Multi-view hierarchical Variational AutoEncoders with Factor Analysis latent space

Alejandro Guerrero-López\textsuperscript{a,b,*}, Carlos Sevilla-Salcedo\textsuperscript{c}, Vanessa Gómez-Verdejo\textsuperscript{a}, Pablo M. Olmos\textsuperscript{a,b}

\textsuperscript{a}Department of Signal Processing and Communications, Universidad Carlos III de Madrid, Leganés, 28911, Spain
\textsuperscript{b}Instituto de Investigación Sanitaria Gregorio Marañón, Madrid, 28007, Spain
\textsuperscript{c}Department of Computer Science, Aalto University, Espoo, 02150, Finland

Abstract

Real-world databases are complex, they usually present redundancy and shared correlations between heterogeneous and multiple representations of the same data. Thus, exploiting and disentangling shared information between views is critical. For this purpose, recent studies often fuse all views into a shared nonlinear complex latent space but they lose the interpretability. To overcome this limitation, here we propose a novel method to combine multiple Variational AutoEncoders (VAE) architectures with a Factor Analysis latent space (FA-VAE). Concretely, we use a VAE to learn a private representation of each heterogeneous view in a continuous latent space. Then, we model the shared latent space by projecting every private variable to a low-dimensional latent space using a linear projection matrix. Thus, we create an interpretable hierarchical dependency between private and shared information. This way, the novel model is able to simultaneously: (i) learn from multiple heterogeneous views, (ii) obtain an interpretable hierarchical shared space, and, (iii) perform transfer learning between generative models.

Keywords: Multi-view, Heterogeneous, Interpretable, Factor Analysis, VAE, Hierarchical

2010 MSC: 00-01, 99-00

\textsuperscript{*}Corresponding author

Email address: alexjorguer@tsc.uc3m.es (Alejandro Guerrero-López)
1. Introduction

In recent years, deep generative models have demonstrated their ability to create heterogeneous and realistic synthetic data such as images, text or sound. Since the proposal of the first Variational AutoEncoder (VAE) [1], there have arisen numerous variations and enhancements for sophisticated issues such as time series modeling [2, 3], finding latent information using deep hierarchical structures [4, 5, 6] or even dealing with heterogeneous and missing data [7, 8].

Deep hierarchical VAEs [9, 10] divide the latent space into disjoint groups of latent variables to increase the expressiveness of both the approximate posterior and prior distributions. They stand out as powerful techniques capable of disentangling private and shared information into hierarchical latent variables. We can find diverse approaches for this purpose in the state-of-the-art (SOTA). For example, in NestedVAE [11], the authors propose to learn a private latent variable per view to, later, use them to feed a fully-connected layer and generate a shared latent variable. In DMVAE [12], the authors follow a similar strategy but they claim that it is better to construct the shared latent space using a Gaussian Product-of-Experts (PoE). Other works, such as GMVAE [13], incorporate unsupervised clustering to decouple the latent variables, or include Student’s-T prior to enhance robustness to outliers [14]. Moreover, recent studies such as UGVAE [7] or HVAEM [15], show that hierarchical structures are better at representing the information shared between views. However, training such deep generative models creates inference bias due to approximate inference techniques. Therefore, as it is pointed out in [15] and [16], time-costly techniques, such as Hamiltonian Monte Carlo sampling, are required. Although transfer learning could be used to speed up the deep VAEs training process and, to the best of our knowledge, there is no method in the SOTA able to perform this.

Many real-world problems require dealing with heterogeneous and mixed data-types (discrete, real, positive or continuous). Thus, generative models have to learn from multiple and heterogeneous inputs representing the same data.
In the literature, some studies face this issue with different techniques. For example, in JMVAE [17], the authors project two different views to a shared latent space in a bidirectional double VAE. Whereas, in JVAE² [18], they prefer to combine adversarial learning with the generative ability of the VAE’s latent features. Other works, such as MVAE [19], combine K input views by having an individual encoder per view but sharing parameters between them and, later, create a common latent space representation by a Gaussian PoE given the private representations. Following this idea, in AMVAE [20], the K views are projected into a common latent space but using a fully-connected layer. Yet, in Multi-VAE [21], the authors propose a probabilistic change, since each view is represented by a private continuous latent variable and the shared information is modeled by a categorical discrete latent variable by means of a Gumbel softmax [22] reparametrization. Some of these above mentioned models can also perform domain adaption between different datasets. For example, MVAE [19] performs machine translation between English and Vietnamese by training a VAE for each language. The Multi-VAE [21] is able to generate new samples of MNIST [23] by conditioning the images to their writer style. Moreover, UGVAE [7] performs domain image adaptation between the datasets CelebA [24] and 3D-Faces [25]. However, none of these approaches are able to obtain interpretable latent spaces, what hinders the analysis of the results beyond the prediction score.

As a solution to these limitations, we propose a novel method to learn an interpretable latent space by combining multi-view VAEs by a hierarchical factor analysis approach called Factor Analysis VAE (FA-VAE). We condition multiple VAEs to a factor analysis model called Sparse Semi-supervised Heterogeneous Interbattery Bayesian Analysis (SSHIBA) [26]. In particular, we first propose to use multiple VAEs to model the private information of every view using latent private variables and, secondly, project them to a shared low-dimensional space, generated by SSHIBA, by a linear transformation which is efficiently learned by mean-field Variational Inference (VI). This way, the proposed FA-VAE approach is able to:
• Model **multiple and heterogeneous views** using one VAE per view.

• Learn hierarchical latents by efficient mean-field VI.

• Obtain an explainable shared latent space which is linearly dependent with each private space.

