Pattern Recognition 148 (2024) 110156

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Self-Supervised Adversarial Variational Learning

Fei Ye, Adrian. G. Bors *

Department of Computer Science, University of York, York YO10 5GH, UK

ARTICLE INFO

Keywords:
Self-supervised learning
Variational Autoencoders (VAE)
Generative Adversarial Nets (GAN)
Representation learning
Mutual information

ABSTRACT

A natural approach for representation learning is to combine the inference mechanisms of VAEs and the generative abilities of GANs, within a new model, namely VAEGAN. Most existing VAEGAN models would jointly train the generator and inference modules, which has limitations when learning representations generated by a pre-trained GAN model without data. In this paper, we develop a novel hybrid model, called the Self-Supervised Adversarial Variational Learning (SS-AVL) which introduces a two-step optimization procedure training separately the generator and the inference model. The primary advantage of SS-AVL over existing VAEGAN models is that SS-AVL optimizes the inference models in a self-supervised learning manner where the samples used for training the inference models are drawn from the generator distribution instead of using real samples. This can allow SS-AVL to learn representations from arbitrary GAN models without using real data. Additionally, we employ information maximization into the context of increasing the maximum likelihood, which encourages SS-AVL to learn meaningful latent representations. We perform extensive experiments to demonstrate the effectiveness of the proposed SS-AVL model.

1. Introduction

Generative Adversarial Nets (GANs) [1] is one of the most popular deep generative models, which has been applied in a wide range of applications, including for compressing sensing images [2], image synthesis, rain removal from images [3] and face image inpainting [4]. The primary drawback of GANs is their lacking of an inference mechanism, which prevents using them for representation learning. The other popular generative model is the Variational Autoencoder (VAE) [5]. Different from employing adversarial learning used for training GANs, VAEs aim to train jointly an inference model and a decoder enabling unified optimization which consists of the maximization of the sample log-likelihood. Unlike GANs, a VAE has an inference mechanism which can provide latent representations. However, VAEs tend to generate approximate results, such as blurred images. Therefore lately, many research studies would focus on hybrid models combining the advantages of both GANs and VAEs. Nevertheless, existing hybrid models usually optimize jointly the inference model and generator, which can lead to unstable training and cannot be used for learning representations from a pre-trained GAN model.

Representation learning in generative modelling aims to define and find several underlying factors that can describe the variability of image data. Meanwhile, unsupervised disentangled representation learning aims to decompose the latent representations into several independent variables, with each describing the variability of a certain characteristic in the data without using any supervision signals. In order to use disentanglement for practical applications, such data sets are assumed to contain semantically distinct clusters representing different categories of characteristics. The level of independence between such data categories should be measured by a criterion. Learning disentangled representations that may capture semantic meaningful information can allow to explicitly edit images and is useful for a variety of tasks [6, 7]. Furthermore, enabling disentangled representations can overcome overfitting during the training, leading to a better generalization in the resulting trained models [8].

Learning interpretable and disentangled representations has been considered in the β−VAE [9] which uses a large penalty on the Kullback–Leibler (KL) divergence term of the loss function, in order to encourage the independence between latent variables. However, a large penalty would sacrifice the quality of image reconstruction when inducing disentangled representations [10]. To address this issue, other research studies focus on using other penalty functions such as the total correlation (TC) [11] which is a measure of multivariate mutual independence. The primary drawback of VAE based approaches is that they generally produce blurred and unclear images compared to Generative Adversarial Networks (GANs) which generated clear images. On the other hand, few research efforts have been made to use GANs for disentangled representations [12] and their outcomes show rather mixed results. GANs are trickier to control and may produce unexpected artifacts in the generated images.

* Corresponding author.
E-mail address: adrian.bors@york.ac.uk (A.G. Bors).

https://doi.org/10.1016/j.patcog.2023.110156
Received 24 August 2022; Received in revised form 23 February 2023; Accepted 23 November 2023
Available online 29 November 2023

Recently, it was shown that by combining the GAN and VAE into a unified learning framework, the resulting model can address the drawbacks of both models [13]. However, the joint optimization of the generator and inference network in a hybrid model usually requires to design two loss functions: the reconstruction loss and the regularization loss [13]. These two loss functions have different training behaviours, which cannot guarantee an optimal solution for both at the same time. For example, the reconstruction loss encourages the encoding-decoding process to provide an accurate reconstruction for an input while it matches the true posterior distribution $p(\mathbf{z}\,|\,\mathbf{x})$ [5], for the encoding distribution $q(\mathbf{z}\,|\,\mathbf{x})$. The regularization loss, implemented through adversarial learning, encourages the encoding distribution $q(\mathbf{z}\,|\,\mathbf{x})$ to match the prior distribution $p(\mathbf{z})$, and would move $q(\mathbf{z}\,|\,\mathbf{x})$ far away from the true posterior distribution $p(\mathbf{z}\,|\,\mathbf{x})$, resulting in a poor generation $\mathbf{x}' \sim q(\mathbf{x}\,|\,\mathbf{z}), \mathbf{z} \sim p(\mathbf{z})$. Moreover, the joint optimization of the generator and inference models also require a careful hyperparameter configuration in order to ensure stable training.

