



HAL
open science

Variational graph autoencoders for multiview canonical correlation analysis

Yacouba Kaloga, Pierre Borgnat, Sundeep Prabhakar Chepuri, Patrice Abry, Amaury Habrard, Sundeep Prabhakar Chepuri

► **To cite this version:**

Yacouba Kaloga, Pierre Borgnat, Sundeep Prabhakar Chepuri, Patrice Abry, Amaury Habrard, et al.. Variational graph autoencoders for multiview canonical correlation analysis. *Signal Processing*, 2021, 188, pp.108182. 10.1016/j.sigpro.2021.108182 . hal-03436007

HAL Id: hal-03436007

<https://hal.science/hal-03436007>

Submitted on 19 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Variational Graph Autoencoders for Multiview Canonical Correlation Analysis

Yacouba Kaloga[§], Pierre Borgnat[§], Sundeep Prabhakar Chepuri^{*}, Patrice
Abry[§] and Amaury Habrard[†] ¹

[§]*Univ Lyon, Ens de Lyon, Univ. Claude Bernard, CNRS, Laboratoire de Physique,
Lyon, France*

^{*}*Department of Electrical and Communication Engineering, Indian Institute of Science,
Bangalore, India*

[†]*University of Lyon, UJM-Saint-Etienne, CNRS, Laboratoire Hubert Curien, UMR
5516, France*

Abstract

We present a novel approach for multiview canonical correlation analysis based on a variational graph neural network model. We propose a non-linear model which takes into account the available graph-based geometric constraints while being scalable to large-scale datasets with multiple views. This model combines the probabilistic interpretation of CCA with an autoencoder architecture based on graph convolutional neural network layers. Experiments with the proposed method are conducted on classification, clustering, and recommendation tasks on real datasets. The algorithm is competitive with state-of-the-art multiview representation learning techniques, in addition to being scalable and robust to instances with missing views.

Keywords: Canonical correlation analysis, Dimensionality reduction, Multiview representation learning, Graph neural networks, Variational inference

¹Supported by the IFCAM project MA/IFCAM/19/56, the ACADEMICS Grant of IDEXLYON, Univ. Lyon, PIA ANR-16-IDEX-0005, the ANR project DataRedux (ANR-19-CE46-0008), and the CBP IT test platform (ENS de Lyon, France) for ML facilities and GPU devices, operating the SIDUS solution [1]. Preliminary results were presented at the conference [2].

1. Introduction

Interconnected societies generate large amounts of structured data that frequently stem from observing a common set of objects (or sources) through different modalities. Such multiview datasets are also encountered in many different fields like computational biology [3], acoustics [4], surveillance [5], or social networks [6], to list a few. In many of these applications, datasets are structured (in graphs, trees or sequences), large and it is common that some of the views have missing entries. Although there exist many tools to analyze and study multiview datasets [7], analyzing large-scale structured multiview datasets with missing or incomplete views efficiently is still a very challenging task.

Canonical Correlation Analysis (CCA) [8, 9] can be used for multiview representation learning, by seeking latent low-dimensional representations that are common to the different views. This common representation that encodes information from different datasets can be leveraged to improve the performance of machine learning tasks, e.g., clustering [10]. There are two general approaches to CCA: *algebraic* or *probabilistic*.

The *algebraic* approaches to CCA were initially proposed for two-view data following [8] and they obtain a latent low-dimensional manifold by maximizing correlations between the projections of the different views onto it. Being nonparametric, these approaches are powerful and versatile but do not scale well to large datasets. Nevertheless, there have been numerous extensions: to the multiview setting, see [9]; or to account for nonlinear dependencies (beyond correlations), see Kernel CCA [11, 12], Deep CCA [13, 14], or Autoencoder CCA [15]. Despite significant improvements in performance, many of these approaches suffer from scalability issues [16, 17], mainly due to the prohibitive costs of the underlying eigendecomposition on which most of such methods rely on, and the difficulty to extent this settings beyond two views.

Alternatively, *probabilistic* approaches to CCA were developed: CCA solves a Bayesian inference problem [18]. As recent advances in variational autoencoders [19] made Bayesian inference scalable, the probabilistic CCA approaches gained popularity because of their potential (e.g., inference task such as generating new dataset samples) and scalability, e.g, see VCCA(p) [20], or VPCCA [21]. Using a probabilistic model, these methods scale easily to large datasets. However, they are less versatile to adjust to model mismatch or to data structure as these methods are model based.

38 Concomitantly to these advances, it was shown in [22, 23] that incor-
 39 porating the available graph-induced knowledge about the common source
 40 into multiview CCA improves performance of various machine learning tasks.
 41 We refer to this graph-aware multiview CCA method from [23] as GMCCA.
 42 However, GMCCA suffers from the involved eigendecomposition costs. In
 43 essence, there are no CCA methods that have the advantages of both worlds:
 44 being able to incorporate prior graph-based structure in the latent space and
 45 being scalable. Such a method is proposed here. The present work attempts
 46 to reconcile scalability and versatility for multiview CCA.

47 In the following, we develop a scalable multiview variational graph au-
 48 toencoder for CCA (MVGCCA), which is robust to the presence of instances
 49 with missing views in multiview datasets. Section 2 recalls some background
 50 and technical elements for multiview CCA. Section 3 describes the proposed
 51 approach and its key contributions. In particular, we show how graph struc-
 52 ture can be enforced in the common latent space while preserving scalability.
 53 Additionally, we discuss how the proposed method is robust to existence of in-
 54 stances with missing views, and how to improve that with the idea of “views
 55 dropout”. Section 4 describes the datasets that are used for numerical exper-
 56 iments. These experiments are described and discussed in Section 5. Finally,
 57 we conclude in Section 6.

58 2. Multiview CCA

59 *Multiview datasets* - We consider M -view data X where each instance² has
 60 M views, each in space \mathbb{R}^{d_m} , $m = 1, \dots, M$. We have n instances in X .
 61 The m -th view of instance i is written as $X_m^i \in \mathbb{R}^{d_m}$, and the collection of
 62 views for instance i is denoted as $X^i = \{X_m^i\}_{m=1}^M$. We also introduce the
 63 data matrix $X_m \in \mathbb{R}^{d_m \times n}$ related the m -th view, whose columns are X_m^i ,
 64 $1 \leq i \leq n$. See Figure 1 for an illustration. Not to be confused with data
 65 $X_m^i \in \mathbb{R}^{d_m}$, we denote x_m any variable vector in \mathbb{R}^{d_m} , when needed in the
 66 text.

67 *Graph* - We assume that each instance is associated to a node in a graph \mathcal{G}
 68 having a structure connecting the different instances. This graph captures
 69 closeness and similarities between the different instances. The adjacency
 70 matrix of the graph \mathcal{G} is denoted A . We denote the neighborhood of node

²The instances can be designed as sources of the views.

71 i in the graph as $\mathcal{V}(i)$ and denote $\mathcal{V}^l(i)$ as the l hop neighborhood of node.
 72 We define also the following sets of features on the nodes (which are the
 73 views of instances) associated to these neighborhoods in the following way:
 74 Let us write $d_{i,j}^A$ the length of the shortest path in graph \mathcal{G} between nodes
 75 (instances) i and j , and define:

$$\mathcal{V}^l(X^i) = \{X^j \in X | d_{i,j}^A \leq l\}$$

76 which is the set of multiview features of the neighborhood of i up to a distance
 77 l in graph \mathcal{G} (of adjacency matrix A). We define the equivalent set, limited
 78 to view m :

$$\mathcal{V}^l(X_m^i) = \{X_m^j \in X_m | d_{i,j} \leq l\}$$

79 *Notations* - For any matrix B , $B(i, :)$ denotes the i -th row and $B(:, i)$ de-
 80 notes the i -th column. The vector $[c_1, c_2, \dots, c_p]^T$ obtained by concatenation
 81 is denoted by $[c_k]_{k=1}^p$. $\|\cdot\|_F$ is the Frobenius norm; $\text{tr}(\cdot)$ is the trace oper-
 82 ator; $\llbracket i, k \rrbracket$ is the set of integers between i and k (including the boundary).
 83 For given distributions p and q , $D_{KL}(p||q)$ is the Kullback-Leibler distance
 84 between these distributions.

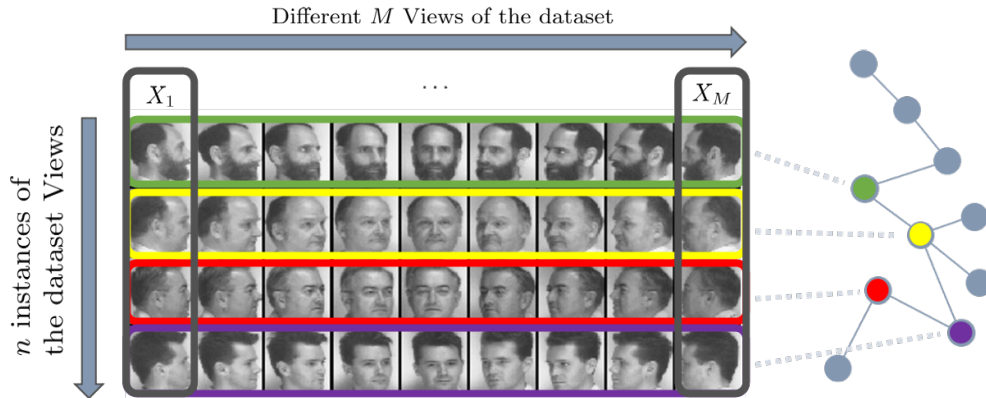


Figure 1: **Multiview dataset with graph structure.** An illustration of a multiview dataset. We have pictures taken from different angles for several people. The set of pictures of one subject is an instance and the set of pictures taken from the same angle is a view. Instances of the dataset can be related by an relationship on a graph. For example here, these instances (i.e subjects) could be part of a social network, in which people are connected according to their friendship relations.