• Use the hierarchical space to **transfer learning between VAEs**.

Therefore, our experiments demonstrate that our approach is able to (i) condition pretrained VAEs to arbitrary attributes, (ii) perform domain adaptation between different datasets by combining multiple VAEs, and, (iii) perform transfer learning between different VAEs structures. In all cases, using a efficient inference formulation based on mean-field VI. The code to reproduce ours experiments can be found at [github.com/aguerrerolopez/FA-VAE](http://github.com/aguerrerolopez/FA-VAE).

The article is organised as follows: Section 2 reviews the SSHIBA [26] and the VAE [1] formulations and, introduces the proposed formulation of FA-VAE. Section 3 defines three experiments to demonstrate that the proposed approach: (i) can condition a pretrained VAE, (ii) performs domain adaptation, and (iii) can be used for transfer learning between VAEs. Finally, in Section 4 we give some final remarks and highlight the main results.

2. **Methods**

2.1. **Sparse Semi-supervised Heterogeneous Interbattery Bayesian Analysis**

SSHIBA [26, 27] presents a solution to multi-view problems with samples represented in $M$ different modalities where each view can be either multilabel, binary, real, categorical, or other multidimensional object. The general model framework, depicted in Fig. 1a, considers that the $n$-th sample of the $m$-th view, $x_{n:}^{(m)} \in \mathbb{R}^{1 \times D}$, can be projected into a low-dimensional latent space, $g_{n:} \in \mathbb{R}^{1 \times K_c}$, where $K_c$ is the number of latent factors of this common space. As seen in Fig. 1b, the $m$-th view for the $n$-th observation, $x_{n:}^{(m)}$, can be generated by linearly
combining $g_n$: with a projection matrix $W^{(m)} \in \mathbb{R}^{D \times K_c}$, i.e.,

$$x_n^{(m)} = g_n \cdot W^{(m)T} + \tau^{(m)},$$  \hspace{1cm} (1)$$

where $\tau^{(m)}$ is zero-mean Gaussian noise, with noise power following a Gamma distribution of parameters $a^{\tau(m)}$ and $b^{\tau(m)}$.

Figure 1: SSHIBA basic structure. Gray circles denote observations, white circles represent random variables (rv), and the non-circles are hyperparameters. Subfigure 1a represents the latent space projection $g_n$: given all $m$ input views and subfigure 1b represents how a real $m$-view is modeled.

SSHIBA also includes a column-wise Automatic Relevance Determination (ARD) prior \cite{28} over each view’s projection matrix through the variable $\alpha_k^{(m)}$ which follows a Gamma distribution, $\alpha_k^{(m)} \sim \Gamma(a^{\alpha(m)}, b^{\alpha(m)})$. This prior removes the need for cross validating the number of latent factors and it implies that each $W^{(m)}$ effectively selects which part of the global latent space $G$ is specific to each $m$-view (intra-view) or shared among views (inter-view).

The model is trained by evaluating the posterior distribution of all rv given the observed data. However, due to the intractability of these posteriors, they are approximated through mean-field VI \cite{29} to the posterior distribution with
a fully factorised variational family \( q \) as:

\[
p(\Theta | X^{(m)}) \approx \prod_{m=1}^{M} (q(W^{(m)})) \prod_{k=1}^{K_c} q(\alpha_k^{(m)}) \prod_{n=1}^{N} q(g_{n,:}),
\]

(2)

where \( \Theta \) is a vector of all rv in the model, shown in Fig. 1b. Therefore, combining the mean-field posterior with the Evidence LowerBOund (ELBO) results into a feasible coordinate-ascent-like optimization algorithm where each rv can be computed by maintaining the rest fixed following:

\[
q^*(G) \propto \mathbb{E}_{\Theta_{-i}} [\log p(\Theta_{-i}, g_{1,:}, ..., g_{N,:})]
\]

(3)

being \( \Theta_{-i} \) all rv but \( \theta_i \). Using this formulation, we create an efficient and feasible optimization problem, since it does not require a complete marginalization of all \( \Theta \) rv from the joint distribution. The update rules of every rv can be seen in Table 1. The complexity bottleneck of SSHIBA is to invert \( \Sigma^{-1}_{G} \in \mathbb{R}^{K_c \times K_c} \), and to invert \( \Sigma^{-1}_{W^{(m)}} \in \mathbb{R}^{D \times D} \), where for high-dimensional data kernel methods can be used as demonstrated in [27].

Furthermore, the Bayesian nature of the model allows it to work in a semi-supervised fashion, using all available information to determine the approximate distribution of the variables. In turn, the model can marginalise out any type of missing values in the data, as well as predict test samples for any view by sampling from its variational distribution.

2.2. Variational AutoEncoders

A VAE is a probabilistic model that assumes there exists a hidden latent variable \( z \in \mathbb{R}^{1 \times D'} \), where \( D' \) is a hyperparameter, capable of generating the observations \( x \in \mathbb{R}^{1 \times D} \), where \( D \) is the observation dimension, through a non-linear model. The latent variables are inferred from the observations following:

\[
p_{\theta}(z|x) = \frac{p_{\theta}(x|z)p(z)}{p(x)},
\]

(4)
Table 1: Updated distributions for the different rv of the model. These expressions have been obtained using the update rules of the mean field approximation. See [26] for further details.

