaussian process flows. In *International Conference on Artificial Intelligence and Statistics*, pp. 3057–3067. PMLR, 2020.

Xu, P., Cheung, J. C. K., and Cao, Y. On variational learning of controllable representations for text without supervision. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10534–10543. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/xu20a.html.