Multi-Level Variational Autoencoder: Learning Disentangled Representations from Grouped Observations

Diane Bouchacourt
OVAL Group
University of Oxford [*]
diane@robots.ox.ac.uk

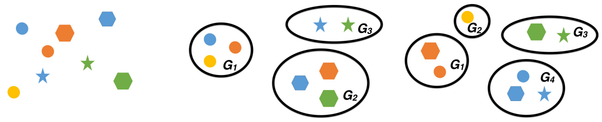
Ryota Tomioka, Sebastian Nowozin
Machine Intelligence and Perception Group
Microsoft Research
Cambridge, UK
{ryoto,Sebastian.Nowozin}@microsoft.com

Abstract

We would like to learn a representation of the data that reflects the semantics behind a specific grouping of the data, where within a group the samples share a common factor of variation. For example, consider a set of face images grouped by identity. We wish to anchor the semantics of the grouping into a disentangled representation that we can exploit. However, existing deep probabilistic models often assume that the samples are independent and identically distributed, thereby disregard the grouping information. We present the Multi-Level Variational Autoencoder (ML-VAE), a new deep probabilistic model for learning a disentangled representation of grouped data. The ML-VAE separates the latent representation into semantically relevant parts by working both at the group level and the observation level, while retaining efficient test-time inference. We experimentally show that our model (i) learns a semantically meaningful disentanglement, (ii) enables control over the latent representation, and (iii) generalises to unseen groups.

1 Introduction

Representation learning refers to the task of learning a representation of the data that can be easily exploited (Bengio, Courville, and Vincent 2013). Our goal is to build a model that disentangles the data into separate salient factors of variation and easily applies to a variety of tasks and different types of observations. Towards this goal there are multiple difficulties. First, the representative power of the learned representation depends on the information one wishes to extract from the data. Second, the multiple factors of variation impact the observations in a complex and correlated manner. Finally, we have access to very little, if any, supervision over these different factors. If there is no specific meaning to embed in the desired representation, the *infomax principle* Linsker (1988) states that an optimal representation is one of bounded entropy that retains as much information about the data as possible. By contrast, in our case there exists a semantically meaningful disentanglement of interesting latent factors. How can we anchor semantics in high-dimensional representations?



(a) Objects of multiple shapes and colors.

(b) Objects grouped by shape.

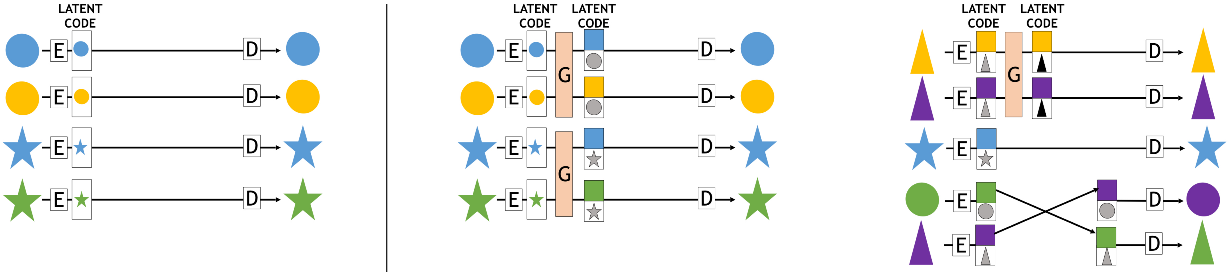
(c) Objects grouped by color.

Figure 1: Shape and color are two factors of variation.

We propose *group-level supervision*: observations (or samples) are organised in groups, where within a group the observations share a common but unknown value for one of the factors of variation. For example, consider a data set of objects with two factors of variation: shape and color, as shown in Figure 1a. A possible grouping organises the objects by shape, as shown in Figure 1b. Another possible grouping organises the objects by color as in Figure 1c. Group supervision allows us to anchor the semantics of the data (shape and color) into the learned representation. Grouping is a form of weak supervision that is inexpensive to collect, and we do not assume that we know the factor of variation that defines the grouping.

Deep probabilistic generative models learn expressive representations of a given set of observations. Examples of such models include Generative Adversarial Networks (GAN) (Goodfellow et al. 2014) and the Variational Autoencoder (VAE) (Kingma and Welling 2014; Rezende, Mohamed, and Daan 2014). In the VAE model, an encoder network (the encoder) encodes an observation into its latent representation (or latent code) and a generative network (the decoder) decodes an observation from a latent code. The VAE model allows efficient test-time inference by using amortised inference, that is, the observations parametrise the posterior distribution of the latent code, and all observations share a single set of parameters to learn. However, the VAE model assumes that the observations are independent and identically distributed (iid). In the case of grouped observations, this assumption is no longer true. Consider again the toy example of the objects data set in Figure 1, and assume that the objects are grouped by shape. The VAE model processes each observation independently and takes no ad-

[*] This work was done while Diane was an intern at Microsoft Research.



(a) Original VAE assumes iid observations.

(b) ML-VAE at training.

(c) At test-time, ML-VAE generalises to unseen shapes and colors and allows control of the latent code.

Figure 2: In (a) the VAE model assumes iid observations. In comparison, (b) and (c) show our ML-VAE working at the group level. In (b) and (c) upper part of the latent code is color, lower part is shape. Black shapes show the ML-VAE accumulating evidence on the shape from the two grey shapes. E is the Encoder, D is the Decoder, G is the grouping operation.

vantage of the grouping information. This is shown in Figure 2a. How can we build a probabilistic model that easily incorporates the grouping information and learns the corresponding relevant representation?

