VARIATIONAL INTERPRETABLE DEEP CANONICAL CORRELATION ANALYSIS

Lin Qiu[1] **Vernon M. Chinchilli**[2] **Lin Lin**[3]
[1] Genentech AI [2] The Pennsylvania State University [3] Duke University
lin.qiu.stats@gmail.com vchinchi@psu.edu l.lin@duke.edu

ABSTRACT

The main idea of canonical correlation analysis (CCA) is to map different views onto a common latent space with maximum correlation. We propose a deep interpretable variational canonical correlation analysis (DICCA) for multi-view learning. The developed model extends the existing latent variable model for linear CCA to nonlinear models through the use of deep generative networks. DICCA is designed to disentangle both the shared and view-specific variations for multi-view data. To further make the model more interpretable, we place a sparsity-inducing prior on the latent weight with a structured variational autoencoder that is comprised of view-specific generators. Empirical results on real-world datasets show that our method is competitive across domains.

1 INTRODUCTION

Canonical correlation analysis (CCA) (Hotelling, 1936) is a popular two-view data analysis technique which extracts the common information between two multivariate random variables by projecting them into a space in which they are maximally correlated. CCA has been used as a standard unsupervised two-view learning (Andrew et al., 2013; Wang et al., 2015), a cross-view classification (Chang et al., 2018; Chandar et al., 2016; Kan et al., 2016), a representation learning on multiple views for prediction (Sargin et al., 2007; Dorfer et al., 2016), and for a classification from a single view when a second view is available (Arora and Livescu, 2012).

When the data comes from several different views or modalities of the same underlying source of variation, all views jointly characterize the same phenomenon. We can consider that there is shared information amongst them (or amongst subsets of them) by which they are all related. Further, we also expect that there might exist unique or private variations in each view, i.e., information specific to an individual view.

A variety of extensions of CCA have been developed to learn a shared low-dimensional feature space of multi-view data, such as kernel-based extensions (Lai and Fyfe, 2000; Akaho, 2001; Bach and Jordan, 2002). The limitation of kernel-based methods is the computational cost for large datasets since kernel-CCA requires an $N \times N$ eigenvalue decomposition, where N is the sample size. To capture the nonlinearity presented in complex data, deep neural network CCA (DCCA) was proposed (Andrew et al., 2013). DCCA maximizes the cross correlation between the nonlinear projections from the outputs of two deep nonlinear neural networks of both views. DCCA is further extended to deep CCA autoencoder (DCCAE) (Wang et al., 2015) to improve the representation learning over DCCA by leveraging autoencoders to additionally reconstruct the inputs through reconstruction error terms for the objective function. While DCCA learns embeddings that capture shared variation, it does not explicitly model view-specific noise as in probabilistic CCA (PCCA) (Bach and Jordan, 2005). PCCA is a probabilistic generative interpretation to the classical CCA.

Latent variable modelling has been used widely for providing interpretable descriptions of data. A generative model is a popular approach to achieve a compact representation through exploiting dependency structures in the observed data. The latent probabilistic version of CCA (Ghahramani, 2015) is attractive in medical applications where the data are typically of small sample sizes but large feature spaces. However, the generative interpretation of CCA will ignore nonlinear structure in complex data such as images. Lasso and group lasso are commonly used for simple interpretable models (Tibshirani, 1994; Yuan and Lin, 2006). They work by shrinking many model parameters

towards zero and have shown great success in regression models, covariance selection (Danaher et al., 2014), linear factor analysis (Hirose and Konishi, 2012), and group factor analysis (Klami et al., 2015). Commonly, sparsity-inducing penalties are considered in the convex optimization literature due to their computational tractability using proximal gradient descent. Recently, two deep variational CCAs (VCCA) were proposed (Tang et al., 2017; Wang et al., 2016) which yield a generative model with shared and view-specific factors. However, these two VCCA models are only applicable to two-view data.

Recent high-throughput techniques, such as next-generation sequencing, have generated a wide variety of multiomics datasets that enable the identification of biological functions and mechanisms via multiple facets. However, integrating these large-scale multiomics data and discovering interpretable insights are, nevertheless, challenging tasks. To address this, previous work embedded biological knowledge into the machine learning model for underlying mechanisms; e.g., interpretable deep neural network modeling (Wang et al., 2018; Ma et al., 2018). The model architecture of those methods relies heavily on the prior biological knowledge.

We present a deep interpretable CCA generative model (DICCA), in which the linear probabilistic layers are extended to deep generative multi-view networks. DICCA captures the variations of the views by a shared latent representation that describes most of the variability of multi-view data, and a set of view-specific factors. Our main contributions can be summarized as follows:

- We propose a novel generative framework for CCA which can disentangle the shared and view-specific variations from multi-view data.
- We leverage the sparsity-inducing hierarchical priors on the latent weight to achieve an interpretable model understanding.
- We evaluate our approach on real datasets to demonstrate that our algorithm achieves better performance compared to the state-of-the-art methods and have wide applications in integrating multiomics data for biomarker discovery.