Method	Complexity	Non Linear	>2 views	Graph	Robustness
CCA	$O(n)$	✗	✗	✗	✗
Kernel CCA	$O(n^2)^3$	✓	✗	✗	✗
Deep CCA	$O(n)$	✓	✗	✗	✗
GMCCA	$O(n^2)$	✗	✓	✓	✗
VCCA(p)	$O(n)$	✓	✗	✗	✗
VPCCA	$O(n)$	✗	✓	✗	✗
MVGCCA	$O(n)$	✓	✓	✓	✓

Table 1: Key properties of methods related to CCA. n is the number of elements in the dataset. The column entitle ‘Graph’ indicates whether or not potential graph structure is taken into account. The column ‘Robustness’ indicates whether or not the model is robust to missing views in the data.

85 *2.1. Algebraic approaches: linear CCA and extension*

86 **CCA.** Let $X_1 \in \mathbb{R}^{d_1 \times n}$ and $X_2 \in \mathbb{R}^{d_2 \times n}$ denote two views of dimension d_1
87 and d_2 for n instances. Given a dimension $d \ll \min(d_1, d_2)$, CCA seeks the
88 best projectors $U_1 \in \mathbb{R}^{d_1 \times d}$ and $U_2 \in \mathbb{R}^{d_2 \times d}$ such that the correlation between
89 $U_1^T X_1$ and $U_2^T X_2$ is maximized. This can be formulated as the following
90 optimization problem:

$$\min_{U_1, U_2} \|U_1^T X_1 - U_2^T X_2\|_F^2 \quad \text{s.t.} \quad U_m^T (X_m X_m^T) U_m = I_{d_m} \quad \text{for } m \in \{1, 2\}. \quad (1)$$

91 Introducing Σ_{11}, Σ_{22} the (regularized) correlation matrices⁴:

$$\Sigma_{mm} = \frac{1}{n-1} X_m X_m^T + r_m I_{d_m} \quad (r_m > 0) \quad (m = 1, 2) \quad (2)$$

92 and the cross correlation matrix $\Sigma_{12} = \frac{1}{n-1} X_1 X_2^T$. The solution (U_1^*, U_2^*) of
93 the problem to Eq. (1) is obtained via an eigendecomposition [13, 23]:

$$(U_1^*, U_2^*) = (\Sigma_{11}^{-\frac{1}{2}} U_d, \Sigma_{22}^{-\frac{1}{2}} V_d) \quad (3)$$

⁴ r_1 and r_2 are regularization parameters allowing to avoid degenerate correlation matrices and irrelevant correlations [24].

94 where $U_d \in \mathbb{R}^{d_1 \times d}$ (resp. $V_d \in \mathbb{R}^{d_2 \times d}$) are the d leading left (resp. right)
 95 eigenvectors of the matrix $T = \Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}}$. This formulation can easily
 96 be extended to take into account non-linear relationships by replacing, in
 97 Eq. (1), X_1 and X_2 by any non-linear function of X_1 and X_2 (e.g., Kernel
 98 CCA [11, 12], Deep CCA [13, 14], autoencoder CCA [15], etc.).

99 **Multiview CCA (MCCA)**. A direct way to extend classical CCA to mul-
 100 tiview datasets with $M > 2$ is to maximise pairwise correlations between all
 101 pairs of views:

$$\min_{(U_m)_{m=1}^M} \sum_{\substack{m=1 \\ m' > m}}^n \|U_m^T X_m - U_{m'}^T X_{m'}\|_F^2 \quad \text{s.t.} \quad U_m^T (X_m X_m^T) U_m = I_{d_m}. \quad (4)$$

102 Unfortunately this problem is well known to be NP-hard [25]. In order to
 103 overcome this issue, it is usual to introduce a unique common and low-
 104 dimensional representation $S \in \mathbb{R}^{d \times n}$. The problem is reduced to finding
 105 S and projections $\{U_m\}_{m=1}^M$ in order to maximize correlations between S and
 106 all the projected views $\{U_m^T X_m\}_{m=1}^M$. It leads to the formulation detailed in
 107 in Eq. (5) with $\gamma = 0$. This relaxation of Eq. (4) is also solved using an
 108 eigenvalue decomposition. Indeed, the matrix S^* has columns equal to the k
 109 leading eigenvectors of the matrix $\sum_{m=1}^M X_m^T (X_m X_m^T) X_m$ [22].

110 **Graph Multiview CCA (GMCCA)**. Chen et al. [23] have proposed GM-
 111 CCA as an extension of MCCA in which graph-based prior knowledge on S ,
 112 when available, can be incorporated. It can lead to an increase in clustering
 113 performance. The graph structure is taken into account by ensuring smooth-
 114 ness of S on the known graph, which is represented using the graph Laplacian
 115 matrix $L \in \mathbb{R}^{n \times n}$. By doing so, a graph-regularized CCA problem can be
 116 posed as follows:

$$\min_{(U_m)_{m=1}^M} \sum_{m=1}^M \|U_m^T X_m - S\|_F^2 + \gamma \text{tr}(S L S^T) \quad \text{s.t.} \quad S S^T = I_d. \quad (5)$$

117 The solution S^* of this problem has columns equal to the k leading eigen-
 118 vectors of the matrix $\sum_{m=1}^M X_m^T (X_m X_m^T) X_m - \gamma L$; see [22] for more details.

119

120 Due to the eigendecomposition involved in all these algebraic methods,
 121 they do not always scale well for large datasets (see Table 2). An alternative

122 method based on a variational approach applied to a probabilistic model has
 123 recently gained attention to reduce the computational cost. This relies on
 124 the work of Bach et al. [18], where it is shown that CCA has an equivalent
 125 probabilistic model.

126 *2.2. Probabilistic CCA*

127 **PCCA.** Bach et al.[18] have shown that linear CCA optimal projections, as
 128 in Eq. (3), can be obtained from a graphical model [18], where the views of
 129 an instance come from a latent variable (the common source) denoted z . Let
 130 us define a prior distribution on this latent space $p(z)$ and the conditional
 131 probability (also called decoders) for each view $p_{\theta_m}(x_m|z)$ which is the prob-
 132 ability to have a certain view x_m given the latent vector z and parameters
 133 θ_m , hence^{5,6} we have $\forall m \in \{1, 2\}, \forall z \in \mathbb{R}^d, \forall x_m \in \mathbb{R}^{d_m}$:

$$\begin{aligned} z &\sim \mathcal{N}(0, I_d); \\ x_m &\sim p_{\theta_m}(x_m|z) = \mathcal{N}(W_m z + \mu_m, \Psi_m) \end{aligned} \tag{6}$$

134 with $\mu_m \in \mathbb{R}^{d_m}$, $W_m \in \mathbb{R}^{d_m \times d}$ and $\Psi_m \in \mathbb{R}^{d_m \times d_m} \succcurlyeq 0$ (positive semidefi-
 135 nite). We collect the trainable parameters in θ_m as $\theta_m = (W_m, \mu_m, \Psi_m)$. The
 136 optimal parameter θ^* is computed by maximizing the data log-likelihood with
 137 respect to $\theta = (\theta_1, \theta_2)$:

$$\log p_{\theta}(X_1, X_2) = \sum_{i=1}^n \sum_{m=1}^2 \log \int_{\mathbb{R}^d} p_{\theta_m}(X_m^i|z)p(z)dz. \tag{7}$$

138 The parameter θ^* is the one for which the dataset $X = (X_1, X_2)$ is the most
 139 probable for p_{θ} , and therefore the best parameter to explain the data. Let
 140 us introduce the distribution p_m that is the unknown true distribution of
 141 data view m . Thanks to Bayes theorem, the optimal encoder distributions
 142 $p_{\theta_m^*}(z|X_m^i)$ are perfectly defined, and we have $\forall i \in \llbracket 0, n \rrbracket$:

$$p_{\theta_m^*}(z|X_m^i) = \frac{p_{\theta_m^*}(X_m^i|z)p(z)}{p_m(X_m^i)}. \tag{8}$$

⁵Notation $z \sim p$ means z follows distribution p .

⁶Abuse of notation: any distribution p is indiscriminately written as $p(x)$ or p .

143 The expectation of optimal decoder is then exactly the optimal projection
 144 (cf. Eq. (3)) coming from CCA (cf. Eq. (1)):

$$\mathbb{E}_{z \sim p_{\theta_m^*}}(z|X_m^i) = M_m^T U_m^{*T} X_m^i \quad (9)$$

145 The solution is known up to some arbitrary matrices $M_m \in \mathbb{R}^{d \times d}$ such that
 146 $M_1^T M_2 = P_d$ where P_d is a diagonal matrix of the first d canonical correlations
 147 [18]. In this framework, CCA has a natural multiview extension to $M > 2$.
 148 To do so we introduce as many decoders as the number of views in data. We
 149 will use such an extension, while incorporating graph regularization like in
 150 [23].

151 Yet, solving the inference problem for a model such as Eq. (6) (i.e., a
 152 multi-dimensional probability distribution) is often intractable: first because
 153 it requires maximization of the log-likelihood and thus to integrate over all
 154 the latent spaces, and, second, because the true distribution p_m of each view
 155 is not known. Even if we were able to compute θ^* , we could not compute the
 156 decoder distributions (Eq. (8)). A variational approach solves these issues,
 157 as we will see next in recalling the method of variational autoencoder [26].

