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 which decomposes an observation into factors of variation which we can independently control. Specifically, we want to use minimal supervision to learn a latent representation 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 collection of face images grouped by identity. We wish to anchor the semantics of the grouping into a relevant and disentangled representation that we can easily exploit. However, existing deep probabilistic models often assume that the observations are independent and identically distributed. We present the Multi-Level Variational Autoencoder (ML-VAE), a new deep probabilistic model for learning a disentangled representation of a set of grouped observations. The ML-VAE separates the latent representation into semantically meaningful parts by working both at the group level and the observation level, while retaining efficient test-time inference. Quantitative and qualitative evaluations show that the ML-VAE model (i) learns a semantically meaningful disentanglement of grouped data, (ii) enables manipulation of the latent representation, and (iii) generalises to unseen groups.

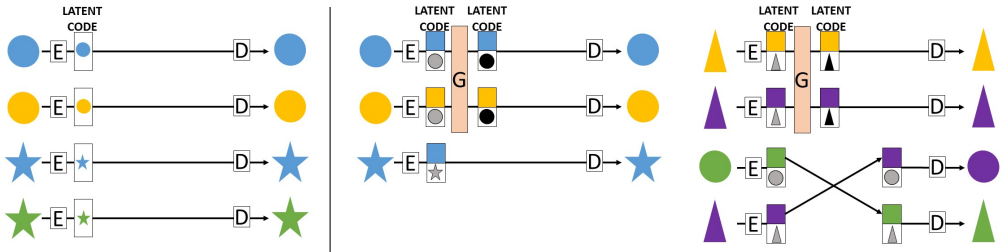
1 Introduction

Representation learning refers to the task of learning a representation of the data that can be easily exploited, see Bengio et al. [2013]. In this work, 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*, described in Linsker [1988], states that an optimal representation is one of bounded entropy which retains as much information about the data as possible. However, we are interested in learning a semantically meaningful disentanglement of interesting latent factors. How can we anchor semantics in high-dimensional representations?

We propose *group-level supervision*: observations 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, take images of circle and stars, of possible colors green, yellow and blue. A possible grouping organises the images by shape (circled or starred). Group observations allow us to anchor the semantics of the data (shape and color) into the learned representation. Group observations are a form of weak supervision that is inexpensive to collect. In the above shape example, we do not need to know the factor of variation that defines the grouping.

Deep probabilistic generative models learn expressive representations of a given set of observations. Among them, Kingma and Welling [2014], Rezende et al. [2014] proposed the very successful

*The work was performed as part of an internship at Microsoft Research.



(a) *Original VAE assumes i.i.d. observations.*

(b) *ML-VAE accumulates evidence.*

(c) *ML-VAE generalises to unseen shapes and colors and allows control on the latent code.*

Figure 1: *In (a) the VAE of Kingma and Welling [2014], Rezende et al. [2014], it assumes i.i.d. 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. Best viewed in color.*

Variational Autoencoder (VAE). In the VAE model, a 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 performs 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. This allows efficient test-time inference. However, the VAE model assumes that the observations are independent and identically distributed (i.i.d.). In the case of grouped observations, this assumption is no longer true. Considering the toy example of objects grouped by shape, the VAE model considers and processes each observation independently. This is shown in Figure 1a. The VAE model takes no advantage of the knowledge of the grouping.

How can we build a probabilistic model that easily incorporates this grouping information and learns the corresponding relevant representation? We could enforce equal representations within groups in a graphical model, using stochastic variational inference (SVI) for approximate posterior inference, Hoffman et al. [2013]. However, such model paired with SVI cannot take advantage of efficient amortised inference. As a result, SVI requires more passes over the training data and expensive test-time inference. Our proposed model retains the advantages of amortised inference while using the grouping information in a simple yet 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 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, *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, for the same set of observations, multiple groupings are possible along different factors of variation. To use group observations the ML-VAE uses a grouping operation that separates the latent representation into two parts, style and content, and samples in the same group have the same content. This in turns makes the encoder learn a semantically meaningful disentanglement. This process is shown in Figure 1b. For illustrative purposes, the upper part of the latent code represents the style (color) and the lower part the content (shape: circle or star). In Figure 1b, after being encoded 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, gets naturally encoded in the upper part. Moreover, while the ML-VAE handles the case of a single sample in a group, if there are multiples samples in a group the grouping operation increases the certainty on the content. This is shown in Figure 1b where black circles show that the model has accumulated evidence of the content (circle) from the two disentangled codes (grey circles). The grouping operation does not need to know that the data are grouped by shape nor what shape and color represent; the only supervision is the organisation of the data in groups. At test-time, the ML-VAE generalises to unseen realisations of the factors of variation, for example the purple triangle in Figure 1c. Using the disentangled representation, we can control the latent code and can perform operations such as swapping part of the latent representation to generate new observations, as shown in Figure 1c. To sum-up, our contributions are as follows.

- We propose the ML-VAE model to learn disentangled representations from group level supervision;

- we extend amortized inference to the case of non-iid observations;
- we demonstrate experimentally that the ML-VAE model learns a semantically meaningful disentanglement of grouped data;
- we demonstrate manipulation of the latent representation and generalises to unseen groups.

2 Related Work

Research has actively focused on the development of deep probabilistic models that learn to represent the distribution of the data. Such models parametrise the learned representation by a neural network. We distinguish between two types of deep probabilistic models. Implicit probabilistic models stochastically map an input random noise to a sample of the modelled distribution. Examples of implicit models include Generative Adversarial Networks (GANs) developed by Goodfellow et al. [2014] and kernel based models, see Li et al. [2015], Dziugaite et al. [2015], Bouchacourt et al. [2016]. The second type of model employs an explicit model distribution and builds on variational inference to learn its parameters. This is the case of the Variational Autoencoder (VAE) proposed by Kingma and Welling [2014], Rezende et al. [2014]. Both types of model have been extended to the representation learning framework, where the goal is to learn a representation that can be effectively employed. In the unsupervised setting, the InfoGAN model of Chen et al. [2016] adapts GANs to the learning of an interpretable representation with the use of mutual information theory, and Wang and Gupta [2016] use two sequentially connected GANs. The β-VAE model of Higgins et al. [2017] encourages the VAE model to optimally use its capacity by increasing the Kullback-Leibler term in the VAE objective. This favors the learning of a meaningful representation. Abbasnejad et al. [2016] uses an infinite mixture as variational approximation to improve performance on semi-supervised tasks. Contrary to our setting, these unsupervised models do not anchor a specific meaning into the disentanglement. In the semi-supervised setting, i.e. when an output label is partly available, Siddharth et al. [2017] learn a disentangled representation by introducing an auxiliary variable. While related to our work, this model defines a semi-supervised factor of variation. In the example of multi-class classification, it would not generalise to unseen classes. We define our model in the grouping supervision setting, therefore we can handle unseen classes at testing.

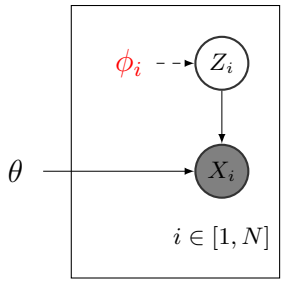
The VAE model has been extended to the learning of representations that are invariant to a certain source of variation. In this context Alemi et al. [2017] build a meaningful representation by using the Information Bottleneck (IB) principle, presented by Tishby et al. [1999]. The Variational Fair Autoencoder presented by Louizos et al. [2016] encourages independence between the latent representation and a sensitive factor with the use of a Maximum Mean Discrepancy (MMD) based regulariser, while Edwards and Storkey [2015] uses adversarial training. Finally, Chen et al. [2017] control which part of the data gets encoded by the encoder and employ an autoregressive architecture to model the part that is not encoded. While related to our work, these models require supervision on the source of variation to be invariant to. In the specific case of learning interpretable representation of images, Kulkarni et al. [2015] train an autoencoder with minibatch where only one latent factor changes. Finally, Mathieu et al. [2016] learn a representation invariant to a certain source of data by combining autoencoders trained in an adversarial manner.