2 METHODS

We now describe the proposed DICCA for multi-view data. We assume that the mth view $\mathbf{X}^m \in \mathbb{R}^{d_m \times 1}$ is independent with N co-occurring observations. $\mathbf{Z}^m \in \mathbb{R}^{K \times 1}$ denote the K-dimensional latent representation specific to the mth view for $m \in \{1, ..., M\}$, where M is the total number of views. $\mathbf{Z} \in \mathbb{R}^{K \times 1}$ denote the K-dimensional latent representation common to all views. $\mathbf{Z}$ is the *shared* latent variable capturing the shared variation across M views, while the *view-specific* latent variables $\mathbf{Z}^m$ accounts for the view-specific variation. We then write the generative process of the latent variables as:

$$\begin{aligned} \mathbf{Z} &\sim \mathcal{N}(\mathbf{0}_K, \mathbf{I}_K), \\ \mathbf{Z}^m &\sim \mathcal{N}(\mathbf{0}_K, \mathbf{I}_K). \end{aligned} \tag{1}$$

View Generator The variational autoencoder (VAE) (Kingma and Welling, 2014) propose the idea of amortized inference to perform variational inference in probabilistic models that are parameterized by deep neural networks. The limitation for deep generative models and VAE is that the learned representations are not easily interpretable due to complex interaction from latent dimensions to the observations. We consider view-specific generators and a linear latent-to-generator mapping with weights from a single latent dimension to a specific view. We model each view with separate generative networks $f^{(m)}_{\theta_m}$ parameterized by θ_m. We write the generative process of the data as:

$$\mathbf{X}^m \sim \mathcal{N}(f^{(m)}_{\theta_m}(\mathbf{\Lambda}^m \mathbf{Z} + \mathbf{W}^m \mathbf{Z}^m), \mathbf{\Psi}^m), \tag{2}$$

where $\mathbf{\Lambda}^m, \mathbf{W}^m \in \mathbb{R}^{d_m \times K}$. $\mathbf{\Psi}^m$ is a diagonal matrix containing the marginal variances of each component of $\mathbf{X}^m$. The latent representation $\mathbf{Z}$ is shared over all the view-specific generators, $\mathbf{Z}^m$ is view-specific. One of the main goals of this framework is to capture interpretable relationships between view-specific activation through the latent representation.

Interpretable Sparsity Prior We refer to $\mathbf{\Lambda}^m, \mathbf{W}^m$ as the *latent-to-view matrices*. When the jth column of the latent-to-view matrix for view m, i.e., $\mathbf{\Lambda}^{(m)}_{:,j}, \mathbf{W}^{(m)}_{:,j}$ is all zeros, then the jth latent

dimension, $\mathbf{z}_j$, will have no influence on view m. We place a hierarchical prior on the columns $\mathbf{\Lambda}_{:,j}^{(m)}$ and $\mathbf{W}_{:,j}^{(m)}$ to induce the column-wise sparsity as follows (Kyung et al., 2010):

$$\begin{aligned} \gamma_{mj}^2 &\sim \text{Gamma}\left(\frac{d_m+1}{2}, \frac{\lambda^2}{2}\right), \\ \mathbf{\Lambda}_{\cdot,j}^{(m)}, \mathbf{W}_{\cdot,j}^{(m)} &\sim \mathcal{N}(\mathbf{0}, \gamma_{mj}^2 \mathbf{I}), \end{aligned} \tag{3}$$

where Gamma$(\cdot,\cdot)$ is defined by shape and rate and d_m is the number of rows in each $\mathbf{\Lambda}^{(m)}$ and $\mathbf{W}^{(m)}$. The amount of sparsity is defined by the rate parameter λ, with larger λ implying more column-wise sparsity in $\mathbf{\Lambda}^{(m)}$ and $\mathbf{W}^{(m)}$. γ_{mj}^2 controls the column specific variation for the mth view. Marginalizing over γ_{mj}^2 induces view sparsity over the columns of $\mathbf{\Lambda}^{(m)}$ and $\mathbf{W}^{(m)}$; the maximum a posterior estimator of the resulting posterior is equivalent to a group lasso penalized objective. Different from linear factor models, the deep structure of this framework encourages the model to learn a set of $\mathbf{\Lambda}^{(m)}$ and $\mathbf{W}^{(m)}$ matrices with very small weights only to have the values revived to "appropriate" magnitudes in the following layers of $f_{\theta_m}^{(m)}$. A standard normal prior on the parameters of each generative network was placed: $\theta_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Figure 1 provides a graphical illustration of our DICCA model under m = 2.

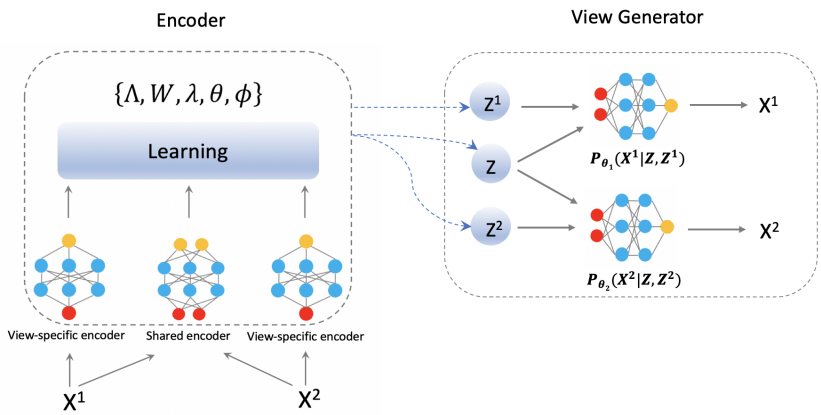

Figure 1: Graphical illustration of the DICCA model.