After considering the drawbacks of existing hybrid models and VAE-based disentangled approaches, we develop a new self-supervised method for learning interpretable and disentangled representations, which combines the advantages of GANs and VAEs. Existing GAN-VAE hybrid methods train GANs and VAEs together, which is not possible for a pre-trained GAN model. Moreover, the inference models in these approaches require access to real data samples, which is challenging when real data samples cannot be accessed. To address these two issues, we propose to split the training of the hybrid model into two independent processes, where adversarial learning is performed only to train the generator, while a VAE optimization process is used to train the inference models in a self-supervised manner using pseudo-samples drawn by the generator. We aim to learn three latent representations $\{\mathbf{z}, \mathbf{c}, \mathbf{d}\}$ from the data, implemented by two inference models $q_\xi(\mathbf{z}\,|\,\mathbf{x})$ and $q_\omega(\mathbf{d}, \mathbf{c}\,|\,\mathbf{x})$, respectively, where the latent variable $\mathbf{z}$ is assumed to be a random noise vector in a generative process, while encouraging $\{\mathbf{c}, \mathbf{d}\}$ to capture continuous and discrete latent representations. To induce the interpretable representations $\{\mathbf{c}, \mathbf{d}\}$, we propose a mutual information optimization procedure that jointly optimizes the generator and inference model $q_\omega(\mathbf{d}, \mathbf{c}\,|\,\mathbf{x})$ during training. The proposed approach does not require a careful balance between the adversarial and VAE losses during training, providing a stable training paradigm and facilitating implementation. We perform a series of experiments on various datasets, and the empirical results demonstrate that our approach can produce sharp generative results when compared with other methods.

This research study brings the following contributions :

(1) A self-supervised learning procedure is proposed, where the inference model is estimated separately from the generator. This learning procedure provides many advantages over other hybrid methods, such that the inference learning does not affect the generator's optimization while it can also learn data representations from a trained GAN model.
(2) We introduce the mutual information optimization for the Self-Supervised Adversarial Variational Learning (SS-AVL) model to encourage learning interpretable and disentangled representations.
(3) Qualitative and quantitative results are provided demonstrating the capability of the proposed approach on disentangled and interpretable representation learning.

The rest of the paper is organized as in the following. The background and related works are presented in Section 2. The proposed SS-AVL model is introduced in Section 3 and its theoretical framework in Section 4. The mutual information maximization for interpretable representations is discussed in Section 5 and the training and implementation of the proposed model in Section 6. In Section 7 we discuss the limitations of the proposed method. The experimental results are presented and discussed in Section 8, while the conclusions are drawn in Section 9.

2. Background and related works

In the following we discuss the main approaches in generative deep learning.

Variational autoencoder (VAE). VAEs represent one of the most popular generative models, consisting of two network components, the encoder and decoder, which during the training aim to represent the conditional distributions $q_\theta(\mathbf{z}\,|\,\mathbf{x})$ and $p_\phi(\mathbf{x}\,|\,\mathbf{z})$, respectively, where $\mathbf{x}$ are the input data and $\mathbf{z}$ are the latent variables. A VAE aims to maximize an error lower bound objective (ELBO) to the marginal log-likelihood of the data distribution :

$$\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\theta(\mathbf{z}|\mathbf{x})}[\log p_\phi(\mathbf{x}\,|\,\mathbf{z})] - D_{KL}(q_\theta(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z})) \leq \log p(\mathbf{x}) \tag{1}$$

Generative adversarial networks (GAN). GANs also consist of two network components: generator and discriminator. These two components are trained alternatively and can be seen as playing a Minimax game, defined by the following loss :

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x}\sim pd(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log[1 - D(G(\mathbf{z}))]]. \tag{2}$$

While the discriminator network is trained to distinguish between real and fake data, the generator aims to produce more realistic data that can fool the discriminator.

Hybrid VAE-GAN models. Hybrid models attempt to address the drawbacks of both GANs and VAEs, by combining their architectures. These models usually consist of three components : an encoder to map data into the latent space, a generator to recover data from the latent space, and a discriminator to distinguish real from fake data. We provide a brief literature review for the hybrid model as follows.