Multiple works perform image-to-image translation between two unpaired images collections using GAN-based architectures, see Zhu et al. [2017], Kim et al. [2017], Yi et al. [2017], Fu et al. [2017], Taigman et al. [2017], Shrivastava et al. [2017], Bousmalis et al. [2016], while Liu et al. [2017] employ a combination of VAE and GANs. Interestingly, all these models require a form of weak supervision that is similar to our setting. We can think of the two unpaired images collections as two groups of observed data, sharing image type (painting versus photograph for example). Our work differs from theirs as we generalise to any type of data and number of groups. It is unclear how to extend the cited models to the setting of more than two groups and other types of data. Also, we do not employ multiple GANs models but a single VAE-type model. While not directly related to our work, Murali et al. [2017] perform computer program synthesis using grouped user-supplied example programs, and Allamanis et al. [2017] learn continuous semantic representations of mathematical and logical expressions. Finally we mention the concurrent recent work of Donahue et al. [2017] which disentangles the latent space of GANs.

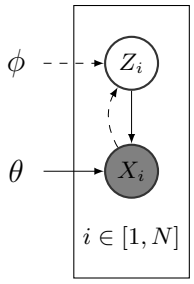
3 Model

3.1 Amortised Inference with the Variational Autoencoder (VAE) Model

We define $\mathbf{X} = (X_1, ..., X_N)$. In the probabilistic model framework, we assume that the observations $\mathbf{X}$ are generated by $\mathbf{Z}$, the unobserved (latent) variables. The goal is to infer the values of the



(a) *SVI graphical model.*



(b) *VAE graphical model.*

Figure 2: *VAE Kingma and Welling [2014], Rezende et al. [2014] and SVI Hoffman et al. [2013] graphical models. Solid lines denote the generative model, dashed lines denote the variational approximation.*

latent variable that generated the observations, that is, to calculate the posterior distribution over the latent variables $p(\mathbf{Z}|\mathbf{X};\theta)$, which is often intractable. The original VAE model proposed by Kingma and Welling [2014], Rezende et al. [2014] approximate the intractable posterior with the use of a variational approximation $q(\mathbf{Z}|\mathbf{X};\phi)$, where ϕ are the variational parameters. Contrary to Stochastic Variational Inference (SVI), 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. Figure 2 shows the SVI and VAE graphical models, we highlight in red that the SVI model does not take advantage of amortised inference.

3.2 The ML-VAE for Grouped Observations

We now assume that 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 $[1, N]$, i.e. each group $G \in \mathcal{G}$ is a subset of $[1, N]$ of arbitary size, disjoint of all other groups. Without loss of generality, we separate the latent representation in two latent variables $Z = (C, S)$ with *style* S and *content* C. The content is the factor of variation along which the groups are formed. In this context, referred as the grouped observations case, the latent representation has a single content latent variable per group C_G. SVI can easily be adapted by enforcing that all observations within a group share a single content latent variable while the style remains untied, see Figure 3a. However, employing SVI requires iterative test-time inference since it does not perform amortised inference. Experimentally, it also requires more passes on the training data as we show in the supplementary material. The VAE model assumes that the observations are i.i.d, therefore it does not take advantadge of the grouping. In this context, the question is how to perform amortised inference in the context of non-i.i.d., grouped observations? In order to tackle the aforementioned deficiency we propose the Multi-Level VAE (ML-VAE).

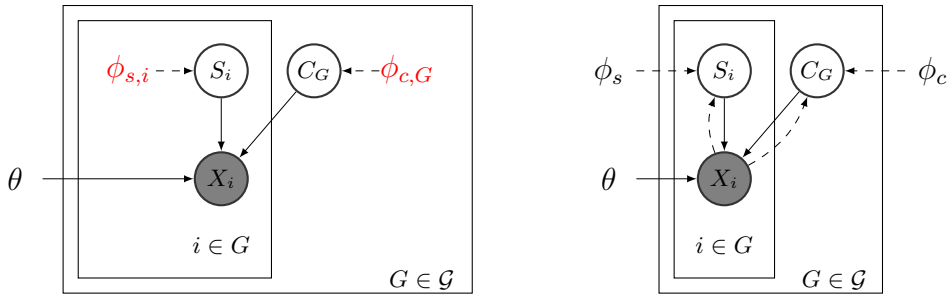
We denote by $\mathbf{X}_G$ the observations corresponding to the group G. We explicitly model each X_i in $\mathbf{X}_G$ to have its independent latent representation for the style S_i, and $\mathbf{S}_G = (S_i, i \in G)$. C_G is a unique latent variable shared among the group for the content. The variational approximation $q(C_G, \mathbf{S}_G|\mathbf{X}_G;\phi)$ factorises and ϕ_c and ϕ_s are the variational parameters for content and style respectively. We assume that the style is independent in a group, so $\mathbf{S}_G$ also factorises. Finally, given style and content, the likelihood $p(\mathbf{X}_G|C_G, \mathbf{S}_G;\theta)$ decomposes on the samples. This results in the graphical model shown Figure 3b.

We do not assume i.i.d. observations, but independence at the grouped observations level. The average marginal log-likelihood decomposes over groups of observations

$$\frac{1}{|\mathcal{G}|}\log p(\mathbf{X}|\theta) = \frac{1}{|\mathcal{G}|}\sum_{G\in\mathcal{G}}\log p(\mathbf{X}_G|\theta). \tag{1}$$

For each group, we can rewrite the marginal log-likelihood as the sum of the group Evidence Lower Bound $\text{ELBO}(G;\theta,\phi_s,\phi_c)$ and the Kullback-Leibler divergence between the true posterior $p(C_G, \mathbf{S}_G|\mathbf{X}_G;\theta)$ and the variational approximation $q(C_G, \mathbf{S}_G|\mathbf{X}_G;\phi_c)$. Since this Kullback-Leibler divergence is always positive, the first term, $\text{ELBO}(G;\theta,\phi_s,\phi_c)$, is a lower bound on the marginal log-likelihood,

$$\begin{aligned}\log p(\mathbf{X}_G|\theta) &= \text{ELBO}(G;\theta,\phi_s,\phi_c) + \text{KL}(q(C_G, \mathbf{S}_G|\mathbf{X}_G;\phi_c)||p(C_G, \mathbf{S}_G|\mathbf{X}_G;\theta)) \\ &\geq \text{ELBO}(G;\theta,\phi_s,\phi_c).\end{aligned} \tag{2}$$



(a) *SVI for grouped observations.* (b) *Our ML-VAE.*

Figure 3: *SVI Hoffman et al. [2013] and our ML-VAE graphical models. Solid lines denote the generative model, dashed lines denote the variational approximation.*

The ELBO$(G; \theta, \phi_s, \phi_c)$ for a group is

$$\begin{aligned}\text{ELBO}(G; \theta, \phi_s, \phi_c) = &\sum_{i \in G} \mathbb{E}_{q(C_G|\mathbf{X}_G;\phi_c)}[\mathbb{E}_{q(S_i|X_i;\phi_s)}[\log p(X_i|C_G, S_i; \theta)]] \\ &- \sum_{i \in G} \text{KL}(q(S_i|X_i; \phi_s)||p(S_i)) - \text{KL}(q(C_G|\mathbf{X}_G; \phi_c)||p(C_G)).\end{aligned} \tag{3}$$

We define the average group ELBO over the dataset, $\mathcal{L}(\mathcal{G}, \theta, \phi_c, \phi_s) := \dfrac{1}{|\mathcal{G}|} \displaystyle\sum_{G \in \mathcal{G}} \text{ELBO}(G; \theta, \phi_s, \phi_c)$ and we maximise $\mathcal{L}(\mathcal{G}, \phi_c, \phi_s, \theta)$. It is a lower bound on $\dfrac{1}{|\mathcal{G}|} \log p(\mathbf{X}|\theta)$ because each group Evidence Lower Bound ELBO$(G; \theta, \phi_s, \phi_c)$ is a lower bound on $p(\mathbf{X}_G|\theta)$, therefore,

$$\frac{1}{|\mathcal{G}|} \log p(\mathbf{X}|\theta) = \frac{1}{|\mathcal{G}|} \sum_{G \in \mathcal{G}} \log p(\mathbf{X}_G|\theta) \geq \mathcal{L}(\mathcal{G}, \phi_c, \phi_s, \theta). \tag{4}$$

In comparison, the original VAE model maximises the average ELBO over individual samples. In practise, we build an estimate of $\mathcal{L}(\mathcal{G}, \theta, \phi_c, \phi_s)$ using minibatches of group.

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

If we take each group $G \in \mathcal{G}_b$, in its entirety this is an unbiased estimate. When the groups sizes are too large, for efficiency, we subsample G and this estimate is biased. We discuss the bias in the supplementary material. The resulting algorithm is shown in Algorithm 1.

For each group G, in step 7 of Algorithm 1 we build the group content distribution by accumulating information from the result of encoding each sample in G. The question is how can we accumulate the information in a relevant manner to compute the group content distribution?