2.1 Variational Inference

One unique feature of VAE (Kingma and Welling, 2014) 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:

$$\log p_\theta(\mathbf{x}) \geq \underbrace{E_{q(\mathbf{z};\phi)}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - D_{KL}(q(\mathbf{z};\phi) \mid\mid p(\mathbf{z}))}_{\mathcal{L}(x;\theta,\phi)}, \tag{4}$$

where $\text{KL}(Q\|P)$ is Kullback-Leibler (KL) 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 Eq. equation 4 as tight as possible.One can train a feedforward *inference network* to find good

variational parameters $\phi(\mathbf{x})$ for a given $\mathbf{x}$, where $\phi(\mathbf{x})$ is the output of a neural network with parameters ϕ that are trained to maximize $\mathcal{L}(\mathbf{x}; \theta, \phi(\mathbf{x}))$ (Kingma and Welling, 2014).

The *KL divergence* between the approximate posterior and the prior distribution of the latent variables regularizes the prior knowledge about the latent variable for the learning algorithm. Let $\mathbf{z}^m$ and $\mathbf{z}$ represent the view-specific and shared latent variables, respectively. We can write the approximate posterior of the set of latent variables as $q_\phi(\mathbf{z}|\mathbf{x}) \prod_{m=1}^{M} q_\phi(\mathbf{z}^m|\mathbf{x}^m)$. Therefore, the *KL divergence* term can be decomposed as

$$D_{KL}[q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})] + \sum_{m=1}^{M} D_{KL}[q_\phi(\mathbf{z}^m|\mathbf{x}^m) \| p(\mathbf{z}^m)]. \tag{5}$$

2.2 Learning

Traditionally, variational inference is learned by applying stochastic gradient methods directly to the ELBO in equation 4. We borrow the idea from oi-VAE (Ainsworth et al., 2018) which extends the basic amortized inference procedure to incorporate the sparsity inducing prior over the columns of the latent-to-view matrices. We consider a collapsed variational objective function. Since our sparsity inducing prior over $\mathbf{W}^{(m)}_{\cdot,j}$ is marginally equivalent to the convex group lasso penalty, we can use proximal gradient descent on the collapsed objective and obtain the true group sparsity (Parikh and Boyd, 2014). Following the standard VAE approach of Kingma and Welling (2014), we use simple point estimates for the variational distributions on the neural network parameters $\mathcal{W} = \left(\mathbf{W}^{(1)}, \cdots, \mathbf{W}^{(m)}\right)$, $\Lambda = \left(\mathbf{\Lambda}^{(1)}, \cdots, \mathbf{\Lambda}^{(m)}\right)$, $\theta = (\theta_1, \ldots, \theta_m)$, and We take $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x})))$, where the mean and variances are parameterized by an inference network with parameters ϕ.

The collapsed objective Under $m \in \{1, ..., M\}$, the data likelihood is defined by

$$\begin{aligned} & p_\theta(\mathbf{x}^1, ..., \mathbf{x}^m, \mathbf{z}, \mathbf{z}^1, ..., \mathbf{z}^m) \\ & = p(\mathbf{z}) \prod_{m=1}^{M} p(\mathbf{z}^m) p_\theta(\mathbf{x}^m|\mathbf{z}, \mathbf{z}^m; \theta_m). \end{aligned} \tag{6}$$

We construct a collapsed variational objective function by marginalizing the γ^2_{mj} to compute $\log p_\theta(\mathbf{x})$ as:

$$\begin{aligned} & \log p_\theta(\mathbf{x}) = \log \int p(\mathbf{x}|\mathbf{z}, \mathbf{z}^1, ..., \mathbf{z}^m, \mathcal{W}, \Lambda, \theta) p(\mathbf{z}) \\ & \times \prod_{m=1}^{M} p(\mathbf{z}^m) p(\mathcal{W}|\gamma^2) \times p(\Lambda|\gamma^2) p(\gamma^2) p(\theta) \, d\gamma^2 \, dz ... \, dz^m \\ & \geq \sum_{m=1}^{M} E_{q_\phi(\mathbf{z}|\mathbf{x}^m), q_\phi(\mathbf{z}^m|\mathbf{x}^m)} \left[\log p_\theta(x^m|\mathbf{z}, \mathbf{z}^m, \mathcal{W}, \Lambda, \theta_m)\right] \\ & - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^1, ..., \mathbf{x}^m) || p(\mathbf{z})) \\ & - \sum_{m=1}^{M} D_{KL}(q_\phi(\mathbf{z}^m|\mathbf{x}^m) || p(\mathbf{z}^m)) \\ & + \sum_{m=1}^{M} \log p(\theta_m) - \lambda \sum_{m,j} ||\mathbf{\Lambda}^{(m)}_{\cdot,j}||_2 - \lambda \sum_{m,j} ||\mathbf{W}^{(m)}_{\cdot,j}||_2 \\ & = \mathcal{L}(\phi, \theta, \mathcal{W}, \Lambda). \end{aligned} \tag{7}$$

To maximize this collapsed ELBO over $\{\phi, \theta, \mathcal{W}, \Lambda\}$, we use efficient proximal gradient descent updates on the latent-to-view matrices $\mathcal{W}$ and Λ. Proximal algorithms achieve better rates of convergence than sub-gradient methods and have shown great success in solving convex objectives with group lasso penalties. We use Adam for the remaining neural net parameters, θ and ϕ.