| Variable | \( q^* \) distribution | Parameters |
|----------|--------------------------|------------|
| \( g_n: \) | \( \mathcal{N}(g_n | \mu_{g_n}, \Sigma_G) \) | \( \mu_{g_n} = \frac{1}{M} \sum_{m=1}^{M} \langle \tau^{(m)} \rangle X^{(m)} (W^{(m)}) \Sigma_G \) \( \Sigma^{-1}_G = I_{K_c} + \frac{1}{M} \sum_{m=1}^{M} \langle \tau^{(m)}(W^{(m)})^T W^{(m)} \rangle \) |
| \( W^{(m)} \) | \( \prod_{d=1}^{D_m} \mathcal{N}(w_d^{(m)} | \mu_{w_d^{(m)}}, \Sigma_{W^{(m)}}) \) | \( \mu_{w_d^{(m)}} = \langle \tau^{(m)} \rangle X^{(m)}(G^{(m)} \Sigma_{W^{(m)}}) \) \( \Sigma^{-1}_{W^{(m)}} = \text{diag}(\langle \alpha^{(m)} \rangle, \langle \gamma^{(m)} \rangle) + \langle \tau^{(m)} \rangle (G^{T} G) \) |
| \( \alpha^{(m)} \) | \( \prod_{k=1}^{K} \Gamma(\alpha_k^{(m)} | a_{\alpha_k^{(m)}}, b_{\alpha_k^{(m)}}) \) | \( a_{\alpha_k^{(m)}} = \frac{D_m}{2} + \alpha^{(m)} \) \( b_{\alpha_k^{(m)}} = b^{(m)} + \frac{1}{2} \sum_{d=1}^{D_m} \langle \gamma_d^{(m)} \rangle (w_d^{(m)} w_d^{(m)}) \) |
| \( \tau^{(m)} \) | \( \Gamma(\tau^{(m)} | a_{\tau^{(m)}}, b_{\tau^{(m)}}) \) | \( a_{\tau^{(m)}} = \frac{D_m}{2} + a^{(m)} \) \( b_{\tau^{(m)}} = b^{(m)} + \frac{1}{2} \sum_{n=1}^{N} \sum_{d=1}^{D_m} \langle \gamma_n^{(m)} \rangle^2 - 2 \text{Tr} \left( (W^{(m)})^T (G^{T} X^{(m)}) \right) + \text{Tr} \left( (W^{(m)})^T (W^{(m)}) (G^{T} G) \right) \) |

where, on the one hand, \( p_\theta(x|z) \) corresponds to the decoder, which is based on a Neural Network (NN) parametric model with \( \theta \) parameters and, on the other hand, \( p_\theta(z|x) \) is the encoder (see in Fig. 2 the probabilistic model). However, its computation by means of Eq. [4] is not feasible, since \( p(x) \) calculation is intractable. Hence, we use VI to approximate it with a tractable parametrised family of distributions defined as \( q_\eta(z|x) \). To do so, the KL divergence between both distributions is minimised:

\[
\min_{\eta} \text{KL}(q_\eta(z|x) || p_\theta(z|x))
\]

which is equivalent to maximizing the following ELBO [1]:

\[
\mathcal{L}_{\theta, \eta} = \mathbb{E}_{q_\eta(z|x)} [\log (p_\theta(x|z))] - \text{KL}(q_\eta(z|x) || p(z)),
\]

where, on the one hand, \( p_\theta(x|z) \) corresponds to the decoder, which is based on a Neural Network (NN) parametric model with \( \theta \) parameters and, on the other hand, \( p_\theta(z|x) \) is the encoder (see in Fig. 2 the probabilistic model). However, its computation by means of Eq. [4] is not feasible, since \( p(x) \) calculation is intractable. Hence, we use VI to approximate it with a tractable parametrised family of distributions defined as \( q_\eta(z|x) \). To do so, the KL divergence between both distributions is minimised:

\[
\min_{\eta} \text{KL}(q_\eta(z|x) || p_\theta(z|x))
\]

which is equivalent to maximizing the following ELBO [1]:

\[
\mathcal{L}_{\theta, \eta} = \mathbb{E}_{q_\eta(z|x)} [\log (p_\theta(x|z))] - \text{KL}(q_\eta(z|x) || p(z)),
\]

7
where the likelihood term $\text{I}$ ensures the reconstruction of the observations and $\text{II}$ promotes that the learned distribution $q_\eta(z|x)$ is similar to the prior distribution $p(z)$.

Here, we use a variant of this model, the $\beta$-VAE [30] which multiplies the regularization term by a $\beta$ hyperparameter ($\beta>1$), resulting in a more disentangled latent representation when the prior $p(z)$ is independent.

2.3. Factor Analysis Variational AutoEncoder

Here, we propose a novel hierarchical method, called Factor Analysis VAE (FA-VAE), able to combine multiple VAEs by disentangling the private and shared information of heterogeneous observations via SSHIBA. In particular, our proposal consists in using the multiview nature of SSHIBA to combine the latent information from different VAEs in a shared latent space. Given the $m$-view, its private information is represented by a continuous latent variable, $z_{n,m}$, provided by a VAE. Then, all private latent variables $z_{n,m}$, for $m = 1, ..., M$, project their shared information into a global latent variable, $g_{n,:}$.