Adversarial Autoencoders (AAE) [13] replace the Kullback–Leibler (KL) divergence, used in the objective function for training VAEs, with the adversarial loss encouraging the output distribution of the encoder to be as similar as possible to the prior distribution. Srivastava et al. [14] introduced the VEEGAN model, which uses a reconstructing network to avoid the model collapse from GANs. The final objective function in VEEGAN aims to minimize the upper bound of the posterior distribution. BiGANs, proposed in [15], is a hybrid model where the Discriminator network is trained to learn the inverse mapping, by projecting data back into the latent space. Huang et al. [16], proposed a hybrid model called the Introspective Variational Autoencoders (IntroVAE), which was applied in photographic image synthesis. Unlike most other hybrids methods which require an additional Discriminator network for adversarial learning, IntroVAE uses the Inference network as a discriminator to distinguish between fake and real data samples. A similar approach was proposed by Ulyanov et al. in [17], where the proposed Adversarial Generator-Encoder (AGE) does not require an extra Discriminator network. The Decoder and Encoder are jointly trained in order to optimize the objective function in an adversarial way. Mescheder et al. [18], introduced the Adversarial Variational Bayes (AVB), which trains a VAE in an adversarial way. This method employs an additional auxiliary Discriminator network that can rewrite the maximum-likelihood of marginal distribution as a two-player game. Both Veegan [14] and AVB [18] use statistical measures aiming to distinguish between joint distributions, similarly to BiGAN [15], but BiGANs do not actually have any connections to VAEs. The asymmetric KL divergence term from the objective function is replaced by its symmetric variant in the Adversarial Symmetric Variational Autoencoder [19], which searches for an adversarial solution to the VAE objective.

To summarize, adversarial learning in existing hybrid models can be performed into the data space [20], latent space [13] or into their joint spaces [21]. Lately, the likelihood estimation as a regularization term was shown to stabilize adversarial distribution matching [22]. The likelihood estimation is also employed to learn latent representations across the domain in mixture models [23]. However, all these methods only focus on improving the generation capability and do not design suitable objective functions for inducing disentangled representations.

This research study is the first to propose an appropriate objective function for training a hybrid VAE-GAN method for learning both continuous and discrete disentangled representations.

Self-supervised learning. Self-Supervised Learning (SSL) was used for classification [24], where the classifier is trained on the labelled augmented dataset to learn visual representations. SSL was also used for semi-supervised learning [25] producing successful results. However, most SSL methods focus on predictive tasks such as image classification, while applying SSL in generative tasks remains unexplored.

In this paper, we introduce a novel representation learning model which can acquire meaningful data representations in a self-supervised manner.

3. Self-Supervised Adversarial Variational Learning (SS-AVL)

In the following we describe the training of the proposed SS-AVL model.

3.1. Adversarial learning for the generator

Let $\mathbf{x} \in \mathbb{R}^d$ represent an observed random variable sampled from the empirical data distribution $\mathbb{P}_{\mathbf{x}}$. We assume that the generation process of $\mathbf{x}$ involves three underlying generative factors corresponding to continuous $\mathbf{z}, \mathbf{c}$ and discrete variables $\mathbf{d}$, respectively. These generative factors are sampled from three independent prior distributions $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{z}})$, $\mathbf{c} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{c}}, \boldsymbol{\Sigma}_{\mathbf{c}})$, $\mathbf{d} \sim Cat(k = K, p = 1/K)$, where $\boldsymbol{\mu}_{\mathbf{c}}$ and $\mu_{\mathbf{z}}$ are the mean vectors of Gaussian distributions, while $\boldsymbol{\Sigma}_{\mathbf{c}}$ and $\boldsymbol{\Sigma}_{\mathbf{z}}$ are identity covariance matrices. K represents the number of potential categories for the dataset and Cat is the categorical distribution. Let us define a generator $G_\psi(\mathbf{z}, \mathbf{d}, \mathbf{c})$, implemented by a neural network with parameters ψ. The generation process for the observed variable $\mathbf{x}$ is defined as :

$$\mathbf{d} \sim p(\mathbf{d}),\ \mathbf{z} \sim p(\mathbf{z}), c \sim p(\mathbf{c}), \mathbf{x} \sim q_\psi(\mathbf{x} \mid \mathbf{z}, \mathbf{d}, \mathbf{c}), \tag{3}$$

where $q_\psi(\mathbf{x} \mid \mathbf{z}, \mathbf{d}, \mathbf{c})$ can be seen as the generator $G_\psi(\mathbf{z}, \mathbf{d}, \mathbf{c})$ in GAN or as the decoder in VAE. One of our goals is to encourage the generator distribution $\mathbb{P}_G$ to match the real data distribution $\mathbb{P}_{\mathbf{x}}$ by using adversarial learning. Therefore, we introduce a discriminator network $D : \mathcal{X} \to \mathbb{R}^d$ and we consider the Wasserstein GAN (WGAN) loss [26], which is defined as the optimal path of transporting information mass from the generator distribution $\mathbb{P}_G$ to the data distribution $\mathbb{P}_{\mathbf{x}}$, corresponding to the Earth Mover Distance. By considering the Kantorovich–Rubinstein duality [27], the optimal transport adversarial learning is defined as :

$$\min_G \max_{D \in \Theta} \left\{ \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\mathbf{x}}}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{x}' \sim \mathbb{P}_G}[D(\mathbf{x}')] \right\}, \tag{4}$$

where Θ represents a set of 1-Lipschitz functions. We introduce a gradient penalty term [28] to enforce the Lipschitz constraint, resulting in :

$$\begin{aligned} \min_G \max_D \Big\{ & \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\mathbf{x}}}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{x}' \sim \mathbb{P}_G}[D(\mathbf{x}')] \Big\} \\ & + \lambda \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_{\tilde{\mathbf{x}}}}[(\left\|\nabla_{\tilde{x}} D(\tilde{\mathbf{x}})\right\|_2 - 1)^2], \end{aligned} \tag{5}$$

where $\mathbb{P}_{\tilde{\mathbf{x}}}$ is defined as sampling uniformly along straight lines between pairs of data originating from two distributions $\mathbb{P}_{\mathbf{x}}$ and $\mathbb{P}_G$. This training procedure is illustrated in Fig. 2.