3 RELATED WORK

Deep variational CCA (DVCCA) (Wang et al., 2016) is for two-view data representation learning. DVCCA shows that by modeling the view-specific variables that are specific to each view, DVCCA can disentangle shared and private variables and provide higher-qualify features and reconstructions. The key model architecture difference between DVCCA and our method is that DVCCA assumes the shared variation only comes from one set of data, we disentangle the shared information from all the datasets.

Two-view deep probabilistic CCA (DPCCA) was proposed (Gundersen et al., 2020) based on PCCA (Bach and Jordan, 2005) and convolutional neural networks for paired medical images and gene expression levels. The estimation of DPCCA requires PCCA to be estimated first, then the projection obtained from PCCA is passed to the convolutional neural network. The requirement for both PCCA and neural network to be available is a major limitation for reconstruction and that is why they perform worse than multimodal autoencoder (MAE) (Ngiam et al., 2011) in their reconstruction experiment. Another limitation is that DPCCA is designed specifically for paired image data.

Interpretable VAEs for nonlinear group factor analysis (oi-VAE) (Ainsworth et al., 2018) is the first generative nonlinear group latent factor model. It combines deep generative models with a hierarchical sparsity-inducing prior that leads to the ability to extract meaningful interpretations of latent-to-observed interactions when the observations are structured into groups. Deep latent variable model for learning longitudinal multi-view data (DLGFA) (Qiu et al., 2020) is an advanced temporal extension of oi-VAE, which can learn the dynamic dependency among groups through the shared latent variable and disentangle the interpretable dynamics among groups. Both oi-VAE and DLGFA use multiple decoders over the same latent variables, with the goal of having interpretable factors for the multi-view data. oi-VAE and DLGFA are more interpretable compared to DVCCA, however, they do not model the common and view-specific variations like DVCCA which leaves a challenge for modeling complex multi-view data. Variational inference for deep probabilistic canonical correlation analysis (VPCCA) (Karami and Schuurmans, 2021) can disentangle the shared and view-specific variations, however, VPCCA lacks model interpretation. Additionally, the existing models' interpretation is based on latent factor level not feature level which also poses a big application challenge for dealing with high-dimensional problems, like multi-omics data, where researchers are more interested in some particular features.

Our model is built on the variational generative model framework for efficient approximation purpose and we jointly model share and view-specific variations for complex multi-view data. Most importantly, we place sparsity-inducing prior on the latent weights to achieve feature-level interpretability.

4 EXPERIMENTS

We empirically evaluate the representation learning and model interpretability of the proposed method. The performances of the proposed method are evaluated over the following two datasets.

4.1 NOISY MNIST DATASET

Two-view noisy MNIST datasets from (Wang et al., 2015; 2016) are widely used for testing multi-view models. This dataset is generated from the MNIST dataset, the first view of the dataset is generated by rotating each image at angles randomly sampled from uniform distribution $\mathcal{U}(-\pi/4, \pi/4)$, while the second view is from randomly sampled images with the same identity to the first view but not necessarily the same image. Then, it is corrupted by random uniform noise. Thus, both views share the same identity of the digit, but they are not of the same style of the handwriting in the same class. The original training set is split into training/tuning sets of size $50K$/$10K$. The data generation process ensures that the digit identity is the only common variable underlying both views. The performance is measured on the $10K$ images in the test set. We tune the latent dimension K over [10,20,30,40,50], and fix $K_1 = K_2 = 30$, where K_1 and K_2 represent the latent dimensions for the two views of the dataset. We choose $\lambda = 1$ based on the mean squared error on the test dataset.

Disentanglement learning To evaluate the learned representation, our model should be able to reconstruct both views using the shared and view-specific latent variables. As baseline, we fit oi-VAE (Ainsworth et al., 2018), DPCCA (Gundersen et al., 2020) and VCCA (Wang et al., 2016) to both

Table 1: Reconstruction comparison on noisy two-view MNIST

Method	View 1 MSE (STD)	View 2 MSE (STD)
oi-VAE	0.059 (0.009)	0.172 (0.009)
DPCCA	0.052 (0.012)	0.134 (0.003)
VCCA	0.023 (0.011)	0.088 (0.0042)
VCCA-p	0.024 (0.011)	0.084 (0.005)
DICCA (Ours)	**0.016 (0.005)**	**0.080 (0.005)**

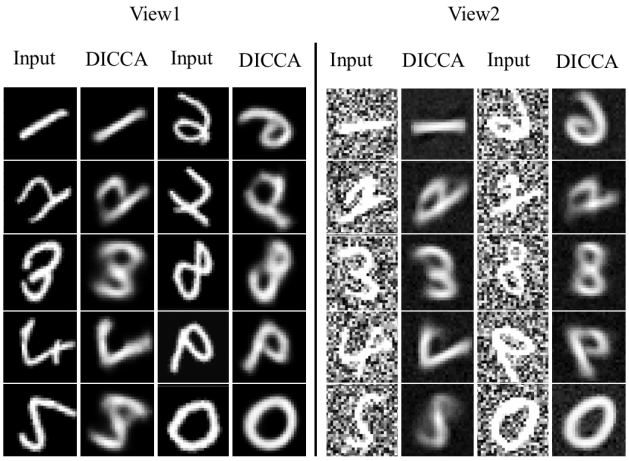

Figure 2: Reconstruction of images from the noisy MNIST test set by DICCA.