We propose a model that retains the advantages of amortised inference while using the grouping information in a simple and flexible manner. We present the Multi-Level Variational Autoencoder (ML-VAE), a new deep probabilistic model that learns a disentangled representation of a set of grouped observations. The ML-VAE separates the latent representation (or latent code) into semantically meaningful parts by working both at the group level and the observation level. Without loss of generality we assume that there are two latent factors of variation, *style* and *content*. The content is common for a group, while the style can differ within the group. We emphasise that our approach is general in that there can be more than two factors. Moreover, multiple groupings of the same data set, along different factors of variation, are possible. To process grouped observations, the ML-VAE uses a grouping operation that separates the latent code into two parts, style and content, and observations in the same group have the same content. This latent code separation is a design choice. This is illustrated in Figure 2b. For illustrative purposes, the upper part of the latent code represents the style (color) and the lower part the content (shape). Recall that we consider the objects grouped by shape. In Figure 2b, after the grouping operation the two circles share the same shape in the lower part of the latent code (corresponding to content). The variations within the group (style), in this case color, get naturally encoded in the upper part. Importantly, the ML-VAE does not need to know that the objects are grouped by shape nor what shape and color represent; the only supervision at training is the organisation of the data into groups. The grouping operation makes the encoder learn a semantically meaningful disentanglement. Once trained the ML-VAE encoder is able to disentangle observations even without grouping information, for example the single blue star in Figure 2c. If samples are grouped the grouping operation increases the certainty on the content: in Figure 2c black triangles show that the model has accumulated evidence of the content (triangle) from the two disentangled codes (grey triangles). The ML-VAE generalises to unseen realisations of the factors of variation, for example a purple triangle, and we can manipulate the latent code to perform operations such as swapping the style to generate new observations, as shown in Figure 2c.

To sum-up, our contributions are (i) We propose the ML-VAE model to learn a disentangled and controllable representations from grouped data; (ii) We extend amortised inference to the case of non-iid observations; (iii) We experimentally show that the ML-VAE model learns a semantically meaningful disentanglement of grouped data, enables manipulation of the latent representation, and generalises to unseen groups.

2 Related work

Unsupervised and semi-supervised settings In the unsupervised setting, the Generative Adversarial Networks (GAN) (Goodfellow et al. 2014) and Variational Autoencoder (VAE) (Kingma and Welling 2014; Rezende, Mohamed, and Daan 2014) models have been extended to the learning of an interpretable representation (Chen et al. 2016; Wang and Gupta 2016; Higgins et al. 2017; Abbasnejad, Dick, and van den Hengel 2016). As they are unsupervised, these models do not anchor a specific meaning into the disentanglement. In the semi-supervised setting, the VAE model has been extended to the learning of a disentangled representation by introducing a semi-supervised variable, either discrete (Kingma et al. 2014) or continuous (Siddharth et al. 2017). Also in the semi-supervised context, Makhzani et al. (2015) and Mathieu et al. (2016) propose adversarially trained autoencoders to learn disentangled representations. However, semi-supervised models require the semi-supervised variable to be observed on a limited number of input points. The VAE model has also been applied to the learning of representations that are invariant to a certain source of variation (Alemi et al. 2017; Louizos et al. 2016; Edwards and Storkey 2016; Chen et al. 2017). As in the semi-supervised case, these models require supervision on the source of variation to be invariant to. Consider the data set of objects, grouped by shape as in Figure 1b, and assume that the training set contains only 2 shapes: circle and star. Semi-supervised models using a discrete variable would have to fix its dimension, denoted K, for example taking $K = 2$ the number of training shapes. This does not allow to have an unbounded number of shapes and unseen shapes such as a triangle at test-time. Semi-supervised models with a continuous latent variable would choose an arbitrary fixed way to construct training labels from grouped data, for example per-shape statistics. At test-time, the unseen triangle shape would be encoded as a mixture of the training shapes: circle and star.

By contrast, we address the setting in which training samples are grouped. A grouping is different from a label because test samples generally do not belong to any of the groups seen during training.

Interpretable representation of grouped data While not directly applied to interpretable representation learning, Murali, Chaudhuri, and Jermaine (2017) perform computer program synthesis from grouped user-supplied example programs, and Allamanis et al. (2017) learn semantic representations of mathematical and logical expressions grouped in equivalence classes. To perform 3D rendering of objects, Kulkarni et al. (2015) enforce a disentangled representation by using training batches where only one factor of variation varies. However, this requires to be able to fix each factor of variation. Multiple works perform image-to-image translation between two unpaired images sets using adversarial training (Zhu et al. 2017; Kim et al. 2017; Yi et al. 2017; Fu et al. 2017; Taigman, Polyak, and Wolf 2017; Shrivastava et al. 2017; Bousmalis et al. 2017; Liu, Breuel, and Kautz 2017). Two images sets can be seen as two groups of images, grouped by image type. Donahue et al. (2017) disentangles the latent space of GAN using images grouped by identity, and Denton and Birodkar (2017) and Tulyakov et al. (2017) learn disentangled representations of videos with adversarial training. A video can be seen as a group of images with common content (identity) and various styles (background). In contrast to these methods, we do not require adversarial networks. Moreover, it is unclear how to extend the cited models to other types of data, more than two groups, and several groupings (along multiple factors of variation) of the same data set. We also relate group supervision to the case of triplets annotations (Veit, Belongie, and Karaletsos 2017; Karaletsos, Belongie, and Rätsch 2016; Tian, Chen, and Zhu 2017). A triplet is an ordering on three oberved data *a,b,c* of the form "*a* is more similar to *b* than *c*". Karaletsos, Belongie, and Rätsch (2016) learn a latent representation jointly from observations and triplets.