The private latent representation of the $m$-th view is calculated by a VAE that encodes the observations using the variational distribution $z_{n,m} \mid x_{n,m} \sim q_\eta(z_{n,m})$, defined as

$$q_\eta(z_{n,m}) \sim \mathcal{N} \left( \mu_{\eta}^{z_{n,m}}(x_{n,m}), \Sigma_{\eta}^{z_{n,m}}(x_{n,m}) \right), \quad (7)$$

where $\mu_{\eta}^{z_{n,m}}(x_{n,m}), \Sigma_{\eta}^{z_{n,m}}(x_{n,m})$ are the output of an independent parametric encoder for the $m$-th view. The FA-VAE’s graphical model is included in Fig. [30]
Figure 3: FA-VAE graphical model example with two VAEs. The blue dotted rectangle denote the SSHIBA rv while the red rectangles indicate the two VAEs structures, one per view. Gray circles denote observations, and white circles represent rv.

where the generative model assumes

$$z^{(m)}_{n, :} \sim \mathcal{N} \left( g_{n, :}; W^{(m)}, \tau^{(m)^{-1}} \right), \quad (8)$$

and the observations $x^{(m)}_{n, :}$ are generated from $z^{(m)}_{n, :}$ using the VAE conditional distribution $x^{(m)}_{n, :} | z^{(m)}_{n, :} \sim p_{\theta}(x^{(m)}_{n, :} | z^{(m)}_{n, :})$, which we define as

$$x^{(m)}_{n, :} \sim \mathcal{N}\left( \mu_{\theta}^{(m)}(z^{(m)}_{n, :}), \sigma \right) \quad (9)$$

where $\mu_{\theta}^{(m)}(z^{(m)}_{n, :})$ is the output of an independent parametric decoder for the $m$-th view and $\sigma$ is fixed.

Regarding each $m$-th pair encoder-decoder, we can arbitrarily choose their internal structure. This flexibility allows FA-VAE to model heterogeneous observations, in addition to those that SSHIBA could already model, such as images with a Convolutional Neural Network (CNN) or sequential data with a Recursive Neural Network (RNN).

As shown in Eq. (6), the ELBO maximization of a VAE ensures that the
variational distribution $q_{\eta}(z_{n}^{(m)} | x_{n}^{(m)})$ is close to the prior distribution $p(z)$ by minimizing their KL divergence. Standard VAE formulation usually considers that $p(z) \sim \mathcal{N}(0, I)$, however, in SSHIBA’s framework the prior over $z_{n}^{(m)}$ is imposed by factor analysis decomposition. Thus, to integrate both models, we have to obey Eq. (8), where $g_{n;:}, W^{(m)}$, and $\tau^{(m)}$ are sampled from their posterior distribution (see Table 1). Therefore, the ELBO maximised by each $m$-VAE is

$$L^{(m)} = \mathbb{E}_{q_{\eta}(z_{n}^{(m)} | x_{n}^{(m)})} \log \left( p_{\theta}(x_{n}^{(m)} | z_{n}^{(m)}) \right) - \beta \text{KL} \left( q_{\eta}(z_{n}^{(m)} | x_{n}^{(m)}) \| \mathcal{N} \left( g_{n;:], W^{(m)}, \tau^{(m)}^{-1} \right) \right),$$

where the first term is the Gaussian Log-Likelihood (GLL) between $x_{n}^{(m)}$ and samples from $p_{\theta}(x_{n}^{(m)} | z_{n}^{(m)})$, and the second term minimises the KL divergence between the variational distribution and SSHIBA’s rv.

Finally, using the mean-field approach, we calculate each $q(\theta_i)$ following Table 1 where, in case of updating a $m$-VAE views, $X^{(m)}$ is now sampled by the encoder $q(Z^{(m)})$ resulting in $Z^{(m)}$ samples. Following this procedure, FA-VAE is trained following the steps detailed in Algorithm 1.

**Algorithm 1: FA-VAE training algorithm**

```
Initialise $G, W^{(m)}, z_{n}^{(m)}, \alpha_k^{(m)}, \tau^{(m)}$;
while FA-VAE not converge do
  Update $q(G)$ following 1st row of Table 1
  for each $m$-view do
    Update $q(W^{(m)})$ following 2nd row of Table 1
    for each epoch do
      Maximise the $m$-VAE’s ELBO (Eq. (10))
    end
    Update $q(z_{n}^{(m)})$ sampling from $q_{\eta}(z_{n}^{(m)} | x_{n}^{(m)}) \sim \mathcal{N}(\mu_{\eta} z_{n}^{(m)}, \Sigma_{\eta} z_{n}^{(m)})$;
    Update $q(\alpha_k^{(m)})$ following 3th row of Table 1
  end
  Update $q(\tau^{(m)})$ following 4th row of Table 1
end
```
3. Experiments

Throughout this section, we demonstrate the flexibility of FA-VAE in addressing relevant problems in deep probabilistic modelling. In Section 3.1, we analyse how FA-VAE is able to condition a pretrained VAE to multilabel targets. In Section 3.2, we apply it over a domain adaptation problem and compare it to the Multi-VAE model [21], besides, we disentangle and analyse the private-shared latent variables. Finally, in Section 3.3 we use the proposed FA-VAE’s framework to perform transfer learning between multiple VAEs, showing how the transfer learning creates a more expressive and understandable latent space than other models such as β-VAE [30]. The code to reproduce the following experiments can be found at [github.com/aguerrerolopez/FA-VAE](http://github.com/aguerrerolopez/FA-VAE).

3.1. FA-VAE as a conditioned generative model

In this scenario, we show how we can use FA-VAE’s framework to easily adapt a pretrained unconditioned VAE, \( p_\theta(x) \), to model a conditional distribution, \( p_\theta(x|a) \), for a given set of labeled data \( \{x_i, a_i\}_{i=1}^N \) where \( x_i \) is an observation and \( a_i \) is an attribute. For this experiment, we use CelebA dataset [24], and we restrict it to 30000 samples and we select three attributes: wearing lipstick, gender and smiling. Therefore, we use 30000 \( \times 64 \times 64 \times 3 \) RGB images with a stratified proportion of these three attributes.