data views. VCCA-p represents VCCA-private which has view-specific latent variables in the model. We find that DICCA can reconstruct both views well relative to these baselines (Table 1). oi-VAE performs the worst since it does not model the view-specific variations. DPCCA also does not perform well because it requires optimizing PCCA in an inner loop so that the parameters are not optimized with other variables in the neural network. VCCA and VCCA-p perform better than oi-VAE and DPCCA, but they are worse than DICCA which indicates that modeling view-specific variations is more powerful to extract the hidden truth than modeling only the common variations. DICCA performs better in reconstruction than VCCA and VCCA-p. This confirms that the view-generator structure of DICCA is more suitable for learning multi-view data rather than using a single encoder of VCCA and VCCA-P. Figure 2 shows sample reconstruction of noisy MNIST dataset by DICCA for view 1 (left) and view 2 (right).

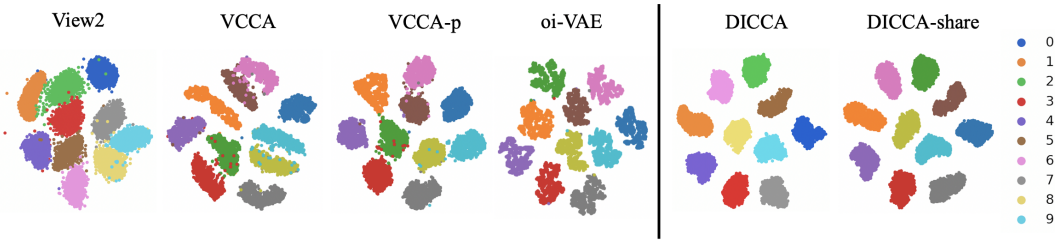

Figure 3: t-SNE visualization of the extracted latent variable $\mathbf{z}$ from images of view 2 on noisy MNIST test set by VCCA, VCCA-p, oi-VAE, DICCA, and DICCA-share. Note: DICCA represents view 2 specific latent variable $\mathbf{z}$, DICCA-share is the shared latent variable between views 1 and 2.

We can see that DICCA can capture the styles of each image very well and it can separate the background noise from the view 2 images. In addition, Figure 3 provides 2D t-SNE embedding

of the view-specific latent representations from view 2 images learned by VCCA, VCCA-p, oi-VAE, DICCA, and DICCA-share (the shared latent projections). All the methods show improved separation performance compared to the original input data. VCCA and VCCA-p perform similarly, but, there are some digits not very well separated, e.g., digit 3 and digit 2, digit 6 and digit 5. We also observe that oi-VAE has similar problem. This indicates that oi-VAE cannot capture the view-specific variations. DICCA and DICCA-share perform surprisingly well which quantitatively verify that the learned features of the images of different classes are well separated by view-specific latent variables and the shared latent projections.

4.2 GENETIC DATA

We now analyze a data set of 200 patients with chronic lymphocytic leukaemia (CLL) (Dietrich et al., 2018). This data combines *ex vivo* drug response measurements ($D = 310$) with somatic mutation status ($D = 69$), transcriptome profiling ($D = 5000$) and DNA methylation assays ($D = 4248$). Thus, there are four measurements on the same patients ($N = 200$). We apply DICCA on CLL dataset to show model interpretability by exploring group dependency relationship and latent dimensions' interpretation and annotation. After tuning, we choose $K = 10$.