The *neural statistician* (Edwards and Storkey 2017) computes representations of *datasets*, where samples in the same dataset share a common *context* latent variable. Statistics of a dataset, such as its average, are fed to a network that outputs the parameters of the posterior of the context. Their concept of dataset can be seen as a group, and the context latent variable would be the content. Our work differs from theirs as we explicitly build the content posterior distribution from the codes of the observations in the group, as detailed in section 3.2. Moreover, we want to learn a disentangled and controllable latent representation. Thereby, we model samples within a group to have a shared group content variable and an independent style variable, with style and content independent given the observation.

In order to learn a *disentangled* and *controllable* representation of grouped data, we propose the Multi-Level Variational Autoencoder (ML-VAE).

3 Model

Random variables are denoted in bold, and their values are denoted in non-bold. We assume that the variable $\boldsymbol{x}$ is generated by a latent variable $\boldsymbol{z}$ via the distribution $p(\boldsymbol{x}|\boldsymbol{z};\theta)$. We consider a data set of n observations $\mathcal{D} = \{x_1, \ldots, x_n\}$. The goal is to infer the values of the latent variable that generated the observations, that is, to compute the posterior distribution $p(\boldsymbol{z}|x;\theta)$, which is often intractable.

3.1 Amortised inference with VAE

The Variational Autoencoder (VAE) model (Kingma and Welling 2014; Rezende, Mohamed, and Daan 2014) approximates $p(\boldsymbol{z}|\boldsymbol{x};\theta)$ with the variational distribution $q(\boldsymbol{z}|\boldsymbol{x};\phi)$, where ϕ are the variational parameters, and maximises a lower-bound on the average marginal log-likelihood (or evidence). Contrary to Stochastic Variational Inference (SVI) (Hoffman et al. 2013), the VAE model performs amortised variational inference, that is, the observations parametrise the posterior distribution of the latent code, and all observations share a single set of parameters ϕ. This allows efficient test-time inference. Figures 3a and 3b shows the SVI and VAE graphical models, we highlight in red that SVI does not perform amortised inference.

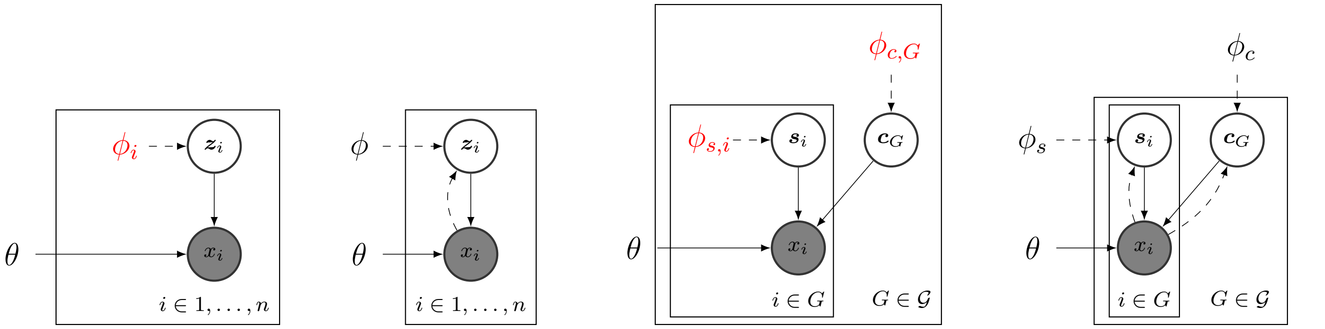
However, the VAE model assumes independent, identically distributed (iid) observed variables. Therefore, the VAE model does not leverage the grouping information. In this context, the question is how to perform amortised inference in the context of non-iid, grouped observations?

3.2 The Multi-Level VAE for grouped data

In the grouped data setting, the observations are organised in a set $\mathcal{G}$ of distinct groups, with a factor of variation that is shared among all observations within a group. The grouping forms a partition of $1, \ldots, n$, i.e. each group $G \in \mathcal{G}$ is a subset of $1, \ldots, n$ of arbitary size, disjoint of all other groups.

Without loss of generality, we separate the latent representation in two latent variables $\boldsymbol{z} = (\boldsymbol{c}, \boldsymbol{s})$ with *style* $\boldsymbol{s}$ and *content* $\boldsymbol{c}$. The content is the factor of variation along which the groups are formed. In this context, referred as the grouped observations setting, the latent representation has a single content latent variable $\boldsymbol{c}_G$ per group. SVI can be adapted by enforcing that all observations within a group share a single content latent variable while the style remains untied, see Figure 3c. However, SVI does not use amortised inference and requires expensive test-time inference. Experimentally, it also needs more training epochs as we show in the supplemental.

We denote by $\boldsymbol{X}_G = (\boldsymbol{x}_i, \forall i \in G)$ the collection of $\boldsymbol{x}_i$ variables of a group G. We do not assume iid observations,



(a) SVI for iid observations.

(b) VAE for iid observations.

(c) SVI for non-iid, grouped observations.

(d) Our ML-VAE for non-iid, grouped observations.