3.3 Accumulating Group Evidence using a Product of Normal densities

Our idea is to build the variational approximation of the single group content variable, $q(C_G|\mathbf{X}_G; \phi_c)$, from the encoding of the grouped observations $\mathbf{X}_G$. While any distribution could be employed, we focus on using a product of Normal density functions. Other possibilities, such as a mixture of density functions, are discussed in the supplementary material.

We construct the probability density function of the latent variable C_G taking the value c by multiplying $|G|$ normal density functions, each of them evaluating the probability of $C_G = c$ given $X_i = x_i, i \in G$,

$$q(C_G = c|\mathbf{X}_G = \mathbf{x}_G; \phi_c) \propto \prod_{i \in G} q(C_G = c|X_i = x_i; \phi_c), \tag{6}$$

where we assume $q(C_G|X_i = x_i; \phi_c)$ to be a Normal distribution $N(\mu_i, \Sigma_i)$. Murphy [2007] shows that the product of two Gaussians is a Gaussian. Similarly, in the supplementary material we show

that $q(C_G = c|\mathbf{X}_G = \mathbf{x}_G; \phi_c)$ is the density function of a Normal distribution of mean μ_G and variance Σ_G

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

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 $\sum_{i \in G} \Sigma_i^{-1}$. 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 $\mathrm{KL}(q(C_G|\mathbf{X}_G; \phi_c)||p(C_G))$ can be evaluated in closed-form. We also assume a Normal distribution for $q(S_i|X_i; \phi_s), i \in G$.

4 Experiments

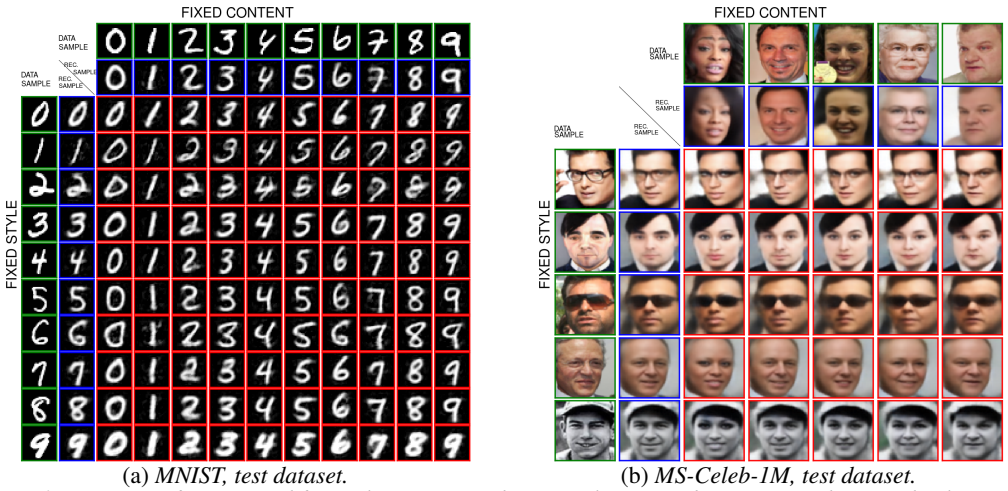
We evaluate the ML-VAE on images, other forms of data are possible and we leave these for future work. In all experiments we use the Product of Normal method presented in Section 3.3 to construct the content latent representation. Our goal with the experiments is twofold. First, we want to evaluate the performance of ML-VAE to learn a semantically meaningful disentangled representation. Second, we want to explore the impact of "accumulating evidence" described in Section 3.3. Indeed when we encode test images two strategies are possible: strategy 1 is disregarding the grouping information of the test samples, i.e. each test image is a group; and strategy 2 is considering the grouping information of the test samples, i.e. taking multiple test images per identity to construct the content latent representation.

MNIST dataset. We evaluate the ML-VAE on MNIST Lecun et al. [1998]. We consider the data grouped by digit label, i.e. the content latent code C should encode the digit label. We randomly separate the $60,000$ training examples into $50,000$ training samples and $10,000$ validation samples, and use the standard MNIST testing set. For both the encoder and decoder, we use a simple architecture of 2 linear layers (detailed in the supplementary material).

MS-Celeb-1M dataset. Next, we evaluate the ML-VAE on the face aligned version of the MS-Celeb-1M dataset Guo et al. [2016]. The dataset was constructed by retrieving approximately 100 images per celebrity from popular search engines, and noise has not been removed from the dataset. For each query, we consider the top ten results (note there was multiple queries per celebrity, therefore some identities have more than 10 images). This creates a dataset of $98,880$ entities for a total of $811,792$ images, and we group the data by identity. Importantly, we randomly separate the dataset in disjoints sets of identities as the training, validation and testing datasets. This way we evaluate the ability of ML-VAE level to generalise to unseen groups (unseen identities) at test-time. The training dataset consists of $48,880$ identities (total $401,406$ images), the validation dataset consists of $25,000$ identities (total $205,015$ images) and the testing dataset consists of $25,000$ identities (total $205,371$ images). The encoder and the decoder network architectures, composed of either convolutional or

Algorithm 1: ML-VAE training algorithm.

for *Each epoch* **do**
  Sample minibatch of groups $\mathcal{G}_b$,
  for $G \in \mathcal{G}_b$ **do**
    for $i \in G$ **do**
      Encode x_i into $q(C_G|X_i = x_i; \phi_c),\ q(S_i|X_i = x_i; \phi_s)$,
    end
    Construct $q(C_G|\mathbf{X}_G = \mathbf{x}_G; \phi_c)$ using $q(C_G|X_i = x_i; \phi_c), \forall i \in G$,
    for $i \in G$ **do**
      Sample $c_{G,i} \sim q(C_G|\mathbf{X}_G = \mathbf{x}_G; \phi_c)$, $s_i \sim q(S_i|X_i = x_i; \phi_s)$,
      Decode $c_{G,i}, s_i$ to obtain $p(X_i|C_G = c_{G,i}, S_i = s_i; \theta)$,
    end
  end
  Update θ, ϕ_c, ϕ_s by taking a gradient step of Equation (5): $\nabla_{\theta,\phi_c,\phi_s} \mathcal{L}(\mathcal{G}_b, \theta, \phi_c, \phi_s)$
end

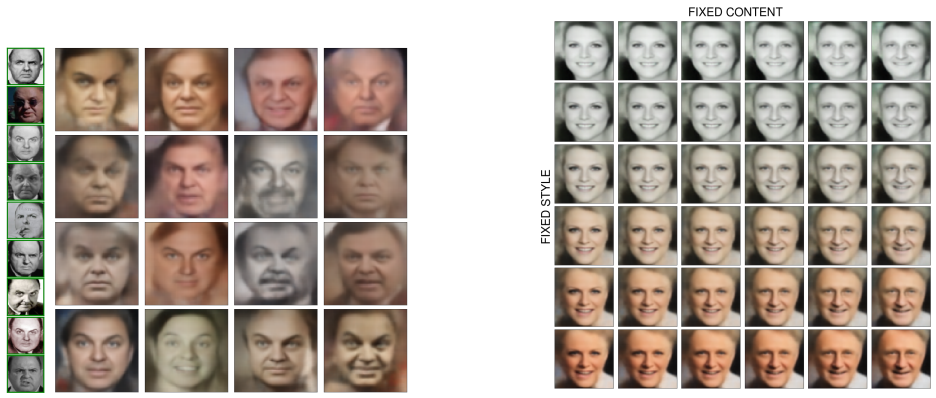


(a) *MNIST, test dataset.* (b) *MS-Celeb-1M, test dataset.*

Figure 4: *Swapping, first row and first column are test data samples (green boxes), second row and column are reconstructed samples (blue boxes) and the rest are swapped reconstructed samples (red boxes). Each row is fixed style and each column is a fixed content. Best viewed in color on screen.*

deconvolutional and linear layers, are detailed in the supplementary material. We resize the images to 64×64 pixels to fit the network architecture.