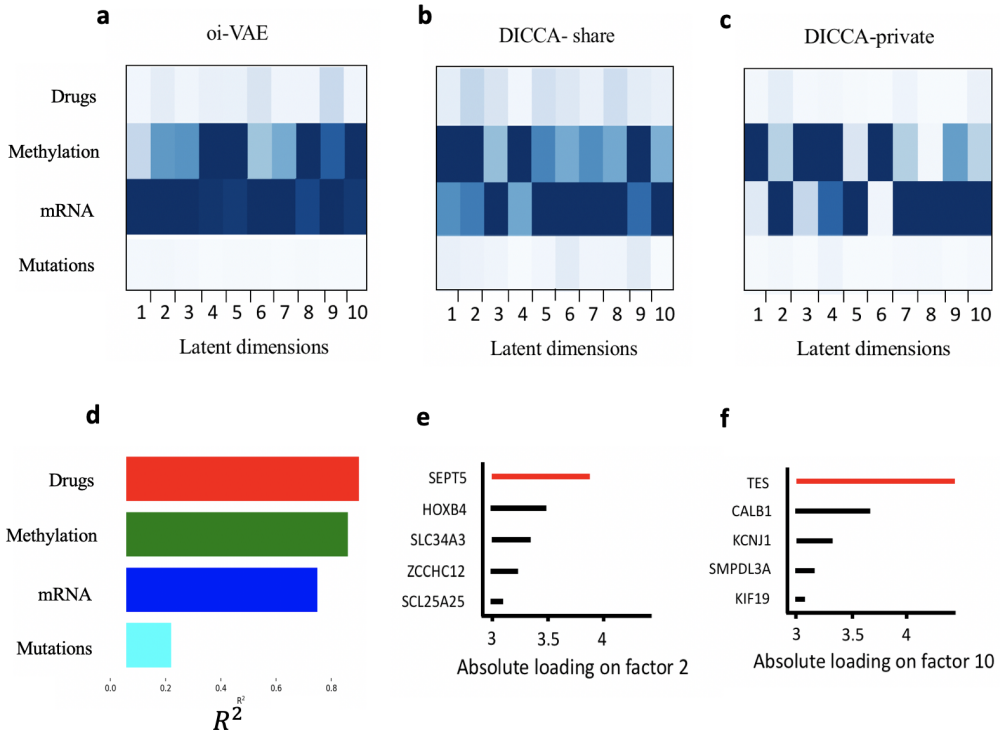

Figure 4: Results on CLL data. a-c: The learned $\mathbf{W}^{(m)}_{\cdot,j}$ from oi-VAE, the learned $\mathbf{\Lambda}^{(m)}_{\cdot,j}$ from DICCA, name as DICCA-share, the learned $\mathbf{W}^{(m)}_{\cdot,j}$ from DICCA, name as DICCA-private. Specifically, the values of latent dimensions are color-coded from white (zero) to dark blue (maximum non-zero value) to indicate the strength of the latent-to-view mappings. d: cumulative proportion of total variance explained (R^2) by each view. e-f: Absolute loadings of top features of latent dimensions 2 and 10 in the mRNA data, top feature is marked as red color.

Group dependency relationship Each latent dimension of $\mathbf{z}$ influences only a sparse subset of the observational groups. We can view the observational groups associated with a specific latent dimension. The latent weight matrix can give us a bipartite graph in which we can quickly identify correlation and independence relationships among the groups themselves. This group dependency correlation among multi-view data is attractive as an exploratory tool independent of building a generative model. In our case, we explore the shared and view-specific dependency respectively. We compare the group dependency extracted by oi-VAE (Ainsworth et al., 2018), DICCA-share and

DICCA-private in Figure 4 a, b, and c. Both oi-VAE and DICCA show that methylation and mRNA data have the dominant variations across the 10 latent dimensions. As expected, we observe the view-specific variations are captured by DICCA-private. For example, under the latent dimension 6, the dominant variation is explained by mRNA in oi-VAE which is the same as in DICCA-share. However, in DICCA-private, the variations are explained by methylation. Another example is under latent dimension 2, oi-VAE shows the dominant variations are from the mRNA, however, in DICCA-share, we observe that the shared dominant variations are from methylation, but, the DICCA-private shows the view-specific variations are from mRNA.

Latent dimension interpretation and biomarker discovery In Figure 4 d, we plot the variance explained by each view, DICCA explains 90%, 86%, 75% variations in drug, methylation, mRNA, respectively, and only 22% in mutations. This is much higher compared to MOFA (Argelaguet et al., 2018). Based on the top weights in mRNA data, factor 2 is aligned with SEPT5 which is a member of the septin gene family of nucleotide binding proteins. Disruption of septin function can disturb cytokinesis and result in large multinucleate or polyploid cells (Elzamly et al., 2018). Cancer-associated chromosomal changes often involve regions containing fragile sites. Factor 10 is aligned with TES which maps to a common fragile site on chromosome 7q31.2 designated FRA7G. TES is a negative regulator of cell growth and may act as a tumour suppressor gene that is inactivated primarily by transcriptional silencing resulting from CpG island methylation (Tobias et al., 2001).

5 Discussion

In this work, we develop a deep interpretable variational canonical correlation analysis for multi-view learning. It has been shown that following the view-generator and group sparsity formulation of the linear latent CCA model, we can obtain an interpretable learning algorithm for multi-view data. Empirical results have shown that this can efficiently disentangle the relationship among multiple views to obtain a more powerful representation. Besides the outperformed representation learning achieved by jointly modeling the share and view-specific variations, the proposed method can also have better interpretations for the latent dimensions and potentially wide applications in multiomics biomarker discovery.

References

S. K. Ainsworth, N. J. Foti, A. K. C. Lee, and E. B. Fox. oi-vae: Output interpretable vaes for nonlinear group factor analysis. In *Proceedings of the 35th International Conference on Machine Learning (ICML 18)*, volume 80, pages 119–128, 2018.

S. Akaho. A kernel method for canonical correlation analysis. In *In Proceedings of the International Meeting of the Psychometric Society*, 2001.

G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning (ICML 13)*, volume 28, pages 1247–1255, 2013.

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

R. Arora and K. Livescu. Kernel cca for multi-view learning of acoustic features using articulatory measurements. *In Symposium on Machine Learning in Speech and Language Processing*, 2012.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

F.R. Bach and M.I. Jordan. A probability interpretation of canonical correlation analysis. Technical report, University of California, Berkeley, California, 2005.

S. Chandar, M. M. Khapra, H. Larochelle, and B. Ravindran. Correlation neural networks. *Neural Computation*, 28(2):257–285, 2016.

X. B. Chang, T. Xiang, and T. M. Hospedales. Scalable and effective deep cca via soft decorrelation. In *In Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR 18)*, 2018.

P. Danaher, P. Wang, and D. M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 76(2):373–397, 2014.

S. Dietrich, M. Oleś, J. Lu, L. Sellner, S. Anders, B. Velten, B. Wu, J. Hüllein, M. da Silva Liberio, and T. Walther. Drug-perturbation-based stratification of blood cancer. *The Journal of Clinical Investigation*, 128:427–445, 2018.