158 With that, probabilistic CCA solves the problem in $O(n)$ and is thus
 159 scalable. Moreover it opens up to perform inference tasks such as generating
 160 and recovering missing views. Conversely, GMCCA has initially the advan-
 161 tage of adding a prior information (coming as a graph) over data structure,
 162 so as to compute better low dimensional representation. This prior imposes
 163 a smoothness property on this representation such that the common view
 164 S is smooth on the associated graph of Laplacian L . While it can be seen
 165 as a prior, it acts in the problem as an additional regularization term and
 166 the solution comes with the additional cost of requiring an eigendecomposi-
 167 tion. Hence the method incurs $O(n^2)$; therefore it does not scale well. Our
 168 objective is to get the best properties of both model, by forming a fully
 169 probabilistic CCA model while having such prior on the graph.

170 2.3. Variational bound and graph autoencoder

171 **VAE.** Kingma et al. [26] have shown that by introducing parametric distribu-
 172 tions $q_\eta(z|X^i) = q_\eta(z|X_1^i, X_2^i)$, with parameters η , instead of the intractable
 173 distribution $p_\theta(z|X^i) = p_\theta(z|X_1^i, X_2^i)$, one can lower bound the log-likelihood
 174 in Eq. (7). This lower bound is referred to as the **evidence lower bound**
 175 **objective** (ELBO), given as:

$$\log p_\theta(X_1, X_2) \geq \sum_{i=1}^n \mathbb{E}_{z \sim q_\eta(z|X_1^i, X_2^i)} [\log(p_\theta(X_1^i, X_2^i|z))] - D_{KL}(q_\eta(z|X_1^i, X_2^i) \| p(z)). \quad (10)$$

176 The first term of ELBO ensures a correct data reconstruction due to
 177 the encoded latent representations. The second term acts as a regularizer
 178 ensuring that the posteriors distributions q_η for each multiview instance re-
 179 main coherent in the latent space. It corresponds to the loss function of the
 180 variational autoencoders, used in most existing variational CCA methods.

181 ELBO is easier to approximate than the data log-likelihood so we max-
 182 imise this lower bound with respect to both θ and η . Moreover this formu-
 183 lation gives directly the decoders as q_{η^*} without requiring the knowledge of
 184 the true view distributions p_m .

185 **Graph VAE (GVAE).** It is possible to account for geometric structure
 186 by using the variational autoencoder extension proposed by Kipf et al. [19]
 187 for link prediction on graphs. In their single-view framework ($M = 1$),
 188 data reside on the nodes of a graph having a weighted adjacency matrix
 189 $A \in [0, 1]^{n \times n}$. On the contrary of Eq. (10) where the ELBO is a sum of terms
 190 depending only on one instance i of the data, here we have to introduce a
 191 latent matrix $Z \in \mathbb{R}^{d \times n}$ to express the probability to have graph A between
 192 n latent variables⁷.

193 Then the graph-aware ELBO loss function is defined as :

$$\mathcal{L}_{ELBO} = \mathbb{E}_{Z \sim q_\eta(Z|X, A)} [\log(p_\theta(A|Z))] - D_{KL}(q_\eta(Z|X, A) \| p(Z)), \quad (11)$$

194 where $q_\eta(Z|X, A)$ is the parametric probability distribution for encoder
 195 which is now parametrized by a graph neural network, $p(A|Z)$ is the graph
 196 decoder distribution and $p(Z) = \prod_{i=1}^n p(Z(:, i))$ is the prior on the latent
 197 space, taken as a multivariate normal distribution. Following [19], ELBO
 198 first term will ensure graph reconstruction from latent space, but it does not
 199 allow for data reconstruction (which is useless for link prediction task). On
 200 the contrary, the second term acts as a regularizer of the latent space. We

⁷Note that we could introduce the model with two latent variables only, and that would be enough to write the model for link prediction.

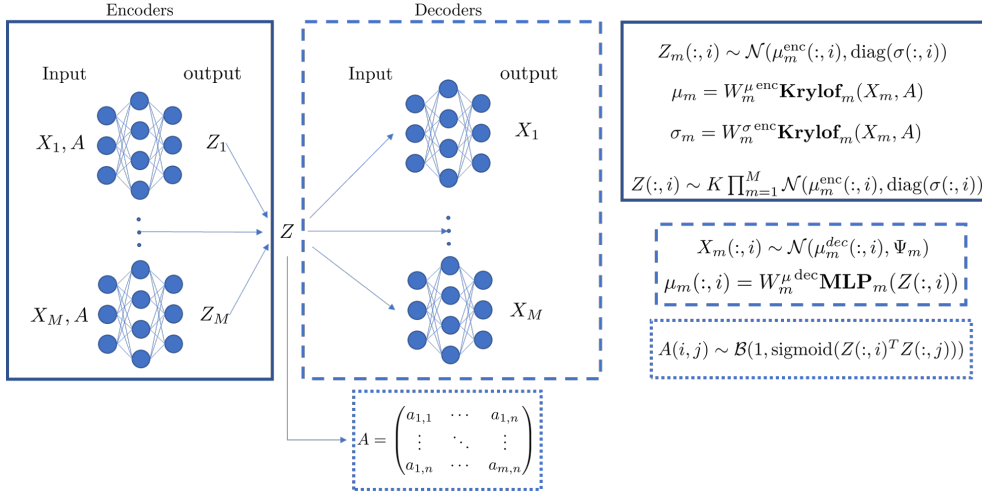


Figure 2: **Representation of MVGCCA.** All the views are encoded to their own latent space Z_m using the common graph. They are merged to form a common view Z . Finally, Z is tailored to decode all the views and original graph.

201 will use a similar approach to develop our method, while extending it for
 202 multiple views.

203 3. Variational graph MCCA

204 3.1. Model

205 We now present our contribution which consists of proposing a proba-
 206 bilistic multiview CCA model that can deal with missing views. Specifically,
 207 starting from Eq. (6), our contributions are as follows:

- 208 1. To extend the framework of [18] for $M > 2$. We introduce M decoders
 209 corresponding to the number of views. These decoders are parametrized
 210 by multilayer perceptrons (noted as **MLP**);
- 211 2. To take into account a prior graph structure on the latent space of z ;
 212 inspired by [19], we add graph decoders into the PCCA frameworks;
- 213 3. To build a model that can deal with missing views in instances of
 214 datasets.

215 To this end, we define the following probabilistic model that forms the basis
 216 of our contribution, $\forall m \in \{1, \dots, M\}, \forall z \in \mathbb{R}^d, \forall x_m \in \mathbb{R}^{d_m}$:

$$\begin{aligned}
 z &\sim \mathcal{N}(0, I_d); \\
 x_m &\sim p_{\theta_m}(x_m|z) = \mathcal{N}(\mathbf{W}_m^{\mu \text{dec}} \mathbf{MLP}_m(z), \Psi_m).
 \end{aligned}
 \tag{12}$$

217 This model relies on two hypotheses, in accordance to previous works [18,
 218 19, 26]. First, z follows a multivariate normal distribution; this assumption
 219 may look unreasonable for most data, but is often used in variational autoen-
 220 coders in the literature [26, 19] as this distribution of the prior is supposed to
 221 be mostly informative through its mean and variance, yet the specific shape
 222 is uninformative. In practice, it leads to good performance and it leads
 223 to a more convenient mathematical framework. In particular, it is easy to
 224 sample elements from Gaussian distribution and it leads to explicit formula-
 225 tion in the ELBO. The second hypothesis is that the conditional probability
 226 distributions of the view decoders are taken as multivariate Gaussian distri-
 227 butions parametrized by a multilayer perceptron for the mean, and a weight
 228 for the covariance matrix. Precisely, this hypothesis combined with the first
 229 one, allows to have an explicit form of Kullback-Leibler divergence on ELBO
 230 (Eq. (24)).

231 Next, we take into account a prior graph structure on the latent space of z
 232 using techniques inspired by [19]. Specifically we introduce a graph condi-
 233 tional probability distribution (graph decoder) which gives the probability
 234 to have an adjacency matrix $A \in [0, 1]^{n \times n}$ given the n latent space vectors
 235 concatenated in the matrix $Z \in \mathbb{R}^{d \times n}$:

$$\begin{aligned} Z(:, i) &\sim \mathcal{N}(0, I_d); \\ A &\sim p_g(A|Z). \end{aligned} \tag{13}$$

236 In this context, we assume that all links of the graph are independent. Hence
 237 we introduce a weight decoder distribution $p_l(a = 1|z, z') \sim \mathcal{B}(1, \ell(z^T z'))$,
 238 parametrized by a Bernoulli law \mathcal{B} , where $\ell(\cdot)$ is the logistic sigmoid function
 239 and $a \in [0, 1]$ is the possible weight of a link. Thus the probability to have a
 240 graph defined by an adjacency matrix A given Z is:

$$p_g(A|Z) = \prod_{i=1}^n \prod_{j=1}^n p_l(A_{i,j}|Z(:, i), Z(:, j)). \tag{14}$$

241 3.2. Evidence lower bound objective

242 Using the hypothesis that all the views are independent one from others, as
 243 well as from the links, the log-likelihood can be written in the following form:

$$\log p_\theta(X, A) = \log (p_\theta(\{X_m\}_{m=1}^M) p_g(A)) . \tag{15}$$

244 In this equation, $p_\theta(\{X_m\}_{m=1}^M)$ is the joint probability on all the views, X_1
 245 to X_M ; hence, this equation gives the probability of obtaining the multi-
 246 view dataset $X = \{X_m\}_{m=1}^M$ with the graph A given the model p_θ . We can
 247 explicitly express these probabilities as in Eq. (7):

$$\begin{aligned} \log p_\theta(\{X_m\}_{m=1}^M) &= \sum_{i=1}^n \sum_{m=1}^M \log \int_{\mathbb{R}^d} p_{\theta_m}(X_m^i|z)p(z)dz; \\ \log p_g(A) &= \sum_{i=1}^n \sum_{j=1}^n \log \int_{\mathbb{R}^d} p_l(A_{i,j}|z, z')p(z)p(z')dzdz'. \end{aligned} \quad (16)$$