3.2. Self-supervised learning for inference models

In this section, we introduce a novel algorithm that trains inference models in a self-supervised manner. The main goal of the proposed algorithm is to learn meaningful latent representations $\mathbf{z}$ and $\mathbf{u} = \{\mathbf{d}, \mathbf{c}\}$, respectively. The maximum log-likelihood has been used in the VAE framework [5,29] to learn generative factors of data by jointly optimizing both the generator and inference models. However, VAEs are only used for learning real samples and therefore its ability is limited when data are not available. In this section, we propose using inference models in the SS-AVL model which are optimized in a self-supervised learning manner.

Notations. Let $\mathbf{x}' \sim G(\tilde{\mathbf{z}}, \tilde{\mathbf{d}}, \tilde{\mathbf{c}})$ be the generated sample where $\tilde{\mathbf{z}}, \tilde{\mathbf{d}}, \tilde{\mathbf{c}}$ are latent variables sampled from the prior distributions $p(\tilde{\mathbf{z}})$, $p(\tilde{\mathbf{d}})$, $p(\tilde{\mathbf{c}})$. Let $q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x})$ and $q_\xi(\mathbf{z} \mid \mathbf{x})$ represent two independent conditional distributions implemented by two inference models with parameters ω and ξ, respectively. Let us define $\{\mathbf{d}, \mathbf{c}\}$ as interpretable representations which model discrete and continuous meaningful variations of the data $\mathbf{x}'$, and $\tilde{\mathbf{z}}$ as the observed variables. Let us define a latent variable model $p_\psi(\mathbf{x}', \tilde{\mathbf{z}}, \mathbf{d}, \mathbf{c}) = p_\psi(\mathbf{x}' \mid \tilde{\mathbf{z}}, \mathbf{d}, \mathbf{c}) p(\mathbf{d}, \mathbf{c}) p(\tilde{\mathbf{z}})$. The log-likelihood of $p_\psi(\mathbf{x}')$ is defined as:

$$\log p_\psi(\mathbf{x}') = \iiint \log p_\psi(\mathbf{x}' \mid \mathbf{d}, \mathbf{c}, \tilde{\mathbf{z}}) p(\mathbf{d}, \mathbf{c}) p(\tilde{\mathbf{z}})\, \mathrm{d}\mathbf{d}\, \mathrm{d}\mathbf{c}\, \mathrm{d}\tilde{\mathbf{z}}. \tag{6}$$

Eq. (6) is intractable since it requires to integrate over all latent variables. To address this problem, we introduce a variational distribution $q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}')$, and derive a lower bound on the model log-likelihood, called ELBO, by using the Jensen's inequality :

$$\begin{aligned} \log p_\psi(\mathbf{x}') &= \log \mathbb{E}_{q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}') p(\tilde{\mathbf{z}})} \left[\frac{p_\psi(\mathbf{x}', \mathbf{d}, \mathbf{c}, \tilde{\mathbf{z}})}{q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}') p(\tilde{\mathbf{z}})} \right] \\ &\geq \mathbb{E}_{q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}') p(\tilde{\mathbf{z}})} \left[\log \frac{p_\psi(\mathbf{x}', \mathbf{d}, \mathbf{c}, \tilde{\mathbf{z}})}{q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}') p(\tilde{\mathbf{z}})} \right] \end{aligned} \tag{7}$$

This can be rewritten, following a derivation provided in the following, as:

$$\begin{aligned} \log p_\psi(\mathbf{x}') &\geq \mathbb{E}_{q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}') p(\tilde{\mathbf{z}})} \left[\log \frac{p_\psi(\mathbf{x}' \mid \mathbf{d}, \mathbf{c}, \tilde{\mathbf{z}}) p(\mathbf{d}) p(\mathbf{c}) p(\tilde{\mathbf{z}})}{q_\omega(\mathbf{d} \mid \mathbf{x}') q_\omega(\mathbf{c} \mid \mathbf{x}') p(\tilde{\mathbf{z}})} \right] \\ &= \mathbb{E}_{q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}') p(\tilde{\mathbf{z}})} \left[\log \frac{p_\psi(\mathbf{x}' \mid \mathbf{d}, \mathbf{c}, \tilde{\mathbf{z}}) p(\mathbf{d}) p(\mathbf{c})}{q_\omega(\mathbf{d} \mid \mathbf{x}') q_\omega(\mathbf{c} \mid \mathbf{x}')} \right] \\ &= \mathbb{E}_{q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}') p(\tilde{\mathbf{z}})} \left[\log p_\psi(\mathbf{x}' \mid \mathbf{d}, \mathbf{c}, \tilde{\mathbf{z}}) \right] \\ &+ \mathbb{E}_{q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}') p(\tilde{\mathbf{z}})} \left[\log \frac{p(\mathbf{d})}{q_\omega(\mathbf{d} \mid \mathbf{x}')} \right] + \mathbb{E}_{q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}') p(\tilde{\mathbf{z}})} \left[\log \frac{p(\mathbf{c})}{q_\omega(\mathbf{c} \mid \mathbf{x}')} \right] \\ &= \mathbb{E}_{q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}') p(\tilde{\mathbf{z}})} [\log p_\psi(\mathbf{x}' \mid \mathbf{d}, \mathbf{c}, \tilde{\mathbf{z}})] \\ &- D_{KL}(q_\omega(\mathbf{d} \mid \mathbf{x}') \,||\, p(\mathbf{d})) - D_{KL}(q_\omega(\mathbf{c} \mid \mathbf{x}') \,||\, p(\mathbf{c})). \end{aligned} \tag{8}$$

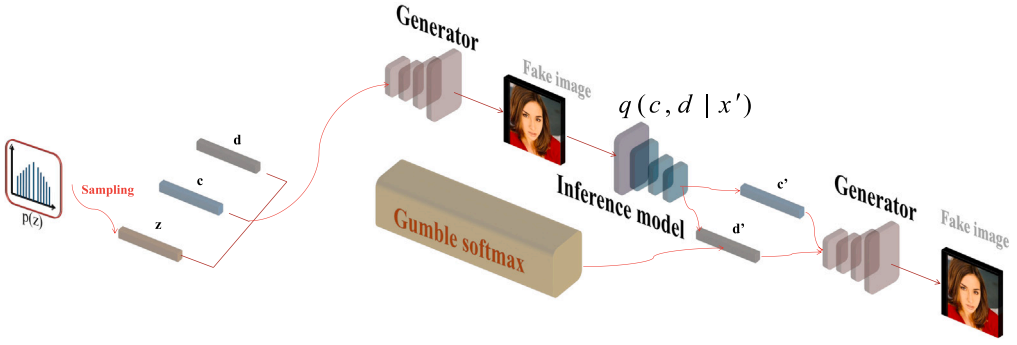
$$\begin{aligned} \log p_\psi(\mathbf{x}') \geq\ & \mathbb{E}_{q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}') p(\tilde{\mathbf{z}})} [\log p_\psi(\mathbf{x}' \mid \mathbf{d}, \mathbf{c}, \tilde{\mathbf{z}})] \\ & - D_{KL}(q_\omega(\mathbf{d} \mid \mathbf{x}') \,\|\, p(\mathbf{d})) - D_{KL}(q_\omega(\mathbf{c} \mid \mathbf{x}') \,\|\, p(\mathbf{c})). \end{aligned} \tag{9}$$

We consider that $\tilde{\mathbf{z}}$ is independent from $\mathbf{d}$ and $\mathbf{c}$ since $\tilde{\mathbf{z}}$ represents the prior information about $\mathbf{x}'$. Meanwhile, we assume that $\mathbf{d}$ is also independent from $\mathbf{c}$. In order to sample from discrete distributions, such as $\mathbf{d}$, we use the Gumbel-softmax distribution [30] which is differentiable and can be embedded in the SGD training algorithm. $q_\omega(\mathbf{d} \mid \mathbf{x}')$ can be implemented by a convolution network with the last layer consisting of the softmax function computing a probability vector $(a_1, \ldots, a_K)$, where K is the length of the discrete representation :

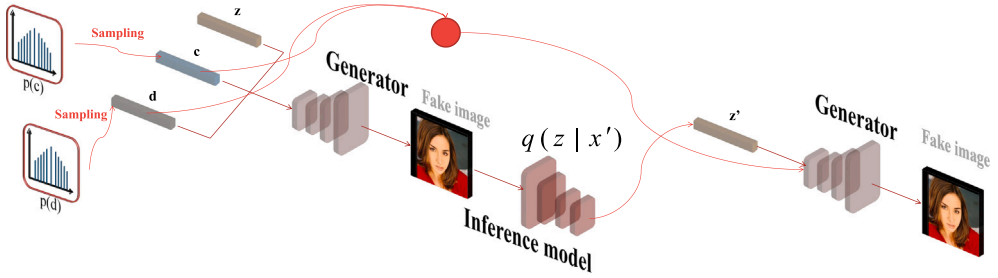
$$d_k = \frac{\exp((\log a_k + \mathbf{g}_k)/T)}{\sum_{i=1}^{K} \exp((\log a_i + \mathbf{g}_i)/T)} \tag{10}$$

where $\mathbf{g}_k$ is sampled from $\mathrm{Gumbel}(0, 1)$ and T is the temperature parameter controlling the smoothness of the Gumbel-softmax function.