M. Dorfer, G. Widmer, and G. Widmerajku. Towards deep and discriminative canonical correlation analysis. In *In Proceedings of Workshop on Multi-view Representation Learning*, 2016.

S. Elzamly, S. Chavali, V. Tonk, S. Tonk, S. Gaur, D. Tarango, and A. Torabi. Acute myeloid leukemia with kmt2a-sept5 translocation: A case report and review of the literature. *SAGE Open Medical Case Report*, 6:1–5, 2018.

Z. Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452, 2015.

G. Gundersen, B. Dumitrascu, J. T. Ash, and B. E. Engelhardt. End-to-end training of deep probabilistic cca on paired biiomedical observations. In *In Proceedings of the 35th Uncertainty in Artificial Intelligence Conference, PMLR*, volume 115, pages 945–955, 2020.

K. Hirose and S. Konishi. Variable selection via the weighted group lasso for factor analysis models. *The Canadian Journal of Statistics/ La Revue Canadienne de Statistique*, 40(2):345–361, 2012.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3-4):321–377, 1936.

M. Kan, S. Shan, H. Zhang, S. H. Lao, and X. L. Chen. Multi-view discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):188–194, 2016.

M. Karami and D. Schuurmans. Variational inference for deep probabilistic canonical correlation analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 21)*, volume 35, 2021.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16*, 2014.

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

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

P. L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural System*, 10(5):365–377, 2000.

J.Z. Ma, M. K. Yu, S. S. Fong, K. Ono, E. Sage, B. Demchak, R. Sharan, and T. Ideker. Using deep learning to model the hierarchical structure and function of a cell. *Nature Methods*, 15(4):290–298, 2018.

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML 11)*, pages 689–696, 2011.

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

L. Qiu, V. M. Chinchilli, and L. Lin. Deep latent variable model for learning longitudinal multi-view data. *arXiv preprint arXiv:2005.05210*, 2020.

M. E. Sargin, Y. Yemez, E. Erzin, and A. M. Tekalp. Audiovisual synchronization and fusion using canonical correlation analysis. *IEEE Transactions on Multimedia*, 97(7):1396–1403, 2007.

Q. M. Tang, W. R. Wang, and K. Livescu. Acoustic feature learning via deep variational canonical correlation analysis. *arXiv preprint arXiv: 1708.04673*, 2017.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

E. S. Tobias, A. F. Hurlstone, E. MacKenzie, R. McFarlane, and D. M. Black. The tes gene at 7q31.1 is methylated in tumours and encodes a novel growth-suppressing lim domain protein. *Oncogene*, 20(22):2844–1853, 2001.

D.F. Wang, S. Liu, J. Warrell, H. Won, X. Shi, F. C. P. Navarro, and D. Clarke. Comprehensive functional genomic resource and integrative model for the human brain. *Science*, 362(6420): eaat8464, 2018.

W. R. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning. In *In Proceedings of the 32nd International Conference on Machine Learning (ICML 15)*, 2015.

W. R. Wang, X. C. Yan, H.L. Lee, and K. Livescu. Deep variational canonical correlation analysis. *arXiv:1610.03454*, 2016.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

A. Supplementary Methods

A.1. Proof of additive property of KL in Equation 5.

The approximate posterior of the set of latent variables can be factorized as

$$q_\phi(\mathbf{z} \mid \mathbf{x}) \prod_{m=1}^{M} q_\phi\left(\mathbf{z}^m \mid \mathbf{z}, \mathbf{x}^m\right). \tag{8}$$

We assume independent prior distribution prior on the latent variables which leads to

$$\begin{aligned} D_{KL}[q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})] + \sum_{m=1}^{M} D_{KL}[q_\phi(\mathbf{z}^m|\mathbf{x}^m)\|p(\mathbf{z}^m)] &= \int q_\phi(\mathbf{z} \mid \mathbf{x}) \prod_{m=1}^{M} q_\phi\left(\mathbf{z}^m \mid \mathbf{z}, \mathbf{x}^m\right) \\ \times \log \frac{q_\phi(\mathbf{z} \mid \mathbf{x}) \prod_{m=1}^{M} q_\phi\left(\mathbf{z}^m \mid \mathbf{z}, \mathbf{x}^m\right)}{p(\mathbf{z}) \prod_{m=1}^{M} p\left(\mathbf{z}^m\right)} \\ = \int q_\phi(\mathbf{z} \mid \mathbf{x}) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{p(\mathbf{z})} \\ + \sum_{m=1}^{M} \int q_\phi\left(\mathbf{z}^m \mid \mathbf{z}, \mathbf{x}^m\right) \log \frac{q_\phi\left(\mathbf{z}^m \mid \mathbf{z}, \mathbf{x}^m\right)}{p\left(\mathbf{z}^m\right)}. \end{aligned} \tag{9}$$

A.2. Proximal gradient descent

A *proximal algorithm* is an algorithm for solving a convex optimization problem which uses the proximal operators of the objective terms. Consider the problem

$$\min_x f(x) + g(x), \tag{10}$$

where $f : \mathbf{R}^n \to \mathbf{R}$ and $g : \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$ are closed proper convex and f is differentiable.