248 Since this computation is also intractable, we follow the methodology of
 249 Kingma et al. [26] in order to define an efficient approach. We introduce a
 250 parametric distribution written as in Eq. (17). As we are going to parametrize
 251 this distribution with graph neural networks, we need the notation \mathcal{V}^l intro-
 252 duced earlier to refer to the features in the neighborhood of a node up to a
 253 distance l in the graph, i.e, we have:

$$q_\eta(Z|X, A) = \prod_{i=1}^n q_\eta(Z(:, i)|\mathcal{V}^l(X^i), A) = \prod_{i=1}^n \prod_{m=1}^M q_{\eta_m}(Z(:, i)|\mathcal{V}^l(X_m^i), A). \quad (17)$$

254 Were η_m is the trainable parameter for view m distribution encoder q_{η_m} and
 255 we have $\eta = (\eta_1, \dots, \eta_M)$. The parameter l depends on the parametrization
 256 of the graph neural networks, which will be decided on the experimental
 257 part. Let us consider the Kullback-Leibler divergence between the parametric
 258 decoders and the decoders as:

$$D_{KL}(q_\eta(Z|X, A)\|p_\theta(Z|X, A)) \geq 0. \quad (18)$$

259 More explicitly, we have:

$$\int_{\mathbb{R}^{n \times d}} q_\eta(Z|X, A) \log \frac{q_\eta(Z|X, A)}{p_\theta(Z|X, A)} dZ \geq 0. \quad (19)$$

260 Using Bayes theorem, we can write Eq. (19) as:

$$\int_{\mathbb{R}^{n \times d}} q_\eta(Z|X, A) \log \frac{q_\eta(Z|X, A)p_\theta(X, A)}{p(Z)p_\theta(X, A|Z)} dZ \geq 0. \quad (20)$$

261 which develops into:

$$\begin{aligned} & \int_{\mathbb{R}^{n \times d}} q_\eta(Z|X, A) \log \frac{q_\eta(Z|X, A)}{p(Z)} dZ \\ & - \int_{\mathbb{R}^{n \times d}} q_\eta(Z|X, A) \log p_\theta(X, A|Z) dZ \geq -\log p_\theta(X, A). \end{aligned} \quad (21)$$

262 In other word, we have the inequality:

$$\log p_\theta(X, A) \geq \mathbb{E}_{Z \sim q_\eta(Z|X, A)} \log p(X, A|Z) - D_{KL}(q_\eta(Z|X, A) \| p(Z)). \quad (22)$$

263 The ELBO (lower bound of Eq. (22)) takes finally the following explicit form
264 (similarly as in Eq. (10)):

$$\begin{aligned} \mathcal{L}_{ELBO} &= \sum_{i=1}^n \sum_{j=1}^n \mathbb{E}_{\substack{z \sim q_\eta(z|\mathcal{V}^l(X^i), A) \\ z' \sim q_\eta(z'|\mathcal{V}^l(X^j), A)}} \log p_g(A_{i,j}|z, z') \\ &+ \sum_{i=1}^n \sum_{m=1}^M \mathbb{E}_{z \sim q_\eta(z|\mathcal{V}^l(X^i), A)} \log p_{\theta_m}(X_m^i|z) \\ &- \sum_{i=1}^n D_{KL}(q_\eta(z|\mathcal{V}^l(X^i), A) \| p(z)). \end{aligned} \quad (23)$$

265 In this formula, we obtain a graph reconstruction term, an explicit data
266 reconstruction term, and a regularizer in the latent space. All these terms
267 involve the parametric view decoders q_η that we define next.

268 3.3. Parametric decoders

269 We choose in this model to parametrize each q_{η_m} by a multivariate Gaus-
270 sian distribution with a **Krylov** Graph convolutional neural networks⁸ (GCN)
271 [27] as:

⁸log in log σ_m is applied element-wise on matrix σ_m .

$$\begin{aligned}
q_{\eta_m}(z|\mathcal{V}^l(X_m^i), A) &= \mathcal{N}(\mu_m^{\text{enc}}(:, i), \text{diag}(\sigma_m^{\text{enc}}(:, i))). \\
\mu_m^{\text{enc}} &= \mathbf{W}_m^{\mu \text{ enc}} \mathbf{Krylov}_m(X_m, A). \\
\log \sigma_m^{\text{enc}} &= \mathbf{W}_m^{\sigma \text{ enc}} \mathbf{Krylov}_m(X_m, A).
\end{aligned} \tag{24}$$

272 where μ_m^{enc} and $\log \sigma_m^{\text{enc}}$ are matrix output by Krylov layer whose i -th column
273 is, respectively, the mean and the diagonal of the covariance matrix of q_{η_m} .
274 We use the log to ensure positivity of sigma.

275 The reason for the choice of a GCN is that they are efficient to extract fea-
276 ture information of a node considering its neighborhood [28, 29]. They have
277 been widely used in node classification, node clustering, and other graph an-
278 alytic tasks. Currently, many GCN have similar performance [30], and since
279 the choice is not critical, we simply use a truncated Krylov GCN architec-
280 ture [27] which has been proven to have good properties when stacking across
281 graph layers.

282 3.4. Parameters of the model and inference

283 Globally, the trainable parameters (i.e., the ones to infer) of the models are
284 the weights Ψ_m , $\mathbf{W}_m^{\mu \text{ dec}}$, $\mathbf{W}_m^{\sigma \text{ enc}}$, $\mathbf{W}_m^{\mu \text{ enc}}$, respectively the parameters of the
285 multilayer perceptron MLP_m and of the Krylov GCN layers \mathbf{Krylov}_m .

286 In order to infer these parameters efficiently, we will use the ELBO of
287 Eq. (23). Even if this ELBO is not separable in a sum of terms depending
288 only of one instance i , it can be decomposed in terms containing only subsets
289 of some instances i . This leads to a training strategy of the model in a batch
290 manner, using a suitable optimization method. For every instance i , each
291 view m (i.e., X_m^i) has it's own latent representation, which is computed sim-
292 ilar to Eq. (9) i.e $\mathbb{E}_{z \sim q_{\eta_m}(z|\mathcal{V}^l(X_m^i), A)} = \mu_m^{\text{enc}}(:, i)$. These latent representations
293 of views allow us to build the common latent representation of instance i as
294 (see Fig. (2) for illustration):

$$\mathbb{E}_{z \sim q_{\eta}(z|\mathcal{V}^l(X^i), A)} = \left[\sum_{m=1}^M \frac{\mu_m^{\text{enc}}(k, i)}{\sigma_m^{\text{enc}2}(k, i)} / \sum_{m=1}^M \frac{1}{\sigma_m^{\text{enc}2}(k, i)} \right]_{k=1}^d. \tag{25}$$

295 These representations are used for the experiments presented later on.

296 *3.5. Robustness and “views dropout”*

297 The formula (25) comes from the choice of q_η as a product of multivari-
 298 ate Gaussian distribution with diagonal covariance matrix. This particular
 299 choice gives the model robustness property to deal with missing views. In-
 300 deed by model assumptions, each view of an instance i of the dataset is
 301 derived from a latent variable we seek for. For each view, the distribution
 302 $q_{\eta_m}(z|\mathcal{V}^l(X_m^i), A)$ gives the probability that the latent variable z generates the
 303 corresponding views. Thus we have $q_\eta(z|\mathcal{V}^l(X^i), A) = \prod_{m=1}^M q_{\eta_m}(z|\mathcal{V}^l(X_m^i), A)$
 304 which is the probability that z generates (all the views of) an instance i ; hence
 305 the probability of an instance is given by the product of probability mass of
 306 the different views of this instance i . Thus, the more probability mass the
 307 different views m gives to z , the more likely z are the sources of i (i.e
 308 of these views). But these views do not contribute equally to the probability.
 309 Some views or groups of views contains more information than others about
 310 z , as can be seen in Eq. (25) by the fact that z is the barycenter (using the
 311 precision (inverse variance) as weight) of view’s most probable latent space
 312 vector (i.e the mean of decoders).

313 In the case in which some views are missing, if the sum of precisions
 314 associated to missing views is not large compared to the one associated to
 315 existing views, the computation of Eq. (25) without these views will be still an
 316 accurate approximation of z . So given enough views, we could still compute
 317 a good approximation of the common latent representation. This potential
 318 ability to deal with missing views is successfully confirmed in the experiments
 319 reported under Section 5.3.2 and 5.3.3. There is no such direct possibility
 320 with other variational CCA approaches. Indeed in the 2-views variational
 321 model VCCA [20], q_η is parametrized only by one of the two views, hence
 322 this cannot be robust to the absence of the corresponding view for some
 323 instances. The improved model VCCA-private in the same article also suffers
 324 from this. The more recent variational model VPCCA [21] can deal with
 325 limited numbers of views available, but needs to have these views available
 326 for each instance to compute the low dimensional common latent space which
 327 makes it obviously not robust to real missing views situations where different
 328 instances could have different missing views (see Fig. (3)). The present model
 329 exploits the maximum amount of information available for each instance.

330 In order to reinforce the robustness of the proposed model, we introduce
 331 for the experiments the novel notion of “views dropout”. This consists of
 332 randomly ignoring certain views during the encoding part of the training
 333 phase while still asking the model to be able regenerate all the views as well

334 as the adjacency matrix. This procedure is more detailed and evaluated in
 335 experiment part 5.3.2 and 5.3.3.

336 As a summary of this methodological part, we repeat that our model is
 337 the only CCA model which is scalable, graph aware, and robust to missing
 338 views.