Figure 3: SVI, VAE and our ML-VAE graphical models. Solid lines denote the generative model, dashed lines denote the variational approximation. Shaded nodes indicate that the variables $\boldsymbol{x}_i$ have been set to their observed value x_i.

but independence at the grouped observations level. The average marginal log-likelihood (or evidence) decomposes over the groups:

$$\frac{1}{|\mathcal{G}|} \log p(\mathcal{D};\theta) = \frac{1}{|\mathcal{G}|} \sum_{G\in\mathcal{G}} \log p(X_G;\theta). \tag{1}$$

By comparison, the VAE model decomposes the evidence on the samples $x_1,\dots,x_n$. We model each $\boldsymbol{x}_i$ in $\boldsymbol{X}_G$ to have its independent latent code for the style $\boldsymbol{s}_i$, and $\boldsymbol{S}_G = (\boldsymbol{s}_i, \forall i \in G)$ is the collection of style latent variables for the group G. By constrast, we model a single content latent code $\boldsymbol{c}_G$ shared among all $\boldsymbol{x}_i$ in $\boldsymbol{X}_G$. We approximate the true posterior $p(\boldsymbol{c}_G, \boldsymbol{S}_G|X_G;\theta)$ with a variational posterior $q(\boldsymbol{c}_G, \boldsymbol{S}_G|X_G;\phi)$ that decomposes as the product of $q(\boldsymbol{c}_G|X_G;\phi_c)$ and $q(\boldsymbol{S}_G|X_G;\phi_s)$, with ϕ_c and ϕ_s the variational parameters for content and style respectively. We design the approximating variational posterior $q(\boldsymbol{S}_G|X_G;\phi_s)$ such that it factorises among the samples in a group as $\prod_{i\in G} q(\boldsymbol{s}_i|x_i;\phi_s)$. Given the style and content, the observed variables in a group are independent and $p(\boldsymbol{X}_G|c_G, S_G;\theta)$ also factorises. This results in the graphical model shown Figure 3d. For each group G, we can write its evidence as the sum of the Kullback-Leibler divergence between the true posterior and the variational approximation, and $\mathcal{L}(X_G;\theta,\phi_c,\phi_s)$, referred as the Group Evidence Lower Bound (Group ELBO):

$$\begin{aligned}\log p(X_G;\theta) &= \mathcal{L}(X_G;\theta,\phi_c,\phi_s)\\ &\quad + \mathrm{KL}(q(\boldsymbol{c}_G, \boldsymbol{S}_G|X_G;\phi_c,\phi_s)||p(\boldsymbol{c}_G, \boldsymbol{S}_G|X_G;\theta)),\\ &\geq \mathcal{L}(X_G;\theta,\phi_c,\phi_s).\end{aligned} \tag{2}$$

since the Kullback-Leibler divergence (KL) is always positive. The Group ELBO is expressed as,

$$\begin{aligned}&\mathcal{L}(X_G;\theta,\phi_c,\phi_s)\\ &= \mathbb{E}_{q(\boldsymbol{c}_G,\boldsymbol{S}_G|X_G;\phi_c,\phi_s)}[\log p(X_G|\boldsymbol{c}_G,\boldsymbol{S}_G;\theta)]\\ &\quad - \mathrm{KL}(q(\boldsymbol{c}_G,\boldsymbol{S}_G|X_G;\phi_c,\phi_s)||p(\boldsymbol{c}_G,\boldsymbol{S}_G))\\ &= \sum_{i\in G} \mathbb{E}_{q(\boldsymbol{c}_G|X_G;\phi_c)}\left[\mathbb{E}_{q(\boldsymbol{s}_i|x_i;\phi_s)}[\log p(x_i|\boldsymbol{c}_G,\boldsymbol{s}_i;\theta)]\right]\\ &\quad - \sum_{i\in G} \mathrm{KL}(q(\boldsymbol{s}_i|x_i;\phi_s)||p(\boldsymbol{s}_i)) - \mathrm{KL}(q(\boldsymbol{c}_G|X_G;\phi_c)||p(\boldsymbol{c}_G)).\end{aligned} \tag{3}$$

Note that we have a single KL term for the group content $\boldsymbol{c}_G$. We learn the model's parameters by maximising the average Group ELBO, that is,

$$\mathcal{L}(\mathcal{D},\phi_c,\phi_s,\theta) := \frac{1}{|\mathcal{G}|}\sum_{G\in\mathcal{G}} \mathcal{L}(X_G;\theta,\phi_c,\phi_s). \tag{4}$$

It is a lower bound on the data set average evidence (1) because each Group ELBO $\mathcal{L}(X_G;\theta,\phi_c,\phi_s)$ is a lower bound on $\log p(X_G;\theta)$. In practise, we use mini-batches $\mathcal{G}_b$ of groups, as follows,

$$\mathcal{L}(\mathcal{G}_b,\theta,\phi_c,\phi_s) := \frac{1}{|\mathcal{G}_b|}\sum_{G\in\mathcal{G}_b} \mathcal{L}(X_G;\theta,\phi_c,\phi_s). \tag{5}$$

If we take each group $G \in \mathcal{G}_b$ in its entirety it is an unbiased estimator of $\mathcal{L}(\mathcal{D},\phi_c,\phi_s,\theta)$. If the groups' sizes are too large to fit into memory we subsample G, resulting in a bias discussed in the supplemental. Our training algorithm is shown in Algorithm 1. Note that in step 3 of Algorithm 1 we sample one content $c_{G,i}$ per observation in the group, but c_G can be sampled once and used for all the samples in a group[1].

3.3 Accumulating group evidence

For each group G, in step 2 of Algorithm 1, we build the group content posterior distribution $q(\boldsymbol{c}_G|X_G;\phi_c)$ by accumulating information from the result of encoding each sample in G. How can we accumulate the information in a relevant manner to compute the group content distribution?

Our idea is to explicitly build the group content posterior distribution $q(\boldsymbol{c}_G|X_G;\phi_c)$ from the encodings of the grouped observations $X_G = (x_i, \forall i \in G)$. While any distribution could be employed, we focus on using a product of Normal density functions, which can be seen as an instance of Structured Variational Autoencoders (SVAE) (Johnson et al. 2016). Other possibilities, such as a mixture of density functions, are discussed in the supplemental. Specifically,

[1] We experimentally tried this method which resulted in similar performances. We attribute this to the fact that the variances of the content distribution tend to be very small.