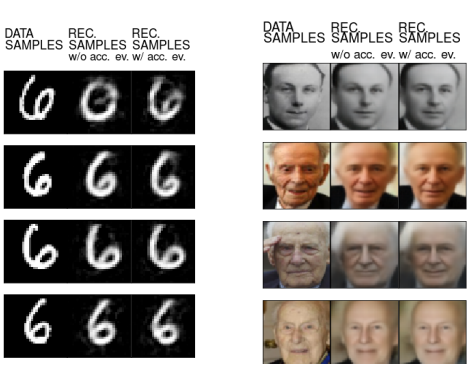
Qualitative Evaluation. As explained in Mathieu et al. [2016], there is no standard benchmark dataset or metric to evaluate a model on its disentanglement performance. Therefore similarly to Mathieu et al. [2016] we perform qualitative and quantitative evaluations. We qualitatively assess the relevance of the learned representation by performing operations on the latent space. First we perform swapping: we encode test images, draw a sample per image from its style and content latent representations, 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 the style and content samples. We present the results of swapping and interpolation with accumulating evidence of 10 other images in the group (strategy 2). Results without accumulated evidence (strategy 1) are also convincing and available in the supplementary material. We also perform generation: for a given test identity, we build the content latent code by accumulating images of this identity. Then take the mean of the resulting content distribution and generate images with styles sampled from the prior. Finally in order to explore the benefits of taking into account the grouping information, for a given test identity, we reconstruct all images for this identity using both these strategies and show the resulting images. Figure 4 shows the swapping procedure, where the first row and the first column show the test data sample input to ML-VAE,



(a) *Generation, the green boxed images are all the test data images for this identity. On the right, sampling from the random prior for the style and using the mean of the grouped images latent code.*

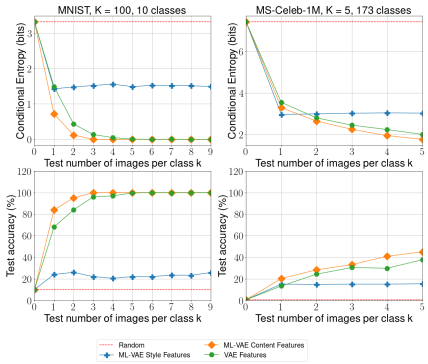
(b) *Interpolation, from top left to bottom right rows correspond to a fixed style and interpolating on the content, columns correspond to a fixed content and interpolating on the style.*

Figure 5: *Left: Generation. Right: Interpolation. Best viewed in color on screen.*



(a) *The four digits are of the same label.*

(b) *The four images are of the same person.*



(c) *Quantitative Evaluation. For clarity on MNIST we show up to $k = 10$ as values stay stationary for larger k (in supplementary material).*

Figure 6: *Accumulating evidence (acc. ev.). Left column are test data samples, middle column are reconstructed sample without acc. ev., right column are reconstructed samples with acc. ev. from the four images. In (a), ML-VAE corrects inference (wrong digit label in first row second column) with acc. ev. (correct digit label in first row third column). In (b), where images of the same identity are taken at different ages, ML-VAE benefits from group information and the facial traits with acc. ev. (third column) are more constant than without acc. ev. (second column). Best viewed in color on screen.*

second row and column are reconstructed samples. Each row is a fixed style and each column is a fixed content. We see that the ML-VAE disentangles the factors of variation of the data in a relevant manner. In the case of MS-Celeb-1M, we see that the model encodes the factor of variation that grouped the data, that is the identity, into the facial traits which remain constant when we change the style, and encodes the style into the remaining factors (background color, face orientation for example). The ML-VAE learns this meaningful disentanglement without the knowledge that the images are grouped by identity, but only the organisation of the data into groups. Figure 5 shows interpolation and generation. We see that our model covers the manifold of the data, and that style and content are disentangled. In Figures 6a and 6b, we reconstruct images of the same group with and without taking into account the grouping information. We see that the ML-VAE handles cases where there is no group information at test-time, and benefits from accumulating evidence if available.

Quantitative Evaluation. In order to quantitatively evaluate the disentanglement power of ML-VAE, we use the style latent code S and content latent code C as features for a classification task. The quality of the disentanglement is high if the content C is informative about the class, while the style S 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 composed of two linear layers of 256 hidden units each, once using S and once using C as input features. Again we explore the benefits of accumulating evidence: while we construct the variational approximation on the content latent code by accumulating K images per class for training the classifier, we accumulate only $k \leq K$ images per class at test time, where $k = 1$ corresponds to no group information. When k increases we expect the performance of the classifer trained on C to improve as the features become more informative and the performance using features S to remain constant. We compare to the original VAE model, where we also accumulate evidence by using the Product of Normal method on the VAE latent code for samples of the same class. The results are shown in Figure 6c. The ML-VAE content latent code is as informative about the class as the original VAE latent code, both in terms of classification accuracy and conditional entropy. ML-VAE also provides relevant disentanglement as the style remains uninformative about the class. Details on the choices of K and this experiment are in the supplementary material.

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 evaluation show the relevance of our method, as the ML-VAE learns a semantically meaningful disentanglement, generalises to unseen groups and enables control on the latent representation. For future work, we wish to apply the ML-VAE to text data.

References

Ehsan Abbasnejad, Anthony R. Dick, and Anton van den Hengel. Infinite variational autoencoder for semi-supervised learning. *arXiv preprint arXiv:1611.07800*, 2016.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. *ICLR*, 2017.

Miltiadis Allamanis, Pankajan Chanthirasegaran, Pushmeet Kohli, and Charles Sutton. Learning continuous semantic representations of symbolic expressions. *arXiv preprint 1611.01423*, 2017.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, August 2013. ISSN 0162-8828.

Diane Bouchacourt, Pawan Kumar Mudigonda, and Sebastian Nowozin. DISCO nets : Dissimilarity coefficients networks. *NIPS*, 2016.

Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *arXiv preprint arXiv:1612.05424*, 2016.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. *NIPS*, 2016.

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *ICLR*, 2017.

Chris Donahue, Akshay Balsubramani, Julian McAuley, and Zachary C. Lipton. Semantically decomposing the latent spaces of generative adversarial networks. *arXiv preprint 1705.07904*, 2017.

Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. *UAI*, 2015.

Harrison Edwards and Amos J. Storkey. Censoring representations with an adversary. *CoRR*, 2015.

T.-C. Fu, Y.-C. Liu, W.-C. Chiu, S.-D. Wang, and Y.-C. F. Wang. Learning Cross-Domain Disentangled Deep Representation with Supervision from A Single Domain. *arXiv preprint arXiv:1705.01314*, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NIPS*, 2014.

Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. *ECCV*, 2016.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *JMLR*, 2013.

T Kim, M Cha, H Kim, J Lee, and J Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *ICLR*, 2014.

Tejas D Kulkarni, Will Whitney, Pushmeet Kohli, and Joshua B Tenenbaum. Deep convolutional inverse graphics network. *NIPS*, 2015.

Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, pages 2278–2324, 1998.

Yujia Li, Kevin Swersky, and Richard S. Zemel. Generative moment matching networks. *ICML*, 2015.

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

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard S. Zemel. The variational fair autoencoder. *ICLR*, 2016.

Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. *NIPS*, 2016.

Vijayaraghavan Murali, Swarat Chaudhuri, and Chris Jermaine. Bayesian sketch learning for program synthesis. *arXiv preprint arXiv:1703.05698v2*, 2017.

Kevin P. Murphy. Conjugate Bayesian Analysis of the Gaussian Distribution. Technical report, 2007.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *ICML*, 2014.

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. *arXiv preprint arXiv:1612.07828*, 2017.

N. Siddharth, Brooks Paige, Alban Desmaison, Frank Wood, and Philip Torr. Learning disentangled representations in deep generative models. *Submitted to ICLR*, 2017.

Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *ICLR*, 2017.

N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *37th annual Allerton Conference on Communication, Control and Computing*, 1999.

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

Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. *arXiv preprint arXiv:1704.02510*, 2017.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.