339 **Remark:** The choice for all q_{η_m} involves that q_{η} (cf. Eq. (17)) is not a
 340 correctly normalized distribution. However, q_{η} is proportional to a distribu-
 341 tion (there is a constant K such that Kq_{η} is a distribution). If we resume the
 342 calculation started in Eq. (18) with Kq_{η} it will lead to the same *ELBO* up
 343 to an additive and a multiplicative constant that does not change inference,
 344 so we can work as if q_{η} was a distribution.

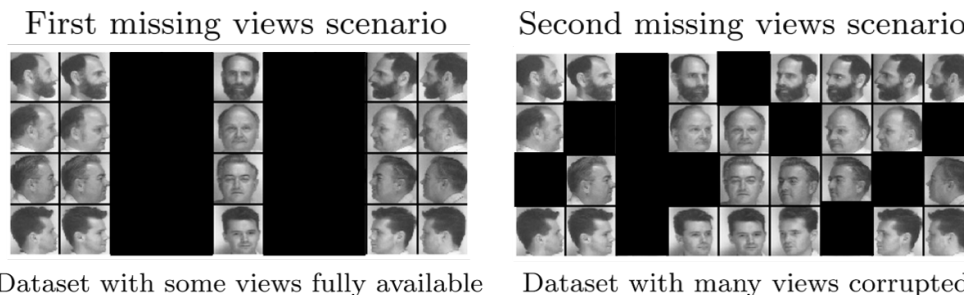


Figure 3: **Multiview dataset with missing views.** On the left, one can see a multiview dataset where some views are missing but some are fully available. Variational model such as VCCA (limited to two-view dataset) and VPCCA can deal with these kind of data. We evaluate MVGCCA in this scenario in Section 5.3.2. On the right one can see a dataset where many views are corrupted. This dataset is a much more realistic representation of errors which can be found on real dataset. The proposed model can deal with these kinds of missing data during both training and testing. MVGCCA exploits the maximum amount of information available for each instance. MVGCCA is evaluated in this scenario in Section 5.3.3.

345 4. Datasets

346 4.1. UCI handwritten digits dataset

347 UCI Handwritten Digits Dataset⁹ is a multiview dataset of $n = 2000$
 348 samples images representing digits. Each image has a label from 0 to 9
 349 (200 instances for each), and 6 views with different dimensions: $d_1 = 76$,

⁹archive.ics.uci.edu/ml/datasets/Multiple+Features

Dataset	uci7			uci10			Recommendation		
Metric	Acc.	ARI	ARI2	Acc.	ARI	ARI2	Prec.	Recall	Mrr
PCA	0.84	0.55	-	0.69	0.42	-	0.1511	0.0795	0.3450
GPCA	0.93	0.71	0.77	0.87	0.63	0.62	0.1578	0.0831	0.3649
MCCA	0.86	0.66	-	0.76	0.59	-	0.0815	0.0429	0.2225
GMCCA	0.95	0.83	0.84	0.90	0.69	0.71	0.2290	0.1206	0.4471
MVGCCA	0.95	0.82	0.85	0.94	0.74	0.77	0.1753	0.0583	0.4432
Dataset							Recommendation Large		
Metric							Prec.	Recall	Mrr
MVGCCA							0.1745	0.0960	0.4301

Table 2: Results of experiments on the different datasets and tasks; see text for the detailed discussion. Acc. stands for accuracy in classification; ARI for adjusted rank index in clustering tasks: ARI1 if using K-means and ARI2 if using spectral clustering. For the Recommendation task, Prec. is precision and Mrr is the mean reciprocal rank.

350 $d_2 = 216$, $d_3 = 64$, $d_4 = 240$, $d_5 = 47$ and $d_6 = 6$. These views correspond
351 to specific transformations of the original image: Fourier coefficients of the
352 character shapes $X_1 \in \mathbb{R}^{d_1 \times n}$; profile correlations $X_2 \in \mathbb{R}^{d_2 \times n}$; Karhunen-
353 Loeve coefficients $X_3 \in \mathbb{R}^{d_3 \times n}$; 240 pixel averages in 2 x 3 windows $X_4 \in$
354 $\mathbb{R}^{d_4 \times n}$; Zernike moments $X_5 \in \mathbb{R}^{d_5 \times n}$; and 6 morphological features $X_6 \in$
355 $\mathbb{R}^{d_6 \times n}$. Clustering, classification and reconstruction tasks are performed on
356 this whole dataset (uci10); also, we consider as in [23] for comparison, a
357 partial version (uci7) where classes 0, 5 and 6 have been removed. The
358 obtention of a prior graph is an important step for the analysis. As the
359 objective is to be able to compare the method to existing one, like [23], the
360 choice is made to follow exactly the method described in Chen et al. [23]
361 (Section VII.C) to build the prior graph over data.

362 4.2. Twitter friend recommendation

363 A multiview dataset¹⁰ based on post from Twitter has been proposed
364 in [6]. It consists of multiview representations of messages of users. They
365 took 1% of the publicly available users data in April 2015. They removed all
366 tweets that are not in english, and those from users who did not post between
367 January and February 2015¹¹. Finally they only kept the last 100 tweets from
368 the remaining users, yielding $n = 102327$ users. These data for each user

¹⁰<http://www.cs.jhu.edu/~mdredze/data/>

¹¹There are other minor exclusion criteria which may be found in [6].



Figure 4: **Illustration of two-view MNIST dataset.** Three samples of two-view MNIST dataset. For each digit (1, 5 and 8) we display on the left : original image, middle: first (rotated) view, right: second (noisy) view.

369 are in the form of 6 1000-dimensional views: EgoTweets, MentionTweets,
 370 FriendTweets, FollowersTweets, FriendNetwork, and FollowerNetwork. A
 371 task of friend recommendation is performed as follows. The followed accounts
 372 are known for each user; given a highly followed account and a part of their
 373 followers, the goal is to determine, for each other users, whether or not he
 374 will follow this account after March 2015. For this task, a graph based on the
 375 Twitter dataset is built as in [23] with the views Egoweets, FollowersTweets,
 376 and FriendNetwork.

377 4.3. Two-view MNIST noisy dataset

378 Two-view MNIST noisy dataset has been proposed by Wang et al. [15].
 379 It is a two-view dataset built from the famous MNIST handwritten digits
 380 28×28 dataset. MNIST dataset is composed of a training set with 50000
 381 instances, a validation set with 10000 instances and a test set with 10000
 382 instances. We merge them. The first view is obtained after performing
 383 a rotation to the 70000 images. Angles of each rotation have been sampled
 384 from continuous uniform distribution $\mathcal{U}([-π/4, π/4])$ with $-π/4$ as minimum
 385 value and $π/4$ as maximum value. For the second view, we choose randomly
 386 another image with identical labels from MNIST and we add a noise sampled
 387 from uniform distribution $\mathcal{U}([0, 1])$ to each pixel. Finally, we truncate pixels
 388 values to keep them between 0 and 1. This data has no pre-defined graph
 389 structure. We build the prior graph by connecting each instance with a
 390 probability 1/10 (resp. 1/1000) to another instances with the same labels
 391 (resp. different labels). At the end each instance is connected to nearly 10%
 392 of instance with a different label. Hence this graph structure gives some
 393 (noisy) supplementary information about relation between instances of this
 394 dataset. Finally, to evaluate our model we split the dataset in a training
 395 set, a validation set and a test set of the same size as at the beginning. A
 396 visualisation of this dataset can be seen on Figure 4.

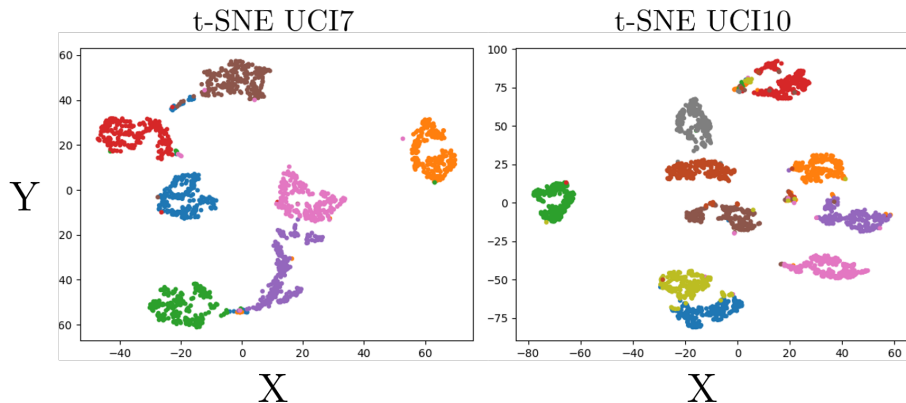


Figure 5: t-SNE visualisation in 2D of the latent space ($d = 3$) for the uci7 and uci10 dataset. Each color represents a different class.

397 5. Experiments

398 All architectures and hyperparameters specified here are the same for all
 399 experiments¹², unless otherwise indicated in the text. We pre-process all the
 400 views: each view is centered and normalized by standard deviation. For each
 401 dataset, the graph adjacency matrix is rescaled with its maximal entry and
 402 diagonal coefficients are set to 1.

403 **Decoders:** Mean decoders computed by MLP_m for each view m are used
 404 with a ReLU non linearity except the last layer which is a linear transforma-
 405 tion. Covariance matrix decoders Ψ_m are trained as a scaled identity matrix:
 406 $\Psi_m = ((\sigma_m + 10^{-6})^2)I_{d_m}$. This choice reduces the complexity and improves
 407 robustness, and we have seen in experiments that it does not decrease per-
 408 formance.

409 **Encoders:** Krylov GCN layers [27] encoding the mean and variance in q_η ,
 410 have a depth of $l = 4$ hop neighborhood.

411 **General features:** Batch size is set to 512. A dropout regularization of
 412 rate 0.5 is also applied after all the hidden layers. The Adam optimizer is
 413 used for training. A decay learning rate is applied: $l_r 1.1^{-50 \frac{e}{E}}$, where e is the
 414 current epoch and E the maximal number of that is set to 600 epochs by
 415 default.