Eq. (9) is only used to optimize the inference models with respect to the latent variables $\{\mathbf{d}, \mathbf{c}\}$. The advantage of this optimization is that $\{\mathbf{d}, \mathbf{c}\}$ would capture invariant representations of $\mathbf{x}'$ since this optimization considers $\tilde{\mathbf{z}}$ as the prior information which is drawn from



(a) The optimization of $q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x})$, defined by Eq. (9).



(b) The optimization of $q_\xi(\mathbf{z} \mid \mathbf{x})$, defined by Eq. (12) .

Fig. 1. Unsupervised learning for the inference module. We separately optimize the inference modules while the generator is fixed.

an independent distribution. Therefore, $\mathbf{x}'$ would not change too much when changing $\tilde{\mathbf{z}}$ while fixing $\{\mathbf{d}, \mathbf{c}\}$.

In the following, we describe how to optimize the inference model with respect to the latent variable $\mathbf{z}$. Let $q_\xi(\mathbf{z} \mid \mathbf{x}')$ be a variational distribution of parameters ξ. Similarly, we define a latent variable model $p_\psi(\mathbf{x}', \mathbf{z}, \tilde{\mathbf{d}}, \tilde{\mathbf{c}}) = p_\psi(\mathbf{x}' \mid \mathbf{z}, \tilde{\mathbf{d}}, \tilde{\mathbf{c}}) p(\mathbf{z}) p(\tilde{\mathbf{d}}, \tilde{\mathbf{c}})$, where we treat $\mathbf{x}', \tilde{\mathbf{c}}, \tilde{\mathbf{d}}$ as observed variables and $\mathbf{z}$ as an unobserved vector. Then we can define the model's log-likelihood and its ELBO as :

$$\begin{aligned} \log p_\psi(\mathbf{x}') &= \log \mathbb{E}_{q_\xi(\mathbf{z}|\mathbf{x}')p(\tilde{\mathbf{d}})p(\tilde{\mathbf{c}})} \left[\frac{p_\psi(\mathbf{x}', \tilde{\mathbf{d}}, \tilde{\mathbf{c}}, \mathbf{z})}{q_\xi(\mathbf{z} \mid \mathbf{x}') p(\tilde{\mathbf{d}}) p(\tilde{\mathbf{c}})} \right] \\ &\geq \mathbb{E}_{q_\xi(\mathbf{z}|\mathbf{x}')p(\tilde{\mathbf{d}})p(\tilde{\mathbf{c}})} \left[\log \frac{p_\psi(\mathbf{x}', \tilde{\mathbf{d}}, \tilde{\mathbf{c}}, \mathbf{z})}{q_\xi(\mathbf{z} \mid \mathbf{x}') p(\tilde{\mathbf{d}}) p(\tilde{\mathbf{c}})} \right]. \end{aligned} \tag{11}$$

Then we rewrite this expression as :

$$\log p_\psi(\mathbf{x}') \geq \mathbb{E}_{q_\xi(\mathbf{z}|\mathbf{x}')p(\tilde{\mathbf{d}})p(\tilde{\mathbf{c}})}[\log p_\psi(\mathbf{x}' \mid \tilde{\mathbf{d}}, \tilde{\mathbf{c}}, \mathbf{z})] - D_{KL}(q_\xi(\mathbf{z} \mid \mathbf{x}') \,||\, p(\mathbf{z})), \tag{12}$$

where $\tilde{\mathbf{c}}$ and $\tilde{\mathbf{d}}$ are drawn from the prior distributions, $p(\tilde{\mathbf{c}})$ and $p(\tilde{\mathbf{d}})$, respectively. In practice, Eqs. (9) and (12) can be optimized considering adversarial learning, which would allow us to learn meaningful representations from a pre-trained GAN model without data. The architectures implementing Eqs. (9) and (12) are shown in Fig. 1a and b, respectively. We introduce the detailed training process in Section 6.

4. Theoretical framework

In existing hybrid VAE-GAN models, the inference model and generator network are trained jointly by using a single objective function. This optimization process requires to access real data samples for both the inference model and generator, which cannot be done in a pre-trained GAN model. Additionally, this optimization process would result in the degradation of the model's generated data quality. The training of SS-AVL, illustrated in Fig. 1, has several different aspects when compared to other VAE-GAN models, by optimizing separately the inference models using two different loss functions, defined by Eqs. (9) and (12). The proposed training for SS-AVL provides many advantages. For instance, the training of the inference model does not interfere with the optimization of the generator, which consequently would provide a stable training procedure. When the generator would model exactly the true data distribution, we can also derive more accurate inference modules. It is easier to achieve the matching of two single distributions individually, than aligning two joint distributions using adversarial learning, as in other VAE-GAN based approaches [18]. Unlike in InfoGAN [31], the proposed model has a full inference mechanism, which enables the inference of both meaningful and nuisance latent representations, benefiting many down-stream tasks such as data reconstruction and interpolation.