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

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

1 Mixture of Normals Method

We discuss here a method to construct the variational approximation $q(C_G = c|\boldsymbol{X_G} = \boldsymbol{x}_G, \phi_c)$ as a mixture of $|G|$ density functions, each of them evaluating the probability of $C_G = c$ given $X_i = x_i$. This is an alternative to the Product of Normals method.

$$q(C_G = c|\boldsymbol{X_G} = \boldsymbol{x}_G, \phi_c) = \frac{1}{|G|}\sum_{i=1}^{|G|} q(C_G = c|X_i = x_i, \phi_c) \tag{1}$$

We assume $q(C_G|X_i = x_i, \phi_c)$ to be a Normal distribution $N(\mu_i, \Sigma_i)$. However, the term $\mathrm{KL}(q(C_G|\mathbf{X}_G; \phi_c)||p(C_G))$ can not be computed in closed form

$$\begin{aligned}\mathrm{KL}(q(C_G|\mathbf{X}_G; \phi_c)||p(C_G)) &= \mathbb{E}_{q(C_G|\boldsymbol{X}_G, \phi_c)}[\log q(C_G|\mathbf{X}_G, \phi_c) - \log p(C_G)] \\ &= \mathbb{E}_{q(C_G|\boldsymbol{X}_G, \phi_c)}[\log q(C_G|\mathbf{X}_G, \phi_c)] - \mathbb{E}_{q(C_G|\boldsymbol{X}_G, \phi_c)}[\log p(C_G)]\end{aligned} \tag{2}$$

We estimate this term by sampling L samples $c_l \sim q(C_G|\mathbf{X}_G; \phi_c)$ and computing the estimate:

$$\frac{1}{L}\sum_{l=1}^{L} \log \frac{1}{|G|}\sum_{i=1}^{|G|} q(C_G = c_l|X_i = x_i, \phi_c) - \frac{1}{L}\sum_{l=1}^{L} \log p(C_G = c_l) \tag{3}$$

In our experiments we used $L = |G|$ as we used the samples we draw to compute the first term of the objective function, that is $\sum_{i \in G} \mathbb{E}_{q(C_G|\mathbf{X}_G; \phi_c)}[\mathbb{E}_{q(S_i|X_i; \phi_s)}[\log p(X_i|C_G, S_i; \theta)]]$.

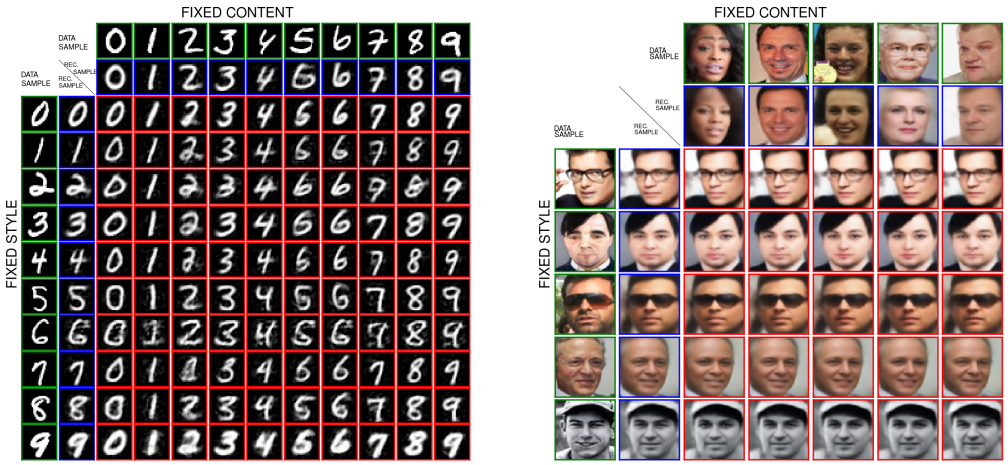
Figures 1 shows the qualitative results of the Mixture of Normal method. Qualitative evaluation indicate a better disentanglement with the Product of Normal method therefore we focused on the Product. On MS-Celeb-1M, the Mixture of Normal densities seems to store information about the grouping into the style: when style gets transfered, the facial features along the columns (which should remain constant as it is the same identity) tend to change[2]. Nevertheless, we present it as it might be better suited to other datasets and other tasks.

2 Experimental Details

In all experiments we use the Adam optimiser presented in Kingma and Ba [2015] with $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-8$. We use diagonal covariances for the Normal variational

*The work was performed as part of an internship at Microsoft Research.

[2]The previous version of the supplementary material had the wrong figure on MS-Celeb-1M (computed with another model). This updated figure shows the actual results and emphasize the conclusion that information about content is stored in the style.



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

Figure 1: *ML-VAE with Mixture of Normal densities. We intentionally show the same images as the ML-VAE with Product of Normal for comparison purposes. Swapping, first row and first column are test data samples (green boxes), second row and column are reconstructed samples (blue boxes) and the rest are swapped reconstructed samples (red boxes). Each row is fixed style and each column is a fixed content. Best viewed in color on screen. As we mention, compared to ML-VAE with Product of Normal, the Mixture of Normal seems to store information about the grouping into the style.*

approximations $q(\mathbf{S}_G|\mathbf{X}_G;\phi_s), q(C_G|\mathbf{X}_G;\phi_c)$. in both experiments we use a Normal distribution with diagonal covariances for the posterior $p(X_i|C_G,S_i;\theta)$. We train the model for 2000 epochs on MNIST and 250 epochs on MS-Celeb-1M. When we compare the original VAE and the ML-VAE we use early stopping on the validation set for MS-Celeb-1M as the architecture is larger and prone to over-fitting. In the specific case of Stochastic Variational Inference (SVI) Hoffman et al. [2013], for MNIST we train the model for 2000 epochs and proceed to 40000 epochs of inference at test-time, and for MS-Celeb-1M we use 500 training epochs and proceed to 200 epochs of inference at test-time.

2.1 Networks Architectures

MNIST Lecun et al. [1998]. We use an encoder network composed of a first linear layer e_0 that takes as input a 1×784-dimensional MNIST image x_i, x_i is a realisation of X_i. Layer e_0 has 500 hidden units and the hyperbolic tangent activation function. After layer e_0 we separate the network into 4 linear layers,$e_{m,s}, e_{v,s}$ and ,$e_{m,c}, e_{v,c}$ each of size $500 \times d$ where d is the dimension of both latent representations S and C. The layers $e_{m,s}, e_{v,s}$ take as input the output of e_0 and output respectively the mean and log-variance of the Normal distribution $q(S_i|X_i = x_i;\phi_s)$. The layers $e_{m,c}, e_{v,c}$ take as input the output of l_0 and output respectively the mean and variance of the Normal distribution $q(C_G|X_i = x_i;\phi_c)$.

We then construct $q(C_G|\mathbf{X}_G;\phi_c)$ from $q(C_G|X_i = x_i;\phi_c), i \in G$. Let us denote G the group in which x_i belongs. As explained in step 9 of Algorithm 1 in the main paper, for each input x_i we draw a sample $c_{G,i} \sim q(C_G|\mathbf{X}_G = \mathbf{x}_G;\phi_c)$ for the content of the group G, and a sample $s_i \sim q(S_i|X_i = x_i;\phi_s)$ of the style latent representation. We concatenate $(c_{G,i}, s_i)$ into a $2 \times d$-dimensional vector that is fed to the decoder.

The decoder network is composed of a first linear layer d_0 that takes as input the $2 \times d$-dimensional vector $(c_{G,i}, s_i)$. Layer d_0 has 500 hidden units and the hyperbolic tangent activation function. A second linear layer d_2 takes as input the output of d_0 and outputs a 784-dimensional vector representating the parameters of the Normal distribution $p(X_i|C_G,S_i;\theta)$. We use in our experiments $d = 10$ for a total latent representation size of respectively 20.

MS-Celeb-1M Guo et al. [2016]. We use an encoder network composed of a four convolutional layers e_1, e_2, e_3, e_4 all of stride 2 and kernel size 4. They are composed of respectively $64, 128, 256$and512 filters. All four layers are followed by Batch Normalisation and Rectified Linear Units (ReLU) activation functions. The fifth and sixth layers e_5, e_6 are linear layers with 256 hidden units, followed by Batch Normalisation and Concatenated Rectified Linear Unit (CReLU) activation functions. Similarly to the MNIST architecture, after layer e_6 we separate the network

into 4 linear layers,$e_{m,s}, e_{v,s}$ and ,$e_{m,c}, e_{v,c}$ each of size $256 \times 2 \times d$ where d is the dimension of both latent representations S and C. The layers for the log-variances are followed by the tangent hyperbolic activation function and multiplied by 5.

Similarly as the MNIST experiment we construct the latent representation and sample it as explained in Algorithm 1 in the main paper.

The decoder network is composed of 3 deconvolutional layers d_1, d_2, d_3 all of stride 2 and kernel size 4. They are composed of respectively $256, 128, 64$ filters. All four layers are followed by Batch Normalisation and Rectified Linear Units (ReLU) activation functions. The seventh and eight layers are deconvolutional layers composed of 3 filters, of stride 1 and kernel size 3 and output respectively the mean and log-variance of the Normal distribution $p(X_i|C_G, S_i; \theta)$. The layer for the log-variances are followed by the tangent hyperbolic activation function and multiplied by 5. We use in our experiments $d = 50$ for a total latent representation size 100. We use padding in the convolutional and deconvolutional layers to match the data size.