![Figure 4: Conditioning a single VAE to a multilabel attribute vector using FA-VAE architecture where \( A \) denote attributes view and \( O \) observations views. Gray circles are observations, and white circles represent rv.](http://example.com/figure4.png)
Using FA-VAE multi-view framework, we propose to model the problem with a two-view setup following the graphical model shown in Fig. 4. The first view is in charge of modeling the attributes \((A)\) as a multilabel vector with three attributes: wearing lipstick, female/male, and smiling. For this view we use the standard multilabel SSHIBA’s configuration without any VAE. In this case, the binary attributes are generated, denoted as \(x^{(A)}_{n,:}\), from the private variable \(z^{(A)}_{n,:}\) using a Bernoulli distribution as

\[
p(x^{(A)}_{n,:} | z^{(A)}_{n,:}) = \prod_{d=1}^{3} \mathbb{E}^{z^{(A)}_{n,d}} \mathbb{E}^{x^{(A)}_{n,d}} \sigma(-z^{(A)}_{n,d})
\]

where \(\sigma\) is the sigmoidal function.

The second view \((O)\) is composed by observed RGB images, denoted as \(x^{(O)}_{n,:} \in \mathbb{R}^{64 \times 64 \times 3}\). For both, encoder and decoder, we borrow the network structure proposed in the \(\beta\)-VAE paper [31]. Namely, we use a CNN network with 5 convolutional layers of 64, 128, 256, 512, 1024 channel size with kernel size of 4, stride of 2, and padding 1. This CNN structure is followed by a fully connected layer to generate \(\mu_{\eta}\), and \(\Sigma_{\eta}\) and, then, infer the real pseudo-observation \(z^{(O)}_{n,:} \in \mathbb{R}^{100}\). The decoder follows the inverse structure of the encoder.

We train from scratch an unconditioned vanilla VAE over CelebA until convergence. Then, we add this unconditioned pretrained vanilla VAE inside FA-VAE’s architecture to condition it to the attributes. Fig 5a analyses the model convergence for these two steps. In particular, Fig. 5a analyses the ELBO of the vanilla VAE trained from scratch, whereas Fig. 5b shows the ELBO evolution during the FA-VAE’s fine tuning. This curve shows that the pretrained VAE keeps stable and does not loss reconstruction power. Therefore, we are able to condition a pretrained VAE without interfering in its learning.

We can use this model to alter the attributes of a given image. To illustrate it, on the one hand, we get a CelebA image \(x^{(O)}_{n,:}\), we project it to its private latent space \(z^{(O)}_{n,:}\) using \(q_{\eta}(z^{(O)}_{n,:} | x^{(O)}_{n,:})\) and fix \(z^{(O)}_{n,:}\) as our reference image. On the other hand, we select a set of attributes to change our image and we generate
Figure 5: VAEs convergence by its own and inside FA-VAE’s architecture. In Fig. 5a is shown the ELBO of a vanilla VAE trained over CelebA from scratch. In Fig. 5b we plug the vanilla VAE from Fig. 5a inside FA-VAE’s architecture.

3.2. Domain adaptation

Due to FA-VAE modularity, we can go one step further and combine multiple VAEs simultaneously. For this purpose, let’s consider an illustrative example based on CelebA [24] dataset and Google Cartoon Set [32] dataset and, let’s consider a three view FA-VAE configuration (Fig. 7). On the first view, we train a VAE with 10000 CelebA images. On the second view, we train another VAE
Figure 6: Images provided by FA-VAE when modifying different attributes. The left column of each subfigure represents the raw image. The center and right columns of each subfigure represent the modified images by new attributes which are indicated as title meaning [smile, lipstick, gender].

with 10000 Cartoon set images. And, in the third view, we add a binary label to model the hair color with a SSHIBA layer.

We hypothesise that the private latent variables $z^{(m)}_n$ can capture the domain information of face images, cartoon avatars and, hair color. Then, the shared latent space variable, $g_n$, works as a bridge between domains to adapt real world faces to 2D cartoon avatars given a hair color.

Figure 7: FA-VAE configuration to perform domain adaptation between two VAE based views representing real-world faces (CelebA) and cartoon avatars (Cartoon), while conditioning to a third categorical view (hair). Gray circles denote observations and, white circles represent rv.

The CelebA view ($C$) is composed by $10000 \times 64 \times 64 \times 3$ observations of images of celebrity real faces whereas the Cartoon view ($D$) is composed by $10000 \times 64 \times 64 \times 3$ observations of cartoon avatar images. For both views, we use the $\beta$-VAE in $[30]$ encoder-decoder configuration already proposed in Section 3.1.
using $\beta > 1$ as the authors recommend. Finally, the hair, $H$, view is composed by a $10000 \times 2$ binary label indicating hair color label: blond or brunet.

We compare FA-VAE with the Multi-VAE [21] model which uses a $c_n$ discrete latent variable to share the context of all views (see Fig. 8). For this model, we incorporate the hair label as an extra-dimension in $x_{n, c}$ and $x_{n, d}$.

Fig. 8 shows how FA-VAE (second row) and Multi-VAE (third row) translates images from the CelebA domain (first row) to the Cartoon domain. Although the only attribute seen by the models is the hair color, FA-VAE outperformed Multi-VAE by capturing other inherent attributes, such as the use of sunglasses. Moreover, Multi-VAE do not properly learn skin color as seen in Images 3 and 9 while FA-VAE does. Finally, FA-VAE generates high-quality 2D avatars with no blur or artifacts.