Algorithm 1: ML-VAE training algorithm.

```
for $t$=1,...,$T$ epochs do
    Sample mini-batch of groups $\mathcal{G}_b$.
    for $G \in \mathcal{G}_b$ do
        for $i \in G$ do
            Encode $x_i$ into $q(\boldsymbol{c}_G|x_i;\phi_c^t)$, $q(\boldsymbol{s}_i|x_i;\phi_s^t)$.
        end
        Construct $q(\boldsymbol{c}_G|X_G;\phi_c^t)$
          from $q(\boldsymbol{c}_G|x_i;\phi_c^t), \forall i \in G$.
        for $i \in G$ do
            Sample $c_{G,i} \sim q(\boldsymbol{c}_G|X_G;\phi_c^t)$.
            Sample $s_i \sim q(\boldsymbol{s}_i|x_i;\phi_s^t)$.
            Decode $c_{G,i}, s_i$ into $p(\boldsymbol{x}_i|c_{G,i}, s_i;\theta^t)$.
        end
    end
    Update $\theta^{t+1}, \phi_c^{t+1}, \phi_s^{t+1} \leftarrow \theta^t, \phi_c^t, \phi_s^t$ by ascending
      the gradient estimate of $\mathcal{L}(\mathcal{G}_b, \theta, \phi_c, \phi_s)$.
end
```

we construct the probability density function of the posterior of the content variable $\boldsymbol{c}_G$ by multiplying $|G|$ Normal density functions, each of them evaluating the probability of $\boldsymbol{c}_G = c_G$, given the observation $\boldsymbol{x}_i = x_i, \forall i \in G$:

$$q(\boldsymbol{c}_G = c_G|\boldsymbol{X}_G = X_G;\phi_c) \propto \prod_{i \in G} q(\boldsymbol{c}_G = c_G|\boldsymbol{x}_i = x_i;\phi_c), \tag{6}$$

where we assume $q(\boldsymbol{c}_G|\boldsymbol{x}_i = x_i;\phi_c)$ to be a Normal distribution $\mathcal{N}(\mu_i, \Sigma_i)$. The normalisation constant is the resulting product marginalised over all possible values of $\boldsymbol{c}_G$. The resulting density function $q(\boldsymbol{c}_G|X_G;\phi_c)$ is the density function of a Normal distribution of mean μ_G and variance Σ_G, expressed as follows (derivations are in the supplemental),

$$\mu_G^T\Sigma_G^{-1} = \sum_{i \in G} \mu_i^T\Sigma_i^{-1},\ \Sigma_G^{-1} = \sum_{i \in G} \Sigma_i^{-1}. \tag{7}$$

It is interesting to note that the variance of the resulting Normal distribution, Σ_G, is inversely proportional to the sum of the group's observations inverse variances. Therefore, we expect that by increasing the number of observations in a group, the variance of the resulting distribution decreases. This is what we refer as "accumulating evidence". We empirically investigate this effect in section 4. Since the resulting distribution is a Normal distribution, the term $\mathrm{KL}(q(\boldsymbol{c}_G|X_G;\phi_c)||p(\boldsymbol{c}_G))$ can be evaluated in closed-form. We also assume a Normal distribution for $q(\boldsymbol{s}_i|x_i;\phi_s)$, $\forall i \in G$.

4 Experiments

Our goal with the experiments is two-fold. First, we want to evaluate the performance of ML-VAE to learn a semantically meaningful disentangled representation. Second, we want to explore the impact of "accumulating evidence" at test-time. To do so, when we encode test images we employ two possible strategies: (i) strategy 1 is no grouping information on the test samples, each test image is a group; (ii) strategy 2 takes into account the grouping information and uses multiple test images per group to construct the content latent code with the product of Normal densities method.

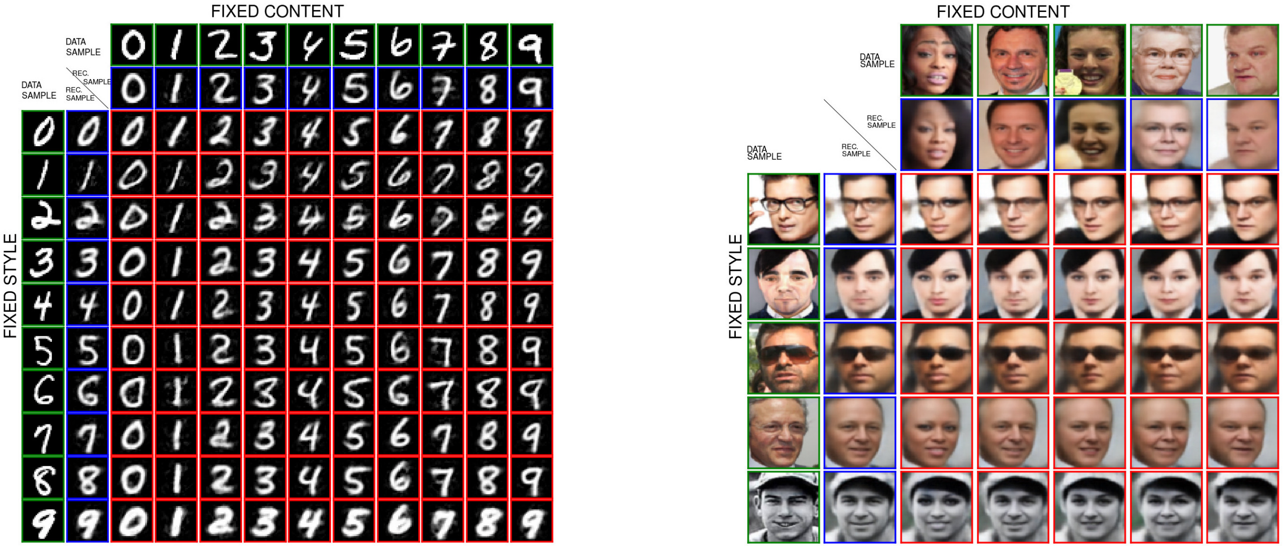
Similar to Mathieu et al. (2016), we propose qualitative and quantitative evaluations. We do not show qualitative results of the original VAE model as there is no objective choice on which part of its code is style or content. However, we perform quantitative comparison with the VAE, to compare with a variational model that does not leverage grouping information at training. Encoder architectures, additional results and training details are in the supplemental.