The proximal gradient method is

$$x^{k+1} = \text{prox}_{\lambda^k g} \tag{11}$$

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. For $g(x) = \eta||x||_2$, the proximal operator is given by

$$\text{prox}_{\lambda^k g}(x) = \frac{x}{||x||_2}\left(||x||_2 - \lambda^k \eta\right)_+ \tag{12}$$

According to Parikh and Boyd, we know $(v)_+ \triangleq \max(0, v)$. This operator can reduce x by $\lambda\eta$, and x can be shrinked to zero under $||x||_2 \leq \lambda^k \eta$.

A.3. Latent dimension interpretation

After the model has been trained, the first step is to disentangle the variations in each view. We compute the fraction of the variance explained (R^2) per view by

$$R_m^2 = 1 - \frac{(\sum_{n,d} X_{n,d}^m - \sum_k z_{nk} W_{kd}^m - \sum_k z_{nk}^m W_{kd}^m)^2}{(\sum_{n,d} X_{n,d}^m)^2}.$$

Subsequently, each dimension is characterized by two complementary analyses:

- Ordination of the samples in factor space: Visualize a low dimensional representation of the main drivers of sample heterogeneity.
- Inspection of top features with largest weight: The loadings can give insights into the biological process underlying the heterogeneity captured by a latent dimension. We scale each weight vector by its absolute value.

B. Model architecture and training procedure

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

B.1. Two-view noisy MNIST experiments

We have view-specific encoder for each view, Encoder$_1$, Encoder$_2$ and shared encoder Encoder$_{share}$. After tuning, we use d_1 = d_2 = d_{share} = 30.

- Encoder$_1$:
  - $\mu(\mathbf{x_1}) = \mathbf{W}_1\text{relu}(\mathbf{x_1}) + b_1$.
  - $\sigma(\mathbf{x_1}) = \mathbf{W}_2\text{softplus}(\mathbf{x_1}) + b_2$.

- Encoder$_2$:
  - $\mu(\mathbf{x_2}) = \mathbf{W}_3\text{relu}(\mathbf{x_2}) + b_3$.
  - $\sigma(\mathbf{x_2}) = \mathbf{W}_4\text{softplus}(\mathbf{x_2}) + b_4$.

- Encoder$_{share}$:
  - $\mu(\mathbf{x_1} + \mathbf{x_2}) = \mathbf{W}_5(\mathbf{x_1} + \mathbf{x_2}) + b_5$.
  - $\sigma(\mathbf{x1} + \mathbf{x_2}) = \exp(\mathbf{W}_6(\mathbf{x_1} + \mathbf{x_2}) + b_6)$.

- Decoder:
  - $\mu(\mathbf{z}) = \mathbf{W_7}\tanh(\mathbf{z}) + b_7$.
  - $\sigma(\mathbf{z}) = \exp(b_8)$.

The learning rate on $\mathcal{W}$ is 1e-4 for encoder and decoder, batch size is 128. Optimization was run for 1,000 epochs.

B.2. CLL experiments

We have view-specific encoder for each view, Encoder$_{drug}$, Encoder$_{methylation}$, Encoder$_{mRNA}$, Encoder$_{mutation}$ and shared encoder Encoder$_{share}$. After tuning, we use d_{drug} = $d_{methylation}$ = d_{mRNA} = $d_{mutation}$ = d_{share} = 10.

- Encoder$_{drug}$:
  - $\mu(\mathbf{x_1}) = \mathbf{W}_1\text{relu}(\mathbf{x_1}) + b_1$.
  - $\sigma(\mathbf{x_1}) = \mathbf{W}_2\text{softplus}(\mathbf{x_1}) + b_2$.

- Encoder$_{methylation}$:
  - $\mu(\mathbf{x_2}) = \mathbf{W}_3\text{relu}(\mathbf{x_2}) + b_3$.
  - $\sigma(\mathbf{x_2}) = \mathbf{W}_4\text{softplus}(\mathbf{x_2}) + b_4$.

- Encoder$_{mRNA}$:
  - $\mu(\mathbf{x_3}) = \mathbf{W}_5\text{relu}(\mathbf{x_3}) + b_5$.
  - $\sigma(\mathbf{x_3}) = \mathbf{W}_6\text{softplus}(\mathbf{x_3}) + b_6$.

- Encoder$_{mutation}$:
  - $\mu(\mathbf{x_4}) = \mathbf{W}_7\text{relu}(\mathbf{x_4}) + b_7$.
  - $\sigma(\mathbf{x_4}) = \mathbf{W}_8\text{softplus}(\mathbf{x_4}) + b_8$.

- Encoder$_{share}$:
  - $\mu(\mathbf{x_1} + \mathbf{x_2} + \mathbf{x_3} + \mathbf{x_4})$

$= \mathbf{W}_9(\mathbf{x_1} + \mathbf{x_2} + \mathbf{x_3} + \mathbf{x_4}) + b_9$.
- $\sigma(\mathbf{x_1} + \mathbf{x_2} + \mathbf{x_3} + \mathbf{x_4})$
$= \exp(\mathbf{W}_{10}(\mathbf{x_1} + \mathbf{x_2} + \mathbf{x_3} + \mathbf{x_4}) + b_{10})$.

- Decoder:
  - $\mu(\mathbf{z}) = \mathbf{W_{11}}\tanh(\mathbf{z}) + b_{11}$.
  - $\sigma(\mathbf{z}) = \exp(b_{12})$.

The learning rate on $\mathcal{W}$ is 1e-4 for encoder and decoder, batch size is 12. Optimization was run for 2,000 epochs.