The private latent variables $Z^{(D)}$ and $Z^{(C)}$ codify the information of each
domain, while the global space $G$ enables information transference between both domains. We can use this to apply domain adaption by translating images from one domain to the other. For this purpose, first, we get two CelebA images and project them to their private latent space as $z_{1:C}$ and $z_{2:C}$. Then, we project them to the shared latent space $g_{1:*}, g_{2:*}$ to, later, sample from the cartoon private latent space $Z^{(D)} \sim \mathcal{N}(g_{m:}^{*}, W^{(D)}, \tau^{(D)})$, obtaining $z_{1:D}$ and $z_{2:D}$. Now, we interpolate the private representation of each pair of images in each domain using the convex combination

$$z_{m:}^{(m)}_{\lambda:} = \lambda z_{1:}^{(m)} + (1-\lambda) z_{2:}^{(m)},$$

where $m \in \{C, D\}$ represent any of the two views, and $\lambda \in [0, 1]$ allows us to move from Image 1 to Image 2. Finally, using the generated private representation $z_{\lambda:}^{(m)}$, we can use $p_{\theta}^{m}(x_{\lambda:}^{(m)} | z_{\lambda:}^{(m)})$ to generate the sample of each domain. The reconstructed sequences $x_{\lambda:}^{(m)} | z_{\lambda:}^{(m)}$ for different values of $\lambda$ are shown in Fig. 10.

Figure 10: Examples of an image transformation and domain adaption application. The first and second rows show the evolution from $z_{1:C}$ to $z_{2:C}$ and, from $z_{1:D}$ to $z_{2:D}$, respectively.

In both rows of Fig. 10 we can see a complete evolution from one image to the other. Any point sampled from the latent space has generated a meaningful image, demonstrating completeness. Additionally, the evolution between both images is clean and gradual, demonstrating continuity. In fact, in cartoon domain we can see how the avatar evolves in hairstyle, hair color and eyeglasses. Hence, both latent variables, $Z^{(D)}$ and $Z^{(C)}$, fulfill completeness and continuity, thus, both are informative and explainable. Similarly, the global latent space $G$ also demonstrates completeness and continuity as shown in Appendix A.2.
3.3. Transfer learning

A problem when training powerful VAEs, i.e., very deep variational and generative networks, is the required computational cost due to the number of epochs needed to learn huge numbers of parameters and the need of large datasets. Thus, we propose a novel method to speed up this process, using FA-VAE as a transfer learning tool between VAEs over the same domain.

Let’s consider an illustrative example based on CelebA dataset with a two-view set-up following the graphic model shown in Fig. 11. On the first view (V), we use a pretrained vanilla VAE, in fact, we use the one trained in Section 3.1. As we plug a pretrained VAE, both encoder and decoder are not going to be trained, thus \( z^{(V)} \) is going to be static. Then, on the second view (O), we start from the architecture of \( \beta \)-VAE shown in Section 3.1 and, to make it deeper, we add an extra final CNN layer of 2048 channel size with kernel size of 4, stride of 2, and padding 1 in the encoder and, likewise, its inverse in the decoder.

![Transfer learning graphical model using FA-VAE](image)

**Figure 11**: Transfer learning graphical model using FA-VAE. The V view represent information pre-learned by a vanilla VAE. As it is pretrained, \( z^{(V)} \) is no longer a rv but an observation. The O view represent CelebA images using a \( \beta \)-VAE. Gray circles denote observations, and white circles represent rv.

We expect that the private space provided by a pretrained vanilla VAE, \( z^{(V)} \), speeds up the training of a deeper VAE or even helps find a better global solution. To demonstrate this functionality, we compare two scenarios: our transfer learning approach using FA-VAE and, a single \( \beta \)-VAE with the same structure as the second view of FA-VAE. Fig. 12 shows the performance of both
approaches. As seen in Fig. 12a, FA-VAE presents an accurate reconstruction capability on the first epochs, meaning that the latent space of the vanilla VAE can provide a good initialisation to β-VAE. Moreover, in terms of speed, FA-VAE gets the same maximum GLL as β-VAE in, approximately, 4 times less epochs. Furthermore, FA-VAE scores a highest absolute value in terms of GLL in comparison to β-VAE by its own. Looking at Fig. 12b we can see that FA-VAE achieves a lower global KL term. The periodical spikes correspond to every time that the SSHIBA part of FA-VAE updates the VAE prior distribution as seen in Algorithm 1. Meanwhile, for β-VAE the KL starts increasing while in FA-VAE directly goes down, justifying the better behavior since the beginning.

![Diagram](image)

(a) Reconstruction term measured by GLL.  
(b) KL divergence term.

Figure 12: ELBO decomposition in reconstruction term and KL divergence term. Red line represents our approach, FA-VAE, while black line represent the β-VAE by its own.

In Fig. 13 we show 5 random test images, i.e., never seen before by any of the models. We use both β-VAE and FA-VAE to encode the images to their private latent space, and, then we reconstruct them back to their original domain. As it might not be clear to the human eye which method reconstruct better the images, in Table 2 we include the R2 score on the reconstruction over 10000 test images. Here, we can see that FA-VAE provides a better performance in terms of R2 score, demonstrating that the transfer learning improves the model performance.

Other advantage of FA-VAE is its ability to create a more expressible and meaningful private latent representation of the images than β-VAE. To demonstrate this, we show the impact of arbitrarily modifying the 10 most relevant
features, determined by their absolute values in the weight matrix \( W^{(m)} \), of both private latent variables: \( z \in \mathbb{R}^{1 \times 100} \) for \( \beta \)-VAE, and \( z^{(O)}_{m,n} \in \mathbb{R}^{1 \times 100} \) for FA-VAE by randomly modifying their values in the interval \([-20, 20]\).