Specific case for Stochastic Variational Inference (SVI) Hoffman et al. [2013]. We compare in our experiments with Stochastic Variational Inference (SVI), from Hoffman et al. [2013]. In the case of SVI, the encoder is an embedding layer mapping each sample x_i in a group G to the non-shared parameters $\phi_{s,i}$ of its style latent representation $q(S_i|\phi_{s,i})$ and to the non-shared parameters $\phi_{c,G}$ of its group content latent representation $q(C_G|\phi_{c,G})$. The decoder is the same as the ML-VAE.

3 Quantitative Evaluation details

We explain in Section 4 of the main paper how we quantitatively evaluate the disentanglement performance of our model. We give details here for the interested reader. Figure 2 show the quantitative evaluation for k up to $k = K = 100$ on MNIST.

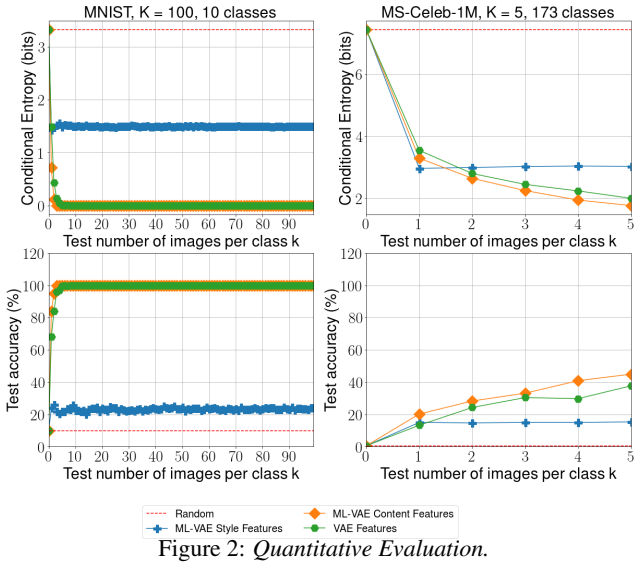


Figure 2: *Quantitative Evaluation.*

3.1 Classifier architecture

The classifier is a neural network composed of two linear layers of 256 hidden units each. The first layer is followd by a tangent hyperbolic activation function. The second layer is followed by a softmax activation function. We use the cross-entropy loss. We use the Adam optimiser presented in Kingma and Ba [2015] with $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-8$. Note that the training, validation and testing sets for the classifier are all composed of test images, and each set is composed of K times the number of classes; hence our choice of K and number of classes for MS-Celeb-1M. In the case of MNIST, there are only 10 classes, therefore when k is small we would take only a small number of images to test the classifier. Therefore we perform 5 trials of test procedures of the classifier, each trial using different test images, and report the mean performance.

3.2 Conditional entropy computation.

We show here that training the neural network classifier with the cross-entropy loss is a proxy for minimising the conditional entropy of the class given the latent code features C or S.

Let us take the example of the latent code C used as features. We denote a class Y and we train the neural network classifier to model the distribution $r(Y|C)$ by minimising the cross-entropy loss, which corresponds to maximising $\mathbb{E}_{p(Y,C)}[\log r(Y|C)]$ where $p(Y, C)$ is estimated using samples

$$\begin{aligned} &\text{Sample a class } y \sim p(Y), \\ &\text{Sample grouped observations for this class } \mathbf{x}_{G_Y} \sim p(\mathbf{X}_{G_Y}), \\ &\text{Sample the latent code to use as features } c_{G_Y} \sim q(C_{G_Y}|\mathbf{X}_{G_Y}, \phi_c) \end{aligned} \tag{4}$$

The conditional entropy of the class Y given the latent code C is expressed as

$$\mathbb{H}(Y|C) = -\mathbb{E}_{p(Y,C}[\log p(Y|C)] \tag{5}$$

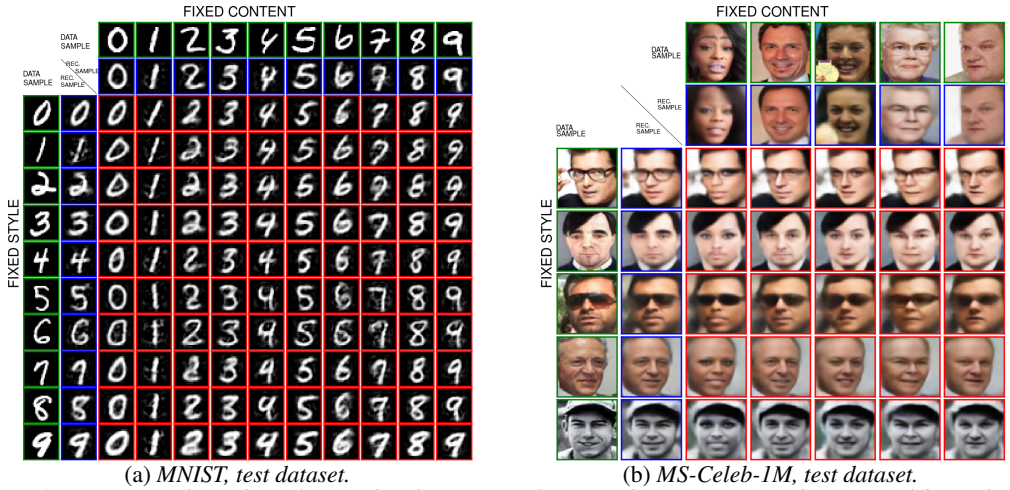
We can write

$$\begin{aligned} \mathbb{H}(Y|C) &= -\mathbb{E}_{p(Y,C}[\log p(Y|C)] = -\mathbb{E}_{p(Y,C)}[\log \frac{p(Y|C)}{r(Y|C)} r(Y|C)] \\ &= -\mathbb{E}_{p(Y,C)}[\log r(Y|C)] - \mathbb{E}_{p(Y,C)}[\log \frac{p(Y|C)}{r(Y|C)}] \\ &= -\mathbb{E}_{p(Y,C)}[\log r(Y|C)] - \mathbb{E}_{p(Y,C)}[\log \frac{p(Y,C)}{r(Y,C)}] \\ &= -\mathbb{E}_{p(Y,C)}[\log r(Y|C)] - \mathrm{KL}(p(Y,C)||r(Y,C)) \leq -\mathbb{E}_{p(Y,C)}[\log r(Y|C)] \end{aligned} \tag{6}$$

since the Kullback-Leibler is always positive. Therefore, training the neural network classifier to minimise the cross-entropy loss is equivalent to minimising an upper bound on the conditional entropy of the class given the latent code features C. We report the value of $\mathbb{E}_{p(Y,C)}[\log r(Y|C)]$ on the classifier test set in the paper as the reported Conditional entropy in bits. Similarly we report the performances with the style latent code or the latent code of the original VAE model.

4 ML-VAE with Product of Normal Without Accumulating Evidence

We show in Figure 3 and 4 the results of swapping and interpolation of the ML-VAE with Product of Normal without accumulating evidence (strategy 1 in the main paper). We intentionally show the same images for comparison purposes.



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

Figure 3: *ML-VAE with Product of Normal without accumulating evidence. Swapping, first row and first column are test data samples (green boxes), second row and column are reconstructed samples (blue boxes) and the rest are swapped reconstructed samples (red boxes). Each row is fixed style and each column is a fixed content. Best viewed in color on screen.*

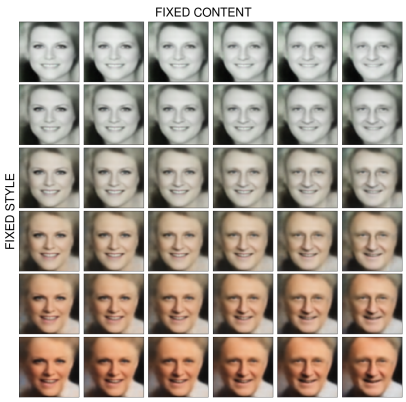


Figure 4: *ML-VAE without accumulating evidence. Interpolation, from top left to bottom right rows correspond to a fixed style and interpolating on the content, columns correspond to a fixed content and interpolating on the style.*

5 Accumulating Group Evidence using a Product of Normal densities: Detailed derivations

We construct the probability density function of the random variable C_G by multiplying $|G|$ normal density functions, each of them evaluating the probability of C_G under the realisation $X_i = x_i$, where $i \in G$.

$$q(C_G|\mathbf{X}_G = \mathbf{x}_G; \phi_c) \propto \prod_{i \in G} q(C_G|X_i = x_i; \phi_c) \tag{7}$$

We assume $q(C_G|X_i = x_i; \phi_c)$ to be a Normal distribution $N(\mu_i, \Sigma_i)$. The normalisation constant is the resulting product marginalised over all possible values of C_G.