MNIST data set. We perform evaluation on MNIST (LeCun et al. 1998). We consider the data grouped by digit. We randomly separate the $60,000$ training examples into $50,000$ training samples and $10,000$ validation samples, and use the standard MNIST testing data set. The style and content vectors are of size 10 each. The decoder network is composed a linear layer with 500 hidden units with the hyperbolic tangent activation function. It is followed by two linear layers of 784 hidden units each that output respectively the mean and log-variance of $p(\boldsymbol{x}_i|c_{G,i}, s_i;\theta)$.

MS-Celeb-1M data set. Next, we perform evaluation on the face aligned version of the MS-Celeb-1M data set (Guo et al. 2016). The data set was constructed by retrieving approximately 100 images per celebrity from popular search engines. We group the data by identity. For each query, we consider the top ten results. There were multiple queries per celebrity so identities can have more than 10 images. Importantly, we randomly separate the resulting data set in disjoints sets of identities as the training ($48,880$ identities, $401,406$ images), validation ($25,000$ identities, $205,015$ images) and testing ($25,000$ identities, $205,371$ images) data sets. This way we evaluate the ability of ML-VAE level to generalise to unseen groups (unseen identities) at test-time.

The style and content vectors are of size 50 each. The decoder network is composed of 3 deconvolutional layers (stride 2, kernel size 4) of respectively $256, 128, 64$ filters, each followed by Batch Normalisation and Rectified Linear Units. These are followed by two deconvolutional layers (stride 1, kernel size 3) of 3 filters that output respectively the mean and log-variance of $p(\boldsymbol{x}_i|c_{G,i}, s_i;\theta)$. The layer for the log-variance is followed by the tangent hyperbolic activation function, multiplied by 5.

Qualitative Evaluation. We qualitatively assess the relevance of the learned representation by performing operations on the latent code. First we perform *swapping*. We encode test images, draw a sample per image from its style and content latent codes, and swap the style between images. Second we perform *interpolation*. We encode a pair of test images, draw one sample from each image style and content latent codes, and linearly interpolate between them. We present the results of swapping and interpolation with accumulating evidence of up to 10 images that belong to the same group (strategy 2). Results using strategy 1 (in supplemental) are also convincing and show the ML-VAE's abil-



(a) MNIST, test dataset.

(b) MS-Celeb-1M, test dataset.

Figure 4: Swapping, first row and first column are test samples (green boxes), second row and column are reconstructed samples (blue boxes) the rest are swapped reconstructed samples (red boxes). Each row is a fixed style, each column is a fixed content.

ity to disentangle without grouping information. Recall that these are test-time strategies, at training the ML-VAE accumulates evidence. Figures 4a and 4b show the swapping results, where the first row and the first column show the test data samples input to ML-VAE (green boxes), the second row and column are reconstructed samples (blue boxes). In the remaining rows and columns, each row is a fixed style and each column is a fixed content (red boxes). Looking at each column in Figure 4b, we see that the model encodes the factor of variation that grouped the data, that is the identity, into the facial traits. Indeed, when style gets transferred, the facial traits remain consistent along each column. The model encodes the remaining factors (for example background, face orientation, sunglasses) into the style latent code. This shows that the ML-VAE learns a disentangled and controllable representation of the data that anchors the semantics of the grouping. The model learns this meaningful disentanglement without knowing that the data is grouped by identity, nor what *is* identity, but only using the organisation of the data into groups. Similarly, Figure 4a shows that the ML-VAE encodes the digit label into the content. Moreover, we see that the ML-VAE generalises to unseen groups, as for MS-Celeb-1M training and testing identities are disjoints.

Figure 5 shows the results of the interpolation task. From top left to bottom right, rows correspond to a fixed style and interpolating on the content, columns correspond to a fixed content and interpolating on the style. We see that the identity, in the form of facial traits, remains consistent along the column, while we linearly interpolate the style. If we look along each line, the style remain consistent and the identity smoothly varies as we interpolate on the content.

Third, we perform *generation*. We build the content latent code by accumulating images of a given identity. We take the mean of the resulting content distribution and generate images with multiple styles drawn from the prior. Figure 6a shows the results. We see that the facial traits remain consistent in the generated images, and different styles gives different head orientation, moustache/no moustache, etc. This emphasises on the disentanglement power of the model and highlight that it covers the data manifold. Finally, in Figure 6b, we reconstruct digits of the same label with and without using the grouping information (strategies 1 and 2). The ML-VAE corrects inference (wrong digit label in first row and second column) by accumulating evidence.