Proposition 1. *Let G^* be the optimal solution of the Wasserstein distance $W(\mathbb{P}_\mathbf{x}, \mathbb{P}^*_\psi)$ and consequently we have $\mathbb{P}_\mathbf{x} \approx \mathbb{P}^*_\psi$, where $\mathbb{P}^*_\psi$ is the distribution of data generated by G^*. Let $\mathbb{P}_{X,Z}$ be the coupling between observed data $\mathbf{x}$ and the latent variables $\mathbf{z}$, inferred by the optimal inference model $q^*_\xi(\mathbf{z} \mid \mathbf{x})$. Based on the assumption that $\mathbb{P}_\mathbf{x} \approx \mathbb{P}^*_\psi$, $q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}')$ adapts the generator to maximize a lower bound on the real data log-likelihood.*

Proof. We can rewrite $\log p_\psi(\mathbf{x}')$ from (9), as :

$$\begin{aligned} \log p_{\psi^*}(\mathbf{x}^*) \geq\ & \mathbb{E}_{q_\omega(\mathbf{d},\mathbf{c}|\mathbf{x}^*)p(\tilde{\mathbf{z}})}[\log p_{\psi^*}(\mathbf{x}^* \mid \mathbf{d}, \mathbf{c}, \tilde{\mathbf{z}})] - D_{KL}(q_\omega(\mathbf{c} \mid \mathbf{x}^*) \,||\, p(\mathbf{c})) \\ & - D_{KL}(q_\omega(\mathbf{d}|\mathbf{x}^*) \,||\, p(\mathbf{d})) = \mathcal{L}(\psi^*, \omega; \mathbf{x}^*), \end{aligned} \tag{13}$$

where $\tilde{\mathbf{z}}$ is the prior latent variable used for sampling $\mathbf{x}^*$ from $G^*(\tilde{\mathbf{z}}, \mathbf{d}, \mathbf{c})$. We only consider $\tilde{\mathbf{z}}$ to be available during the decoding phase since $\mathbf{d}, \mathbf{c}$ are treated as unobserved variables in the case when optimizing $q_\omega(\mathbf{d}, \mathbf{c} \mid \mathbf{x}^*)$. Let us define $\mathcal{L}(\tilde{\psi}, \tilde{\omega}; \mathbf{x})$ as the optimal lower bound on $\log p_\psi(\mathbf{x})$, where :

$$(\tilde{\psi}, \tilde{\omega}) = \arg \max_{\psi \in \Theta, \omega \in \Phi} \mathcal{L}(\psi, \omega; \mathbf{x}). \tag{14}$$

Then we have :

$$\sum_{i=1}^{n} \mathcal{L}(\tilde{\psi}, \tilde{\omega}; \mathbf{x}_i) \geq \sum_{i=1}^{n} \mathcal{L}(\psi^*, \omega; \mathbf{x}_i) \rightarrow \log p_\psi(\mathbf{x}) \geq \mathcal{L}(\psi^*, \omega; \mathbf{x}), \tag{15}$$

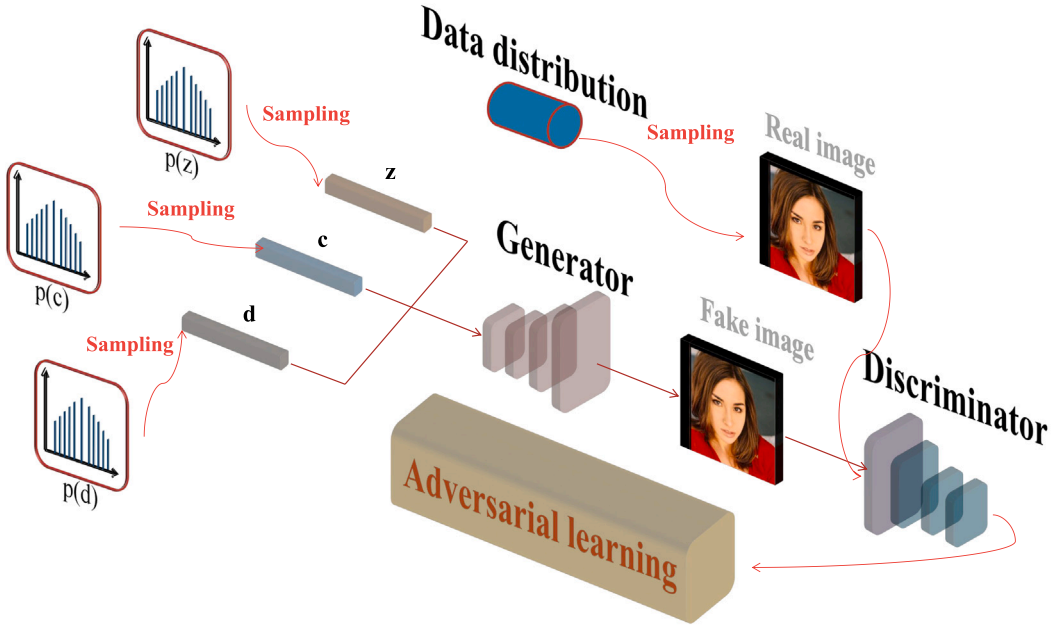


Fig. 2. Unsupervised learning structures in the generative models. **c** and **d** are the continuous and discrete variables, while **z** represents Gaussian noise.