The result of the product of $|G|$ normal density functions is proportional to the density function of a Normal distribution of mean μ_G and variance Σ_G.

$$\begin{aligned} \Sigma_G^{-1} &= \sum_{i \in G} \Sigma_i^{-1} \\ \mu_G^T \Sigma_G^{-1} &= \sum_{i \in G} \mu_i^T \Sigma_i^{-1} \end{aligned} \tag{8}$$

We show below how we derive the expressions of mean μ_G and variance Σ_G.

$$\begin{aligned}
\prod_{i\in G} q(C_G = c|X_i = x_i;\phi_c) &= \prod_{i\in G} \frac{1}{\sqrt{(2\pi)^d|\Sigma_i|}} \exp\Big(-\frac{1}{2}(c-\mu_i)^T\Sigma_i^{-1}(c-\mu_i)\Big) \\
&= K_1 \exp\Big(-\frac{1}{2}\sum_{i\in G}(c-\mu_i)^T\Sigma_i^{-1}(c-\mu_i)\Big) \\
&= K_1 \exp\Big(-\frac{1}{2}(\sum_{i\in G} c^T\Sigma_i^{-1}c + \mu_i^T\Sigma_i\mu_i - 2\mu_i^T\Sigma_i^{-1}c)\Big) \\
&= K_1K_2 \exp\Big(\sum_{i\in G}\mu_i^T\Sigma_i^{-1}c - \frac{1}{2}c^T\Sigma_i^{-1}c\Big) \\
&= K_1K_2 \exp\Big(\sum_{i\in G}\mu_i^T\Sigma_i^{-1}c - \frac{1}{2}c^T\sum_{i\in G}\Sigma_i^{-1}c\Big) \\
&= K_1K_2 \exp\Big(\mu_G^T\Sigma_G^{-1}c - \frac{1}{2}c^T\Sigma_G^{-1}c\Big) \\
&= K_1K_2 \exp\Big(-\frac{1}{2}(c^T\Sigma_G^{-1}c - 2\mu_G^T\Sigma_G^{-1}c)\Big) \\
&= K_1K_2 \exp\Big(-\frac{1}{2}(c^T\Sigma_G^{-1}c - 2\mu_G^T\Sigma_G^{-1}c + \mu_G^T\Sigma_G^{-1}\mu_G - \mu_G^T\Sigma_G^{-1}\mu_G)\Big) \\
&= K_1K_2\exp(-\mu_G^T\Sigma_G^{-1}\mu_G)\exp\Big(-\frac{1}{2}(c^T\Sigma_G^{-1}c - 2\mu_G^T\Sigma_G^{-1}c + \mu_G^T\Sigma_G^{-1}\mu_G\Big) \\
&= K_1K_2K_3 \exp\Big(-\frac{1}{2}(c^T\Sigma_G^{-1}c - 2\mu_G^T\Sigma_G^{-1}c + \mu_G^T\Sigma_G^{-1}\mu_G\Big) \\
&= K_1K_2K_3K_4\frac{1}{\sqrt{(2\pi)^d|\Sigma_G|}} \exp\Big(-\frac{1}{2}(c-\mu_G)^T\Sigma_G^{-1}(c-\mu_G)\Big)
\end{aligned} \tag{9}$$

where d is the dimensionality of c and

$$\begin{aligned}
K_1 &= \prod_{i\in G}\frac{1}{\sqrt{(2\pi)^d|\Sigma_i|}} \\
K_2 &= \exp(-\frac{1}{2}\sum_{i\in G}\mu_i^T\Sigma_i^{-1}\mu_i) \\
K_3 &= \exp(\frac{1}{2}\mu_G^T\Sigma_G^{-1}\mu_G) \\
K_4 &= \sqrt{(2\pi)^d|\Sigma_G|}
\end{aligned} \tag{10}$$

This is a normal distribution, scaled by $K_1K_2K_3K_4$, of mean μ_G and variance Σ_G

$$\begin{aligned}
\Sigma_G^{-1} &= \sum_{i\in G}\Sigma_i^{-1}, \\
\mu_G^T\Sigma_G^{-1} &= \sum_{i\in G}\mu_i^T\Sigma_i^{-1}
\end{aligned} \tag{11}$$

However, the constant of normalisation disappears when we rescale the resulting product in order for the resulting product to integrate to 1.

$$
\begin{aligned}
q(C_G = c|X_G = x_G; \phi_c) &= \frac{K_1K_2K_3K_4 \frac{1}{\sqrt{(2\pi)^d|\Sigma_G|}} \exp\Big(-\frac{1}{2}(c-\mu_G)^T\Sigma_G^{-1}(c-\mu_G)\Big)}{\int_c K_1K_2K_3K_4 \frac{1}{\sqrt{(2\pi)^d|\Sigma_G|}} \exp\Big(-\frac{1}{2}(c-\mu_G)^T\Sigma_G^{-1}(c-\mu_G)\Big)dc} \\
&= \frac{\exp\Big(-\frac{1}{2}(c-\mu_G)^T\Sigma_G^{-1}(c-\mu_G)\Big)}{\int_c \exp\Big(-\frac{1}{2}(c-\mu_G)^T\Sigma_G^{-1}(c-\mu_G)\Big)dc} \\
&= \frac{1}{\sqrt{(2\pi)^d|\Sigma_G|}} \exp\Big(-\frac{1}{2}(c-\mu_G)^T\Sigma_G^{-1}(c-\mu_G)\Big)
\end{aligned} \tag{12}
$$

6 Bias of the Objective

We detail the bias induced by taking a subset of the samples in a group as mentioned in Section 3 of the main paper. We build an estimate of $\mathcal{L}(\mathcal{G}, \theta, \phi_c, \phi_s)$ using mini-batches of grouped observations.

$$
\mathcal{L}(\mathcal{G}_b, \theta, \phi_c, \phi_s) = \frac{1}{|\mathcal{G}_b|} \sum_{G \in \mathcal{G}_b} \text{ELBO}(G; \theta, \phi_s, \phi_c) \tag{13}
$$

If we take all observations in each group $G \in \mathcal{G}_b$, it is an unbiased estimate. However when the groups sizes are too large and we subsample G, this estimate is biased. The $\text{ELBO}(G; \theta, \phi_s, \phi_c)$ for a group is

$$
\begin{aligned}
\text{ELBO}(G; \theta, \phi_s, \phi_c) = &\sum_{i \in G} \mathbb{E}_{q(C_G|\mathbf{X}_G;\phi_c)}[\mathbb{E}_{q(S_i|X_i;\phi_s)}[\log p(X_i|C_G, S_i; \theta)]] \\
&- \sum_{i \in G} \text{KL}(q(S_i|X_i; \phi_s)||p(S_i)) - \text{KL}(q(C_G|\mathbf{X}_G; \phi_c)||p(C_G)).
\end{aligned} \tag{14}
$$

Let us take a subsample H of G and consider the estimate $\text{ELBO}(G; \theta, \phi_s, \phi_c)_H$. The superscript H denotes the fact that we use a subsample of G to estimate $\text{ELBO}(G; \theta, \phi_s, \phi_c)_H$

$$
\begin{aligned}
\text{ELBO}(G; \theta, \phi_s, \phi_c)_H = &\sum_{i \in H \subseteq G} \mathbb{E}_{q(C_G|\mathbf{X}_H;\phi_c)}[\mathbb{E}_{q(S_i|X_i;\phi_s)}[\log p(X_i|C_G, S_i; \theta)]] \\
&- \sum_{i \in H \subseteq G} \text{KL}(q(S_i|X_i; \phi_s)||p(S_i)) - \text{KL}(q(C_G|\mathbf{X}_H; \phi_c)||p(C_G)).
\end{aligned} \tag{15}
$$

where $q(C_G|\mathbf{X}_H; \phi_c)$ is computed using $\mathbf{X}_H$.