Quantitative Evaluation. In order to quantitatively evaluate the disentanglement power of our model, we use the style la-

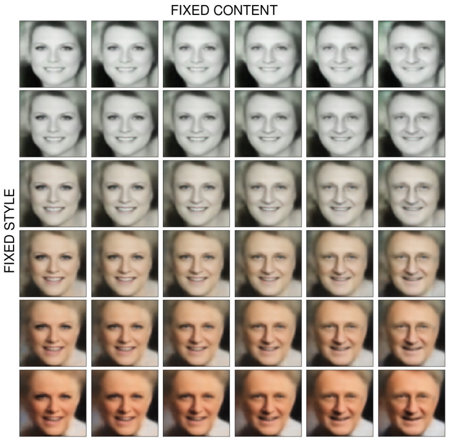
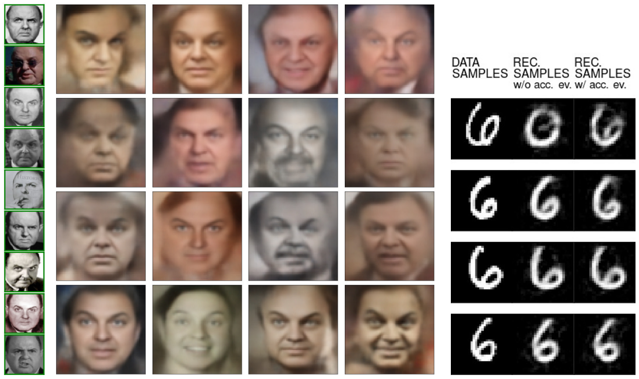


Figure 5: Interpolation, from top left to bottom right, rows show a fixed style and interpolating the content, columns show a fixed content and interpolating the style.



(a) Generation.

(b) Accumulating evidence.

Figure 6: In (a): generation. Style is sampled from the prior and the content is computed using the test images for this identity (green boxes on the left). In (b): accumulating evidence. Left column are test samples, middle column are reconstructed samples without accumulating evidence (w/o acc. ev.), right column are reconstructed samples with accumulating evidence (w/ acc. ev.) , that is using the four digits images to build the content code).

tent code and content latent code as features for a classification task. We denote by $\boldsymbol{y}$ the random variable representing the class, and by G_y a group of observations from the same class. The quality of the disentanglement is high if the content latent variable $\boldsymbol{c}_{G_y}$ is informative about the class, while the style latent variable $\boldsymbol{S}_{G_y}$ is not. In the case of MNIST the class is the digit label and for MS-Celeb-1M the class is the identity. We emphasise that in the case of MS-Celeb-1M test images are all unseen classes (unseen identities) at training. We learn to classify the test images with a neural network classifier once using $\boldsymbol{S}_{G_y}$ and once using $\boldsymbol{c}_{G_y}$ as input features. We also compare to using the original VAE model full latent code as features. In this case, we also accumulate evidence with the product of Normal densities method for samples of the same class to construct the features from the VAE code.

Let us take the example of the latent code $\boldsymbol{c}_{G_y}$ used as features. We train the neural network classifier to learn a distribution $r(\boldsymbol{y}|\boldsymbol{c}_{G_y})$ by minimising the cross-entropy loss $-\mathbb{E}_{p(\boldsymbol{y},\boldsymbol{c}_{G_y})}\left[\log r(\boldsymbol{y}|\boldsymbol{c}_{G_y})\right]$. Thereby, we minimise an upper bound on $\mathbb{H}(\boldsymbol{y}|\boldsymbol{c}_{G_y})$ the conditional entropy of the class given the latent code. Indeed, we can upper bound $\mathbb{H}(\boldsymbol{y}|\boldsymbol{c}_{G_y})$ as follows (detailed in the supplemental),

$$\mathbb{H}(\boldsymbol{y}|\boldsymbol{c}_{G_y}) \leq -\mathbb{E}_{p(\boldsymbol{y},\boldsymbol{c}_{G_y})}\left[\log r(\boldsymbol{y}|\boldsymbol{c}_{G_y})\right]. \tag{8}$$

We report the classifier test accuracy, and the value of $-\mathbb{E}_{p(\boldsymbol{y},\boldsymbol{c}_{G_y})}\left[\log r(\boldsymbol{y}|\boldsymbol{c}_{G_y})\right]$ as the conditional entropy in bits on the classifier testing set. Similarly, we report performance using the ML-VAE style latent code, and the VAE model full latent code. We explore the benefits of accumulating evidence: (i) for training the classifier, we construct the posterior distribution of the content by accumulating K images per class (ii) for testing the classifier, we use only $k \leq K$ images per class, where $k = 1$ is no grouping information. When k increases we expect the performance of the classifier trained on $\boldsymbol{c}_{G_y}$ to improve as the features become more informative. We expect the performance using the style $\boldsymbol{S}_{G_y}$ to remain constant. The results are shown in Figure 7. We see that for small values of k, the ML-VAE content latent code is more informative about the class than VAE latent code, especially on MNIST. When k increases this shows the benefit of accumulating evidence. Recall that we also accumulate evidence, for samples of the same class, to construct the features from the original VAE latent code. The ML-VAE also provides a relevant disentanglement as the style remains uninformative about the class.

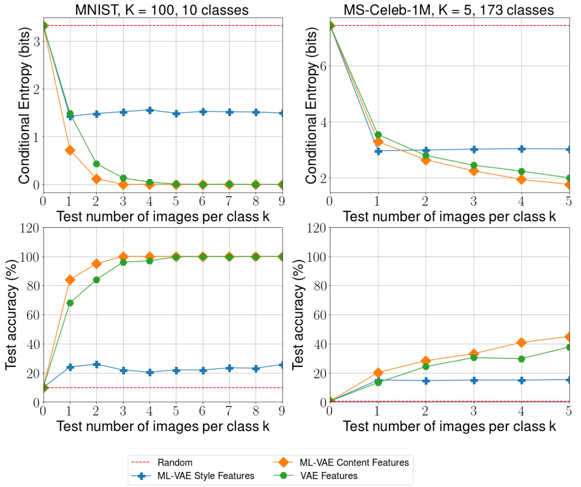


Figure 7: Accuracy (higher is better) and conditional entropy (lower is better). For clarity on MNIST we show up to $k = 10$. Values stay stationary for larger k (in supplemental.)