¹²Code available : <https://github.com/Yacnmm/MVGCCA>

416 *5.1. Classification and clustering*

417 We performed classification and clustering experiments. A first experi-
418 ment involves a comparison to GMCCA (which is the only method which
419 can deal with graph). And a second one shows comparisons to many other
420 non linear method such as Deep CCA, Kernel CCA, Variational CCA, etc
421 (cf Tab. (3)).

422 *5.1.1. UCI*

423 The model was trained on uci7 and uci10 datasets. The loss function
424 was the ELBO from Eq. (23), trained with a batch of dataset. For each
425 batch, the graph used is the subgraph of samples in the batch. The latent
426 space is of dimension $d = 3$. We perform a grid search over the learning
427 rate (1e-3,1e-4), number of hidden layers (3,4) and hidden layers size (512,
428 1024). We also decide to apply a decay learning rate or not, and to look for
429 decoders covariance matrix as full matrix or scalar matrix based on a grid
430 search. For each combination of these parameters we trained the algorithm
431 three times, on 600 epochs and we save each latent embedding every 100
432 epochs (to perform early stopping).

433 We take 90% instance of dataset as the train set and the remaining 10%
434 as test set. We perform a 5-fold cross validation with SVM-RBF accuracy
435 to find optimal parameters (MVGCCA and SVM-RBF hyperparameters and
436 early stopping step) on the train set. Finally, for the best parameters, we
437 train the SVM-RBF on the train set and evaluate it on test set. We also
438 perform a K-means and spectral clustering on the whole embeddings (the
439 train and test sets). A 2D t-SNE projection of latent space can be visualized
440 in Figure 2.

441 Because of the small size of UCI dataset, the ouput of this procedure de-
442 pends on initial choice of dataset splitting; so in order to overcome this issue,
443 we perform this experiment 100 times and average the results. Following
444 [23], results are compared to PCA (applied on concatenated views), graph-
445 regularized PCA, MCCA, and GMCCA. In order to make fair comparison in
446 terms of hyperparameter tuning, all the experiments were done with the same
447 protocol for all the methods.

448 The results are summarized in Table 2. We see that MVGCCA is competi-
449 tive in both classification (Acc.) and clustering tasks (ARI if using K-means
450 & ARI2 if using spectral clustering). It achieves the best performance on the
451 more complex dataset uci10 over other graph aware methods. This indicates
452 that the graph structure is well encoded in the latent space.

453 **Remark:** In using SVM-RBF, we have conducted a grid search (with sklearn
454 package) for the following parameters: \mathbf{C}^{13} : 7 values logarithmically spaced
455 in $[10^{-3}, 10^3]$ and **gamma**: 7 values logarithmically spaced in $[10^{-3}, 10^3]$. The
456 K-means and the spectral clustering (with **gamma** = 5) are performed with
457 the default parameters of sklearn.

458 5.1.2. Two-view MNIST

459 This dataset is composed of a training, validation and test sets. For this
460 experiment we train our model only on the training set. We performed a
461 small grid search based on insight given by previous experiments; hence we
462 only tune the latent space dimension $d = \{30, 60\}$ and if we use a scalar
463 decoder or not. We used a 4 hidden layers and the hidden layers size is set to
464 1024 for both encoders and decoders. We trained the algorithm during 200
465 epochs and we save each latent embedding every 10 epochs after 50 epochs to
466 perform early stopping. Once the model has been trained, the latent space
467 of train, validation and test set is then inferred from the model. In order
468 to perform fair comparison with others variational method VCCA [20] and
469 VPCCA [20], we suppose that only the first view is available to produce
470 the test set latent space (first scenario of Figure 3), it is equivalent to use
471 Z_1 as latent representation. We use these representations for classification.
472 We use VPCCA [21] and [15] as the baseline. We train a Linear SVM with
473 regularization parameter \mathbf{C} having possibly 8 values that are logarithmically
474 spaced in $[10^{-3}, 10^3]$. We used the validation set to select best parameter for
475 this linear SVM, the dimension of our latent space, and to decide to stop
476 early the epochs. Finally, we trained the linear SVM on both training and
477 validation set with these parameters and evaluate our method on the test
478 set. On contrary to the UCI dataset, two-view MNIST is a large dataset,
479 so the results of these experiments have been averaged only over three runs.
480 The results of these experiments are summarized in Table 3.

481 As it can be seen, the results are competitive against the best methods of
482 the state of art, thanks to the information brought by the graph structure.
483 This illustrates again how CCA can benefit from taking into account some
484 geometrical structure in the multiview dataset.

Dataset	Two-view MNIST
Metric	Acc.
Linear CCA	0.804
SpliAE	0.881
Kernel CCA	0.949
Deep CCA	0.971
DCCAE	0.978
VCCA	0.970
VCCA-(p)	0.976
VPCCA	0.981
MVGCA	0.985

Table 3: **Results of classification with a linear SVM on two-view MNIST**; see text for the detailed discussion. The baseline is issue from Wang et.al [15] and Karami et al.[21]. The baseline involve linear CCA [8]; non linear method: Kernel CCA [17], Deep CCA [13], Deep CCA autoencoders [15], and probabilistic model Variational CCA (-private) [20] or Variational Probabilistic model [21]. For VCCA, VPCCA and MVGCCA, all views are available during training time, while only the first view is available for inference. MVGCCA and VPCCA are the only model which can deal with more than two views.

485 *5.2. Twitter friend recommendation*

486 For this dataset and the recommendation task, no further hyperparameter
487 tuning is done and values from previous experiments are used with a learning
488 rate = 10^{-4} , number of hidden layers = 4, hidden layers size = 1024, number
489 of epochs = 600, a decay learning rate and a scalar covariance matrix for
490 decoders. The parameters of the methods used for comparison are extracted
491 from [16, 23] with their best parameters. The twitter dataset is large with
492 more than 100,000 users, which makes it intractable for existing methods.
493 Hence, 2506 twitter users are randomly selected from the database as in [23]
494 for fair comparison.

495 The 20 most followed accounts (over the whole dataset) are selected,
496 For each of these, 10 users following them are chosen (at random) and the
497 average representation from latent space is computed. These average profiles
498 will represent the typical users who follow the 20 most followed accounts.
499 The latent space is set to dimension $d = 5$. Finally, the cosine similarity is
500 computed between this average profile and the one of the $L = 100$ closest
501 users to these representations. If one of these 100 users actually followed the

¹³Regularizer parameters of Linear SVM. See sklearn implementation of Linear SVM.

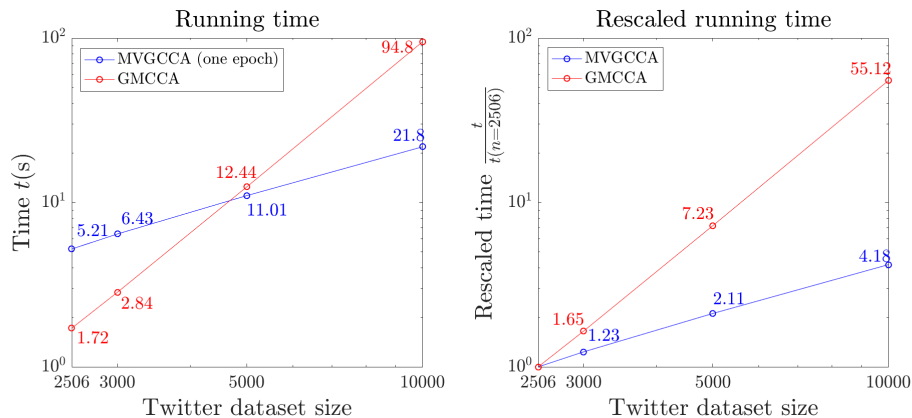


Figure 6: Log-log plot of running time between the two graph CCA methods: MVGCCA and GMCCA. We measure the running time for a run of GMCCA and one epoch of MVGCCA for four twitter dataset size ($n = 2506, 3000, 5000, 10000$). On the right figure we rescaled the running time by the running time at $n = 2506$. One can see that the GMCCA running time grows much more quickly than for MVGCCA.

502 initially chose account, this is considered as a good friend’s recommendation.
 503 To assess performance, precision, recall, and Mrr (mean reciprocal rank)
 504 metrics are computed (averaged for the 20 most followed accounts). This
 505 experiment has been repeated and averaged over 100 sampling of 2506 users.
 506 The results are in Table 2 (right).

507 The performance of MVGCCA is comparable (except for recall) to that
 508 of GMCCA, which is currently the best method for this task. We recall here
 509 that the results are assessed on a limited dataset that all methods can process
 510 while our model could process a larger sample of dataset. To check that, we
 511 re-run the same experiment on larger dataset in order to show the scaling
 512 of the running time for a dataset from $n = 2506$ to $n = 10000$. The result
 513 is displayed in Fig. 6. We observe that the scaling is, as expected, better
 514 for MVGCCA and, for data of size $n = 5000$ of larger, MVGCCA epochs
 515 are quicker despite the overhead in computing time due to the use of GCN.
 516 Considering that after 100 epochs we already have goods results.