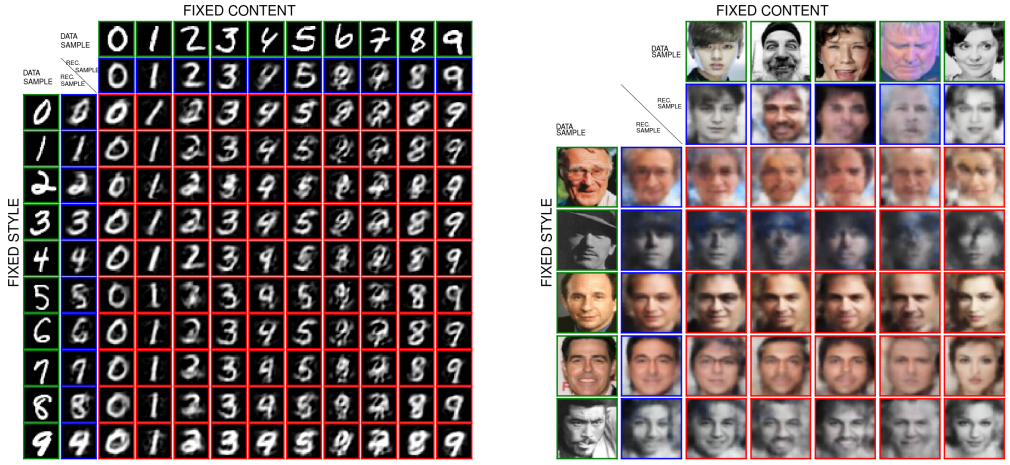
The gradient with respect to the parameters θ, ϕ_s, ϕ_c is

$$
\begin{aligned}
\nabla_\theta \text{ELBO}(G; \theta, \phi_s, \phi_c)_H = &\sum_{i \in H \subseteq G} \mathbb{E}_{q(C_G|\mathbf{X}_H;\phi_c)}[\mathbb{E}_{q(S_i|X_i;\phi_s)}[\nabla_\theta \log p(X_i|C_G, S_i; \theta)]], \\
\nabla_{\phi_s} \text{ELBO}(G; \theta, \phi_s, \phi_c)_H = &\sum_{i \in H \subseteq G} \mathbb{E}_{q(C_G|\mathbf{X}_H;\phi_c)}[\nabla_{\phi_s}\mathbb{E}_{q(S_i|X_i;\phi_s)}[\log p(X_i|C_G, S_i; \theta)]], \\
&- \sum_{i \in H \subseteq G} \nabla_{\phi_s} \text{KL}(q(S_i|X_i; \phi_s)||p(S_i)) \\
\nabla_{\phi_s} \text{ELBO}(G; \theta, \phi_s, \phi_c)_H = &\sum_{i \in H \subseteq G} \nabla_{\phi_c}\mathbb{E}_{q(C_G|\mathbf{X}_H;\phi_c)}[\mathbb{E}_{q(S_i|X_i;\phi_s)}[\log p(X_i|C_G, S_i; \theta)]] \\
&- \nabla_{\phi_c} \text{KL}(q(C_G|\mathbf{X}_H; \phi_c)||p(C_G)).
\end{aligned} \tag{16}
$$

Since we build the content variational approximation in a non-linear manner with the Product of Normals or the Mixture of Normals methods, the gradients $\nabla_\theta \text{ELBO}(G; \theta, \phi_s, \phi_c)_H$ and $\nabla_{\phi_c} \text{ELBO}(G; \theta, \phi_s, \phi_c)_H$ do not decompose in an unbiased manner, i.e. by summing the gradient of subsampled groups we do not retrieve the gradient computed using the entire group. The resulting bias will depend on the method employed. For future work, we plan on analysing the effect of the bias. In detail we want to derive a manner to correct the bias and verify that the biased estimator do not over-estimate the true objective function, that is the sum of the groups Evidence Lower Bound.

7 Stochastic Variational Inference (SVI) Results

We show in Figure 5 the qualitative results of Stochastic Variational Inference (SVI) Hoffman et al. [2013] on the swapping evaluation. We see that while SVI disentangles the style and the content, but the resulting image quality is poor. In the case of MS-Celeb-1M, we trained SVI for twice the number of epochs of the other models, that is in total 500 epochs, the training objective to maximise, that is the average group Evidence Lower Bound, remains lower than the ML-VAE model at the end of the training. This is because SVI does not share the parameters ϕ_c, ϕ_s among observations at training, hence take a longer time to train. It is the first disadvantadge of SVI compared to VAE-based model. At test-time, the SVI model requires expensive iterative inference. In the case of MS-Celeb-1M we used 200 epochs of test inference, it is possible that more epochs of test-time inference would have lead to better quality images but this already shows the limits of non-amortised variational inference and the advantages of the ML-VAE. We see that while SVI disentangles the style and the content, but the resulting image quality is poor.



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

Figure 5: *Stochastic Variational Inference (SVI) Hoffman et al. [2013] qualitative results. Swapping, first row and first column are test data samples (green boxes), second row and column are reconstructed samples (blue boxes) and the rest are swapped reconstructed samples (red boxes). Each row is fixed style and each column is a fixed content. Best viewed in color on screen.*

8 Other Formulations Explored

We explored other possible formulations and we detail them here for the interested reader, along with the reasons for which we did not pursue them.

SVI-Encode. We explained the disavantadges of Stochastic Variational Inference (SVI), see Hoffman et al. [2013]. In order to remove the need for costly inference at test-time, we tried training an encoder to model the variational approximation of the latent representation C, S corresponding to the generative model of the trained SVI model. We do not use training data but generated observations, sampling the latent representation from the prior. The encoder does not have any group information, and each sample has a separate content and style latent representation C_i, S_i. The model maximises the log-likelihood of the generative model under the latent code representation.

$$\sum_{i \in N} \mathbb{E}_{q(C_i, S_i | X_i; \phi)} [\log p(X_i | C_i, S_i, \theta)] \tag{17}$$

where $p(X_i|C_i, S_i, \theta)$ is the distribution corresponding to the generative model and $q(C_i, S_i|X_i; \phi)$ is the variational approximation. We refer to this method as SVI-Encode. The qualitative quality of this model on test samples is highly dependent on the quality of the generative model. Therefore it gives satisfactory qualitative results on MNIST, but poor qualitative results on MS-Celeb-1M where the qualitative results of SVI are not satisfactory. However, we think that a model trained alternatively between SVI and this SVI-Encode could benefit from the disentanglement power of SVI and amortised inference at test-time. We leave this for possible future work.

Regularising the objective. A possible formulation of the problem is to employ a regular VAE with an additional term to enforce observations within a group to have similar variational approximations of the content. This model separates the latent representation of the style, S and the latent representation of the content C but each observation X_i within a group has its own latent variables S_i and C_i. The sharing of the content within a group is enforced by adding a penalisation term based on a symmetrised Kullback-Leibler divergence between the content latent representation of the observations belonging to the same group. The resulting model maximises the objective[3]

$$\begin{aligned}
&\frac{1}{|\mathcal{N}|}\sum_{i\in N}\mathbb{E}_{q(C_i,S_i|X_i;\phi)}[\log p(X_i|C_i,S_i,\theta)] \\
&\quad-\frac{1}{|\mathcal{N}|}\sum_{i=1}^{N}\mathrm{KL}(q(S_i|X_i;\phi_s)||p(S_i))-\frac{1}{|\mathcal{N}|}\sum_{i=1}^{N}\mathrm{KL}(q(C_i|X_i;\phi_c)||p(C_i)) \\
&\quad-\frac{\lambda}{|\mathcal{G}|}\sum_{G\in\mathcal{G}}\frac{1}{|G|/2}\sum_{\substack{i\in[1,|G|/2],\\ s.t.X_{2i}\in G,X_{2i+1}\in G}}\frac{1}{2}\Big[\mathrm{KL}(q(C_{2i}|X_{2i},\phi_c)||q(C_{2i+1}|X_{2i+1},\phi_c)) \\
&\quad+\mathrm{KL}(q(C_{2i+1}|X_{2i+1},\phi_c)||q(C_{2i}|X_{2i},\phi_c)\Big],
\end{aligned} \tag{18}$$

where λ is an hyper-parameter to cross-validate. In our experiment, the drawback of this model is that if the latent representation size is too large (in detail, with the size 100 used in our experiments for MS-Celeb-1M), the model sets the content latent representation to the prior $p(C)$ is order to encounter a null penalty. The observations are encoded in the style latent representation only. This is a known problem of VAE, see Chen et al. [2017]. On the opposite, the ML-VAE model is more robust to this problem.

[3]The equation was corrected compared to the submitted version.

References

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *ICLR*, 2017.

Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. *ECCV*, 2016.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *JMLR*, 2013.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, pages 2278–2324, 1998.