5 Discussion

We proposed the Multi-Level VAE model for learning a meaningful disentanglement from a set of grouped observations. The ML-VAE model handles an arbitrary number of groups of observations, which needs not be the same at training and testing. We proposed different methods for incorporating the semantics embedded in the grouping. Experimental evaluations show the relevance of our method, as the ML-VAE learns a semantically meaningful disentangled representation, generalises to unseen groups and enables control on the latent representation. For future work, we wish to apply the ML-VAE to text data.

6 Acknowledgments

This work is co-funded by the Microsoft Research PhD Scholarship Programme in EMEA and the University of Oxford. We thank Stephan Garbin for his insightful comments.

References

Abbasnejad, E.; Dick, A. R.; and van den Hengel, A. 2016. Infinite variational autoencoder for semi-supervised learning. *arxiv:1611.07800*.

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. P. 2017. Deep variational information bottleneck. *ICLR*.

Allamanis, M.; Chanthirasegaran, P.; Kohli, P.; and Sutton, C. 2017. Learning continuous semantic representations of symbolic expressions. *ICML*.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8):1798–1828.

Bousmalis, K.; Silberman, N.; Dohan, D.; Erhan, D.; and Dilip Krishnan, D. 2017. Unsupervised pixel-level domain adaptation with generative adversarial networks. *CVPR*.

Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. *NIPS*.

Chen, X.; Kingma, D. P.; Salimans, T.; Duan, Y.; Dhariwal, P.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2017. Variational lossy autoencoder. *ICLR*.

Denton, E., and Birodkar, V. 2017. Unsupervised learning of disentangled representations from video. *arxiv:1705.10915*.

Donahue, C.; Balsubramani, A.; McAuley, J.; and Lipton, Z. C. 2017. Semantically decomposing the latent spaces of generative adversarial networks. *arxiv:1705.07904*.

Edwards, H., and Storkey, A. J. 2016. Censoring representations with an adversary. *ICLR*.

Edwards, H., and Storkey, A. J. 2017. Towards a neural statistician. *ICLR*.

Fu, T.-C.; Liu, Y.-C.; Chiu, W.-C.; Wang, S.-D.; and Wang, Y.-C. F. 2017. Learning cross-domain disentangled deep representation with supervision from a single domain. *arxiv:1705.01314*.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *NIPS*.

Guo, Y.; Zhang, L.; Hu, Y.; He, X.; and Gao, J. 2016. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. *ECCV*.

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

Hoffman, M. D.; Blei, D. M.; Wang, C.; and Paisley, J. 2013. Stochastic variational inference. *J. Machine Learning Research*.

Johnson, M. J.; Duvenaud, D.; Wiltschko, A. B.; Datta, S. R.; and Adams, R. P. 2016. Composing graphical models with neural networks for structured representations and fast inference. *NIPS*.

Karaletsos, T.; Belongie, S.; and Rätsch, G. 2016. Bayesian representation learning with oracle constraints. *ICLR*.

Kim, T.; Cha, M.; Kim, H.; Lee, J. K.; and Kim, J. 2017. Learning to discover cross-domain relations with generative adversarial networks. *ICML*.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational Bayes. *ICLR*.

Kingma, D. P.; Rezende, D. J.; Mohamed, S.; and Welling, M. 2014. Semi-supervised learning with deep generative models. *NIPS*.

Kulkarni, T. D.; Whitney, W.; Kohli, P.; and Tenenbaum, J. B. 2015. Deep convolutional inverse graphics network. *NIPS*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *IEEE* 86(11):2278–2324.

Linsker, R. 1988. Self-organization in a perceptual network. *Computer* 21(3):105–117.

Liu, M.; Breuel, T.; and Kautz, J. 2017. Unsupervised image-to-image translation networks. *arxiv:1703.00848*.

Louizos, C.; Swersky, K.; Li, Y.; Welling, M.; and Zemel, R. 2016. The variational fair autoencoder. *ICLR*.

Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I. J.; and Frey, B. 2015. Adversarial autoencoders. *ICLR Workshop*.

Mathieu, M. F.; Zhao, J. J.; Zhao, J.; Ramesh, A.; Sprechmann, P.; and LeCun, Y. 2016. Disentangling factors of variation in deep representation using adversarial training. *NIPS*.

Murali, V.; Chaudhuri, S.; and Jermaine, C. 2017. Bayesian sketch learning for program synthesis. *arxiv:1703.05698v2*.

Rezende, D. J.; Mohamed, S.; and Daan, W. 2014. Stochastic backpropagation and variational inference in deep generative models. *arxiv:1401.4082v3*.

Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; and Webb, R. 2017. Learning from simulated and unsupervised images through adversarial training. *CVPR*.

Siddharth, N.; Paige, B.; ; Van de Meent, J.-W.; Desmaison, A.; Wood, F.; Goodman, N. D.; Kohli, P.; and Torr, P. H. 2017. Learning disentangled representations with semi-supervised deep generative models. *arXiv:1706.00400*.

Taigman, Y.; Polyak, A.; and Wolf, L. 2017. Unsupervised cross-domain image generation. *ICLR*.

Tian, T.; Chen, N.; and Zhu, J. 2017. Learning attributes from the crowdsourced relative labels. *AAAI*.

Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2017. MoCoGAN: Decomposing motion and content for video generation. *CVPR*.

Veit, A.; Belongie, S.; and Karaletsos, T. 2017. Conditional similarity networks. *CVPR*.

Wang, X., and Gupta, A. 2016. Generative image modeling using style and structure adversarial networks. *ECCV*.

Yi, Z.; Zhang, H.; Tan, P.; and Gong, M. 2017. DualGAN: Unsupervised dual learning for image-to-image translation. *arxiv:1704.02510*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*.