517 In order to give insight of what happens when we deal with a larger
 518 dataset, we also report the result of the recommendation task for the largest
 519 scale experiment, i.e with dataset 5 times larger. Here, we consider $n = 12530$
 520 users and the 100 most followed accounts. We choose 50 users following
 521 these most followed accounts to compute average representation. The same

522 training parameters are used but latent space dimension which is set to $d =$
523 10. For evaluation, we take $L = 500$ in order to keep the same difficulty (we
524 have 5 times the number of users) so as to make this experiment comparable
525 to the previous one. A graph based on the Twitter dataset is built as before,
526 from the exact same views Egoweets, FollowersTweets, and FriendNetwork,
527 but with the neighbor parameters equal to 100 (instead of 50 before). Please
528 refer to [23] to have more information about this parameter. The experiments
529 have been averaged over 20 sampling of 12530 users. The result of this larger
530 scale experiment is in Table 2 (last line). As it can be seen, the performance
531 is quite the same on precision and Mrr metrics while there is an improvement
532 for the recall metrics. It means that, thanks to the larger amount of data, the
533 algorithm is capable of finding more of what it is possible to be find, while
534 conserving a good precision. This results in a greater ability to perform
535 recommendation.

536 **Remark:** We did not process a larger experiment because of the limita-
537 tions associated to graph building procedure. On a data coming naturally
538 with a graph structure, this problem would not exist; we choose here this
539 dataset for the sake of comparison on a situation studied in the literature.

540 5.3. Inference

541 In a final experiment, we illustrate the robustness of the model to missing
542 views for some instances, and its ability to reconstruct these missing views.
543 This property of the model is unique among other graph aware CCA methods.
544 To do so, we rely on the probabilistic nature of our model. Once the model
545 has been trained, for a new instance not seen during training we apply the
546 formula of Eq. (25) restricting the sum to the available views. Then this
547 approximation of the latent representation z^{approx} is used to regenerate any
548 missing views m as:

$$x_m^{regenerate} = \mathbb{E}_{x \sim p_{\theta_m}}(x | z^{approx}) = \mathbf{W}_m^{\mu \text{ dec}} \mathbf{MLP}_m(z^{approx}). \quad (26)$$

549 5.3.1. Recovering missing views

550 We consider the following scenario: We train the model on a training set
551 (90% of data) where all the views are available for all instances. Then, for
552 every $k \in [0, 4]$, we randomly select 10 instances in a test set¹⁴ for which we

¹⁴Whose elements have not been seen during training.

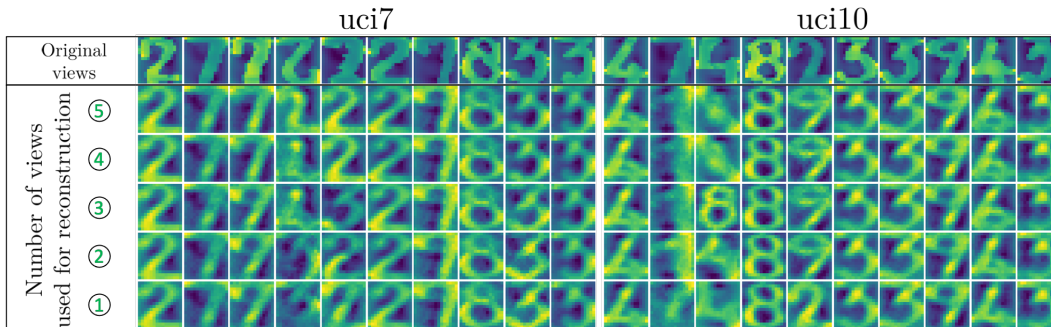


Figure 7: **UCI 4th view reconstruction.** We see here the result of the reconstruction experiment of the 4th view. The first line corresponds to the original view. Every figure in line $l > 2$ corresponds to a reconstruction from $7 - l$ views. We did not use views dropout in this experiment.

553 remove the 4th view and k additional views¹⁵. Note that, in this dataset, the
 554 4th view is a subsampling of the original image. Using the other views, we
 555 regenerate 4th views according to Eq. (25) restricted to available views.

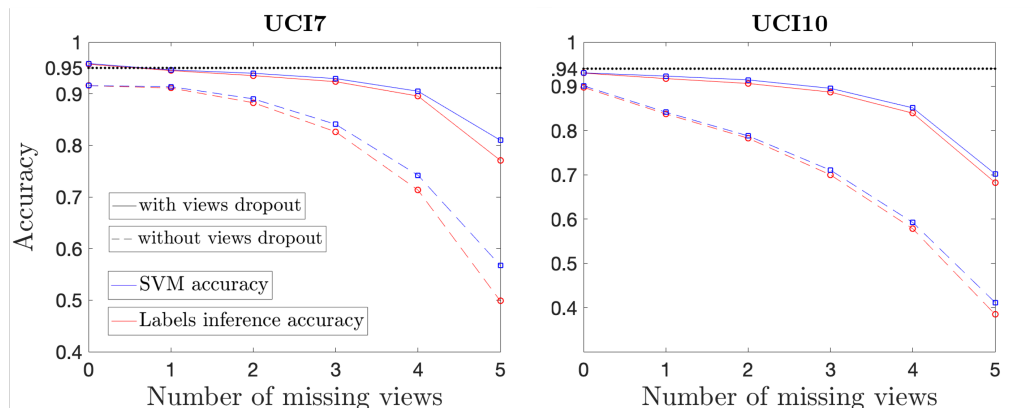
556 The results are given in Fig. 7. We see that in the case of uci7 the
 557 reconstruction goes very well in the majority of cases whatever the number
 558 of missing views. Still, there are some problematic cases. For instance, in
 559 column 4 we can clearly see that the original view is upside down, while in
 560 column 5 the original view is degraded – this shows that in both cases we have
 561 an atypical point of the dataset. Also, in column 7 we see some confusion
 562 between a 7 and a 9. The right part of the Figure is for uci10 which is a
 563 more difficult dataset than uci7. The method has additional difficulties to
 564 regenerate some views, because of some confusions between 4 and 6, and
 565 between 7 and 1. Still, this difference between uci7 and uci10 suggests that
 566 a training on a larger database should solve these problems. Anyway, we see
 567 that the majority of the proposed reconstructions from the model are very
 568 close to original views, thus demonstrating experimentally the ability of our
 569 model to reconstruct missing views.

570 5.3.2. Robustness wrt. missing views – first scenario

571 This scenario corresponds to the first of Figure 3. In this experiment the
 572 quality of the reconstruction is evaluated more quantitatively and this will

¹⁵The k views removed are the k views with the smallest dimension.

a. First scenario



b. Second scenario

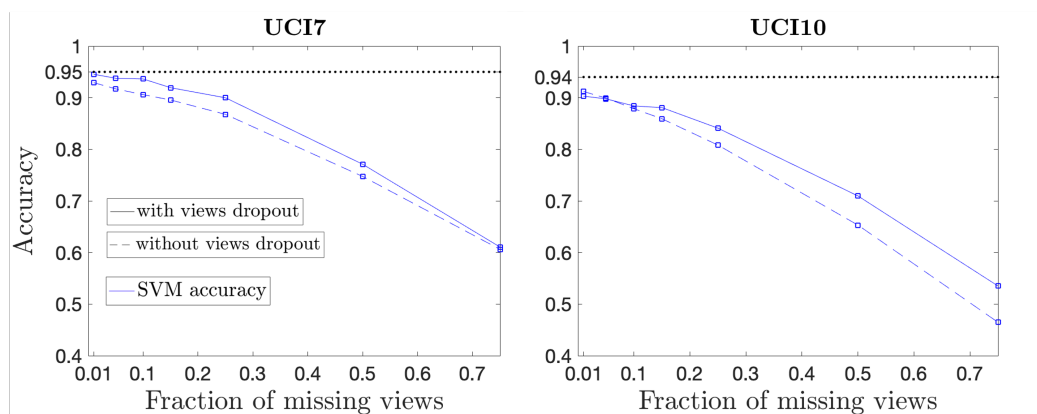


Figure 8: **Illustration of MVGCCA robustness.** See section 5.3.2 and 5.3.3 and Fig. 3 for details on both scenario. **a. First scenario:** The accuracy of clustering is compared for the two procedures of test (by SVM-RBF or inference of 1-hot encoding), without and with “views dropout”. The black dotted line corresponds to accuracy found in Section 5.1.1 with all views. Note that we did not search for optimal parameters here, so it is expected to find a lower accuracy for 0 missing views. Still, for uci7, we obtain a better accuracy when using “views dropout”. **b. Second scenario:** We evaluated on the SVM-RBF accuracy in this scenario. One can see similar behavior. The more the fraction of missing views, the more the performance decrease. The procedure of “views dropout” helps somehow the algorithm to deal with missing views, with a diminished the effect in comparison to the first scenario. Globally, this scenario seems to be harder, yet MVGCCA is currently the only method to deal with it.

573 show the robustness of the proposed model when only a subset of views are
574 (fully) available. To do so, we work with regular UCI and an extended UCI
575 where we add to the training set a 7th view that corresponds to a one hot
576 encoding of the instance label. Then, we train MVGCCA on this two sets.
577 Finally, we evaluate accuracy on test set in two manners:

- 578 • For regular UCI, we train a SVM-RBF on train set embeddings from
579 trained model. Then we evaluate this SVM on test set embeddings
580 where some views have been removed. These embedding are computed
581 as in section 5.3.1.
- 582 • For extended UCI, we regenerate the 7th views of test set instances.
583 This directly gives us an estimation of their labels. Once again, some
584 views are removed.

585 For a given number of available views $v \in [1, 5]$, we consider any possi-
586 ble combination of views to form novel datasets, and we average the ob-
587 tained accuracies (e.g if $v = 2$ the set of available views considered are
588 $\{(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), \dots, (5, 6)\}$). It allows us to obtain a
589 result that does not depend on the nature of the missing views. Moreover,
590 we repeat this experiment 10 times and average the results. They are pro-
591 vided in Figure 8.a as dashed line. As we can see, the model can deal with
592 a small number of missing views; still the performance decreases with the
593 number of these missing views and that is not surprising.