For \( \beta \)-VAE, in Fig. 14a each row represents one of the 10 most relevant features. The \( z \) latent space shows that there are 3 features with visual interpretation. Blue rows show that we can control the facial hair by increasing or decreasing their value. Besides, green row shows an impact into the contrast of the image. However, the remaining features do not present a visual explainable impact on the images.

Fig 14b shows the same analysis over FA-VAE, revealing that FA-VAE’s private latent space greatly outperforms \( \beta \)-VAE’s in both explainability and cleanliness terms. In particular, 9 out of the 10 most relevant features have a visual interpretation; for example, golden marked latent features control the skin tone and the rotation of the face, the green rows control the gender and the hair style, the blue row controls the background color between cold and warm, and, the gray marked latent features control the smiling. Moreover, we can see the proposed model is capable of focusing on the face information and filter noisy

| Model  | Samples | R2 score  |
|--------|---------|-----------|
| \( \beta \)-VAE | 10000   | 0.941 ± 0.032 |
| FA-VAE | 10000   | **0.969 ± 0.027** |

Table 2: Reconstruction performance measured in R2 score over 10,000 CelebA test samples.
Figure 14: Latent space evolution. Each row represents the 10 most relevant features based on absolute value. The red column represents the image generated by the model without any modification. Then, all images at the left of the red column are images generated by arbitrarily decreasing the value of each feature. Likewise, all images at the right of the red column are images generated by increasing the value of each feature.

background in the latent space.

4. Conclusions

This paper presents FA-VAE model, to the best of our knowledge, the first deep hierarchical VAE for heterogeneous data using an interpretable factor analysis latent space. FA-VAE is capable of conditioning multiple VAEs creating a model that can work with multiple data domains, from multi-label to continuous, binary, categorical, or even image data, depending on the VAE architecture. This conditioning is performed by a deep hierarchical structure using factor analysis, learning a disentangled and explainable latent space.

Besides, given the multiview capability of this model, we have shown how we can perform domain adaptation between two different databases while conditioning them on external attributes outperforming SOTA models. In addition, FA-VAE is the first model able to perform transfer learning between generative models, thus accelerating the learning process and even improving the performance. Thus, FA-VAE provides a robust model capable of dealing with
real-world data sets.

Acknowledgements

This work was supported by MCIN AEI 10.13039 501100011033 [grant numbers TEC2017-83838-R, RTI2018-099655-B-10, PID2020-115363RB-I00, PID2021-123182OB-I00, TED2021-131823B-I00]; the Comunidad de Madrid [grant numbers IND2018/TIC-9649, Y2018/TCS-4705]; the European Union (FEDER and the European Research Council (ERC) through the European Unions Horizon 2020 research innovation program [grant number 714161]; and the IISGM [grant INTRAMURAL].

References

[1] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114.

[2] D. Barrejón, P. M. Olmos, A. Artés-Rodríguez, Medical data wrangling with sequential variational autoencoders, IEEE Journal of Biomedical and Health Informatics.

[3] X. Yan, D. She, Y. Xu, M. Jia, Deep regularized variational autoencoder for intelligent fault diagnosis of rotor–bearing system within entire life-cycle process, Knowledge-based Systems 226 (2021) 107142.

[4] L. Maaløe, M. Fraccaro, V. Liévin, O. Winther, Biva: A very deep hierarchy of latent variables for generative modeling, Advances in neural information processing systems 32.

[5] C. He, L. Zheng, T. Tan, X. Fan, Z. Ye, Manifold discrimination partial adversarial domain adaptation, Knowledge-Based Systems (2022) 109320.

[6] X. Zhang, Y. Yang, D. Zhai, T. Li, J. Chu, H. Wang, Local2global: Unsupervised multi-view deep graph representation learning with nearest neighbor constraint, Knowledge-Based Systems 231 (2021) 107439.
[7] I. Peis, P. M. Olmos, A. Artés-Rodríguez, Unsupervised learning of global factors in deep generative models, arXiv preprint arXiv:2012.08234.

[8] A. Nazabal, P. M. Olmos, Z. Ghahramani, I. Valera, Handling incomplete heterogeneous data using vaes, Pattern Recognition 107 (2020) 107501.

[9] C. K. Sonderby, T. Raiko, L. Maaløe, S. K. Sonderby, O. Winther, Ladder variational autoencoders, Advances in neural information processing systems 29.

[10] A. Klushyn, N. Chen, R. Kurle, B. Cseke, P. van der Smagt, Learning hierarchical priors in vaes, Advances in neural information processing systems 32.

[11] M. J. Vowels, N. C. Camgoz, R. Bowden, Nestedvae: Isolating common factors via weak supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9202–9212.

[12] M. Lee, V. Pavlovic, Private-shared disentangled multimodal vae for learning of latent representations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1692–1700.

[13] N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, M. Shanahan, Deep unsupervised clustering with gaussian mixture variational autoencoders, arXiv preprint arXiv:1611.02648.

[14] L. Yang, W. Fan, N. Bouguila, Robust unsupervised image categorization based on variational autoencoder with disentangled latent representations, Knowledge-Based Systems 246 (2022) 108671.

[15] I. Peis, C. Ma, J. M. Hernández-Lobato, Missing data imputation and acquisition with deep hierarchical models and hamiltonian monte carlo, arXiv preprint arXiv:2202.04599.

[16] A. L. Caterini, A. Doucet, D. Sejdinovic, Hamiltonian variational autoencoder, Advances in Neural Information Processing Systems 31.
[17] M. Suzuki, K. Nakayama, Y. Matsuo, Joint multimodal learning with deep generative models, arXiv preprint arXiv:1611.01891.

[18] Y. Li, Y. Zhang, C. Yang, Unsupervised domain adaptation with joint adversarial variational autoencoder, Knowledge-Based Systems (2022) 109065.

[19] M. Wu, N. Goodman, Multimodal generative models for scalable weakly-supervised learning, Advances in Neural Information Processing Systems 31.

[20] W. Youpeng, L. Hongxiang, G. Yiju, Z. Liang, Amvae: Asymmetric multimodal variational autoencoder for multi-view representation, in: International Conference on Artificial Neural Networks, Springer, 2021, pp. 391–402.

[21] J. Xu, Y. Ren, H. Tang, X. Pu, X. Zhu, M. Zeng, L. He, Multi-vaes: Learning disentangled view-common and view-peculiar visual representations for multi-view clustering, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 9234–9243.

[22] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, arXiv preprint arXiv:1611.01144.

[23] L. Deng, The mnist database of handwritten digit images for machine learning research, IEEE Signal Processing Magazine 29 (6) (2012) 141–142.

[24] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of International Conference on Computer Vision (ICCV), 2015.

[25] S. Fidler, S. Dickinson, R. Urtasun, 3d object detection and viewpoint estimation with a deformable 3d cuboid model, Advances in neural information processing systems 25.

[26] C. Sevilla-Salcedo, V. G’omez-Verdejo, P. Olmos, Sparse Semi-supervised Heterogeneous Interbattery Bayesian Analysis, Pattern Recognit. doi:10.1016/j.patcog.2021.108141

23
[27] C. Sevilla-Salcedo, A. Guerrero-López, P. M. Olmos, V. Gómez-Verdejo, Bayesian sparse factor analysis with kernelized observations, Neurocomputing.

[28] C. Bishop, Bayesian pca, in: M. Kearns, S. Solla, D. Cohn (Eds.), Advances in Neural Information Processing Systems, Vol. 11, MIT Press, 1998. URL https://proceedings.neurips.cc/paper/1998/file/c88d860a8097754525e02c22d8d27f-Paper.pdf

[29] D. M. Blei, A. Kucukelbir, J. D. McAuliffe, Variational inference: A review for statisticians, Journal of the American statistical Association 112 (518) (2017) 859–877.

[30] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, Understanding disentangling in β-vae, arXiv preprint arXiv:1804.03599.

[31] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, beta-vae: Learning basic visual concepts with a constrained variational framework.

[32] A. Royer, K. Bousmalis, S. Gouws, F. Bertsch, I. Mosseri, F. Cole, K. Murphy, Xgan: Unsupervised image-to-image translation for many-to-many mappings, in: Domain Adaptation for Visual Understanding, Springer, 2020, pp. 33–49.
Appendix A. Experiments

In this appendix we add extra experiments that demonstrate the potential of FA-VAE. The structure follows the same order as in the main article.

Appendix A.1. FA-VAE as a conditioned generative model

Following the same experiment shown in Section 3.1, we show that we can also generate random conditioned faces.

To analyse it, we show how the proposed model is able to generate images conditioned to certain attributes, we proceed as follows: (i) we create random \( x^{(A)}_{n,:} \) multilabel vectors using [wearing lipstick, gender, smiling] binary notation, (ii) we create their real pseudo-observation \( z^{(A)}_{n,:} \), (iii) we generate the posterior distribution of \( g_{n,:} \) given \( z^{(A)}_{n,:} \) which follows a Gaussian distribution with parameters:

\[
\Sigma^{-1}_{g_{n,:}} = I_{K_c} + \tau^{(A)}_{(A)} W^{(A)}^T W^{(A)}
\]

\[
\mu_{g_{n,:}} = \tau^{(A)}_{(A)} z^{(A)}_{n,:} W^{(A)} \Sigma_{g_{n,:}}^{-1}
\]

(A.1)

(iv) we sample \( z^{(O)}_{n,:} \sim N(g^{(A)}_{n,:}, W^{(O)}_{n,:}, \tau^{(O)}) \), and, finally (v) we use FA-VAE’s generative distribution \( p_\theta(\alpha_{n,:}\mid z^{(O)}_{n,:}) \) to sample artificially generated conditioned images. Fig. A.15 shows these images generated by 8 random \( x^{(A)}_{n,:} \) multilabel attributes.

As shown, a pretrained VAE can easily be conditioned to arbitrary attributes by training 150 epochs of FA-VAE.

Appendix A.2. Domain adaptation

Following the same experiment shown in Section 3.2 we show that the global latent variable is also complete and continuous.

Similarly, as did in 3.2 with the private latent space interpolation, we could implement such interpolation directly from the global space by sampling from \( g_{n,:} \) and generating a sample that can be understood by both domains. The next two examples illustrate that behavior. Again, we select two new CelebA
observations and we project them to the global space $G$ as $g_1$ and $g_2$. Then, we calculate $g_{\lambda}$ following Eq. (13). For this scenario, each $g_{\lambda}$ can be decoded by both generative networks simultaneously creating a pair of images. In Fig. A.16 the reconstructed sequences $x^{(m)}_{\lambda}$ are shown.

Every $g_{\lambda}$ generates meaningful content showing completeness. However, there is a trade-off between both domains. On one hand, the CelebA domain shows better continuity showing a clear transition between both images. On the other hand, the Cartoon domain, characterised by a discrete set of features, shows better completeness where every $g_{\lambda}$ creates a cartoon avatar without any distortion or artifact.