594 In the same figure, we show results with continuous line where we robus-
595 tify the method by applying a “*views dropout*” during training phase. This
596 consists in randomly removing between 1 and 5 views to each instance of
597 the batch size for the encoding part of the model only. Hence the model is
598 trained in order recover all views from only a subset of views. As we can
599 see this procedure makes the model very robust: for instance, when 4 views
600 are missing, the accuracy with *views dropout* remain better than using the
601 model without *views dropout*, even with a smaller number of missing views.
602 This allows even for better results to be obtained when no view is missing.

603 5.3.3. Robustness wrt. missing views – second scenario

604 This scenario corresponds to the second of Figure 3. In the previous
605 experiment, limited subsets of available views were available for generating
606 the latent space but all instances had the same set of available views. In
607 real scenario different instances may have different missing views. We will

608 evaluate MVGCCA in this scenario. Once again we will train MVGCCA
 609 on train set (90% of the dataset) and then infer the test set latent space
 610 after having randomly removed some views. We removed a percentage $r =$
 611 $\{1\%, 5\%, 10\%, 15\%, 25\%, 50\%, 75\%\}$ of views on the test sets while asserting
 612 that each instance will have at least one views available. We repeat the
 613 experiment 10 times and for each we evaluated 10 times each percentage of
 614 missing views r with and without views dropout.

615 As can be seen in Figure 8.b, we have a similar behavior than for previous
 616 experiments. The method is able to deal with this type of corrupted dataset.
 617 This is the first method to do so in the literature. The procedure of “views
 618 dropout” is again useful somehow (this time by removing between 1 and 3
 619 views to each instance), but its effect is diminished as compared to scenario
 620 1.

621 **Remark:** As discussed before, in order to compute a low dimensional repre-
 622 sentation when some views are missing, we restrict the formula of Eq. (25)
 623 to available views. In this second scenario, we need to make some adjuste-
 624 ment in order to compute $\mu_m^{\text{enc}}(:, i)$ and $\sigma_m^{\text{enc}}(:, i)$ from Eq. (25), as they de-
 625 pend on the views m of neighbours of i (see Eq. (24)); these views could be
 626 missing. We overcome this issue by replacing in $q_{\eta_m}(z|\mathcal{V}^l(X_m^i), A)$ the ad-
 627 jacency matrix A by A_m which is the adjacency matrix where unavailable
 628 views have been suppressed. Thus $\mathcal{V}^l(X_m^i)$ contains only available views m
 629 in the neighborhood of instance i .

630 6. Conclusion

631 We proposed MVGCCA, a novel multiview and non linear extension of
 632 CCA based on a Bayesian inference model. The proposed model is scalable,
 633 and can take into account the available graph structural information from
 634 the data. We have also proposed also a robustification method to handle
 635 missing data by applying “*views dropout*” during training. The probabilistic
 636 graphical nature of the model can be used for other tasks, such as addressing
 637 link prediction for multiview datasets, and that is a perspective for future
 638 work.

639 References

- 640 [1] M. E. Quemener, ”SIDUS”, the solution for extreme deduplication of
 641 an operating system, The Linux Journal (2014).

- 642 [2] Y. Kaloga, P. Borgnat, S. P. Chepuri, P. Abry, A. Habrard, Multiview
643 variational graph autoencoders for canonical correlation analysis, in:
644 2021 IEEE International Conference on Acoustics, Speech and Signal
645 Processing (ICASSP), 2021. [arXiv:2010.16132](https://arxiv.org/abs/2010.16132).
- 646 [3] Y. Yamanishi, J.-P. Vert, A. Nakaya, M. Kanehisa, Extraction of cor-
647 related gene clusters from multiple genomic data by generalized kernel
648 canonical correlation analysis, *Bioinformatics* 19 (2003) i323–i330.
- 649 [4] R. Arora, K. Livescu, Multi-view cca-based acoustic features for pho-
650 netic recognition across speakers and domains, in: 2013 IEEE Interna-
651 tional Conference on Acoustics, Speech and Signal Processing, 2013, pp.
652 7135–7139.
- 653 [5] R. T. Collins, A. J. Lipton, H. Fujiyoshi, T. Kanade, Algorithms for
654 cooperative multisensor surveillance, *Proceedings of the IEEE* 89 (2001)
655 1456–1477.
- 656 [6] A. Benton, R. Arora, M. Dredze, Learning multiview embeddings of
657 twitter users, in: *Proc. Annual Meeting Assoc. Comput. Linguistics*,
658 volume 2, 2016.
- 659 [7] Y. Li, M. Yang, Z. Zhang, A survey of multi-view representation learn-
660 ing, *IEEE Transactions on Knowledge and Data Engineering* 31 (2019)
661 1863–1883.
- 662 [8] H. Hotelling, Relations between two sets of variates, *Biometrika* 28
663 (1936) 321–377.
- 664 [9] J. Kettenring, Canonical analysis of several sets of variables, *Biometrika*
665 58 (1971) 433–451.
- 666 [10] K. Chaudhuri, S. Kakade, K. Livescu, K. Sridharan, Multi-view clus-
667 tering via canonical correlation analysis, in: *Proceedings of the 26th*
668 *Annual International Conference on Machine Learning, ICML '09, As-*
669 *sociation for Computing Machinery, 2009*, p. 129–136.
- 670 [11] S. Akaho, A kernel method for canonical correlation analysis, in: *In*
671 *Proceedings of the International Meeting of the Psychometric Society*
672 *(IMPS2001, Springer-Verlag, 2001*.

- 673 [12] L. P. Ling, C. Fyfe, Kernel and nonlinear canonical correlation analysis.,
674 International Journal of Neural Systems 10 (2000) 365–377.
- 675 [13] G. Andrew, R. Arora, J. Bilmes, K. Livescu, Deep canonical correla-
676 tion analysis, volume 28 of *Proceedings of Machine Learning Research*,
677 PMLR, Atlanta, Georgia, USA, 2013, pp. 1247–1255.
- 678 [14] A. Benton, H. Khayrallah, B. Gujral, D. A. Reisinger, S. Zhang,
679 R. Arora, Deep generalized canonical correlation analysis, in: Pro-
680 ceedings of the 4th Workshop on Representation Learning for NLP
681 (RepL4NLP-2019), Florence, Italy, 2019, pp. 1–6.
- 682 [15] W. Wang, R. Arora, K. Livescu, J. Bilmes, On deep multi-view rep-
683 resentation learning, volume 37 of *Proceedings of Machine Learning
684 Research*, PMLR, Lille, France, 2015, pp. 1083–1092.
- 685 [16] X. Chang, T. Xiang, T. M. Hospedales, Scalable and effective deep
686 CCA via soft decorrelation, in: 2018 IEEE Conference on Computer
687 Vision and Pattern Recognition, CVPR, IEEE Computer Society, 2018,
688 pp. 1488–1497.
- 689 [17] D. Lopez-Paz, S. Sra, A. Smola, Z. Ghahramani, B. Schölkopf, Random-
690 ized nonlinear component analysis, 2014.
- 691 [18] F. R. Bach, M. I. Jordan, A probabilistic interpretation of canonical cor-
692 relation analysis, Technical Report, Department of Statistics, University
693 of California, Berkeley, 2005.
- 694 [19] T. N. Kipf, M. Welling, Variational graph auto-encoders, arXiv:stat.ML
695 1611.07308 (2016). [arXiv:1611.07308](https://arxiv.org/abs/1611.07308).
- 696 [20] W. Wang, H. Lee, K. Livescu, Deep variational canonical correlation
697 analysis, ArXiv abs/1610.03454 (2016).
- 698 [21] M. Karami, D. Schuurmans, Variational inference for deep probabilistic
699 canonical correlation analysis, 2020. [arXiv:2003.04292](https://arxiv.org/abs/2003.04292).
- 700 [22] J. Chen, G. Wang, Y. Shen, G. B. Giannakis, Canonical correlation
701 analysis of datasets with a common source graph, IEEE Transactions
702 on Signal Processing 66 (2018) 4398–4408.

- 703 [23] J. Chen, G. Wang, G. B. Giannakis, Graph multiview canonical cor-
704 relation analysis, *IEEE Transactions on Signal Processing* 67 (2019)
705 2826–2838.
- 706 [24] T. D. Bie, B. D. Moor, K. Arenberg, On the regularization of canonical
707 correlation analysis, 2002.
- 708 [25] J. Rupnik, P. Skraba, J. Shawe-Taylor, S. Guettes, A comparison of
709 relaxations of multiset canonical correlation analysis and applications,
710 *CoRR* abs/1302.0974 (2013). [arXiv:1302.0974](https://arxiv.org/abs/1302.0974).
- 711 [26] D. Kingma, M. Welling, Auto-encoding variational bayes, in: 2nd In-
712 ternational Conference on Learning Representations, ICLR, 2014. URL:
713 <http://arxiv.org/abs/1312.6114>.
- 714 [27] S. Luan, M. Zhao, X.-W. Chang, D. Precup, Break the ceiling: Stronger
715 multi-scale deep graph convolutional networks, in: *Advances in Neural*
716 *Information Processing Systems* 32, 2019, pp. 10945–10955.
- 717 [28] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural net-
718 works on graphs with fast localized spectral filtering, in: *Advances in*
719 *Neural Information Processing Systems* 29, 2016, pp. 3844–3852.
- 720 [29] T. N. Kipf, M. Welling, Semi-supervised classification with graph con-
721 volutional networks, *CoRR* abs/1609.02907 (2016). [arXiv:1609.02907](https://arxiv.org/abs/1609.02907).
- 722 [30] F. Wu, T. Zhang, A. H. S. Jr., C. Fifty, T. Yu, K. Q. Weinberger,
723 Simplifying graph convolutional networks, volume 97 of *Proceedings of*
724 *Machine Learning Research*, 2019, pp. 6861–6871. [arXiv:1902.07153](https://arxiv.org/abs/1902.07153).