given that $\{\tilde{\psi}, \tilde{\omega}\}$ are the optimal parameters for the model log-likelihood $\log p_{\psi}(\mathbf{x})$. We assume that $\mathbb{P}_{\psi^*}$ has an optimal behaviour, which allows it to gradually approximate $\mathbb{P}_{\mathbf{x}}$. Under this assumption, $\mathcal{L}(\psi^*, \omega; \mathbf{x})$ is increased towards $\mathcal{L}(\tilde{\psi}, \tilde{\omega}; \mathbf{x})$ during the learning's convergence. This shows that when the generator is an exact approximation for $\mathbb{P}_{\mathbf{x}}$ and we have the optimal coupling $\mathbb{P}_{X,Z}$ optimizing $q_{\omega}(\mathbf{d}, \mathbf{c} \mid \mathbf{x}^*)$, it results in the maximization of a lower bound on the sample log-likelihood. □

Proposition 2. *Following the definitions from Proposition 1 and assuming that the optimal coupling $\mathbb{P}_{X,D,C}$ exists, $q_{\xi}(\mathbf{z} \mid \mathbf{x}')$ is trained to be an adapter to the generator in order to maximize a lower bound on the real sample log-likelihood.*

Proof. When optimizing $q_{\xi}(\mathbf{z} \mid \mathbf{x}')$, we only consider accessing $\tilde{\mathbf{d}}, \tilde{\mathbf{c}}$ during the decoding phase. We can rewrite $\log p_{\psi}(\mathbf{x}')$ as :

$$\begin{aligned}\log p_{\psi^*}(\mathbf{x}^*) \geq & \mathbb{E}_{q_{\xi}(\mathbf{z}|\mathbf{x}^*)p(\tilde{d})p(\tilde{\mathbf{c}})}[\log p_{\psi^*}(\mathbf{x}^* \mid \mathbf{z}, \tilde{\mathbf{d}}, \tilde{\mathbf{c}})] \\ & - D_{KL}[q_{\xi}(\mathbf{z} \mid \mathbf{x}^*) \,||\, p(\mathbf{z})] = \mathcal{L}(\psi^*, \xi; \mathbf{x}^*) \quad \square\end{aligned} \tag{16}$$

Similarly to Proposition 1, we define $\{\tilde{\psi}, \tilde{\xi}\} = \arg\max_{\psi \in \Theta, \xi \in \Gamma} \mathcal{L}(\psi, \xi; x)$ as the optimal parameters and its optimal lower bound is $\mathcal{L}(\tilde{\psi}, \tilde{\xi}; \mathbf{x})$, then we have for n data samples, $\mathbf{x}_i$, $i = 1, \ldots, n$:

$$\sum_{i=1}^{n} \mathcal{L}(\tilde{\psi}, \tilde{\varsigma}; \mathbf{x}_i) \geq \sum_{i=1}^{n} \mathcal{L}(\psi^*, \xi; \mathbf{x}_i) \rightarrow \log p_{\psi}(\mathbf{x}) \geq \mathcal{L}(\psi^*, \xi; \mathbf{x}) \tag{17}$$

The conclusions of Propositions 1 and 2, support the hypothesis that by maximizing the log-likelihood objective function, the model is able to learn accurate data representations when the generator approximates well the empirical data distribution $\mathbb{P}_{\mathbf{x}}$. In practice, the inference model provides a flexible learning manner in which the networks can be trained individually.

Proposition 3. *For a given well-trained inference model, we can estimate the sample log-likelihood for data $\mathbf{x}_t$, generated by a GAN, as :*

$$\begin{aligned}\log p_{\psi}(\mathbf{x}_t) \geq & \mathbb{E}_{q_{\omega,\xi}(\mathbf{d},\mathbf{c},\mathbf{z}|\mathbf{x}_t)}[\log p_{\psi}(\mathbf{d}, \mathbf{c}, \mathbf{z} \mid \mathbf{x}_t)] - D_{KL}(q_{\omega}(\mathbf{d} \mid \mathbf{x}_t) \,||\, p(\mathbf{d})) \\ & - D_{KL}(q_{\omega}(\mathbf{c} \mid \mathbf{x}_t) \,||\, p(\mathbf{c})) \\ & - D_{KL}(q_{\xi}(\mathbf{z} \mid \mathbf{x}_t) \,||\, p(\mathbf{z})) = \mathcal{L}(\psi, \xi, \omega; \mathbf{x}_t)\end{aligned} \tag{18}$$

The model $p_{\psi}(\mathbf{x}_t)$ combines the two inference models from Fig. 1 and a generator.

Proposition 3 provides an explicit way to estimate the data sample log-likelihood in GAN models, when providing a pre-trained model.

Proof. We combine the two inference models for continuous and discrete variables, and a generator as a single model $\log p_{\psi}(\mathbf{x}_t) = p_{\psi}(\mathbf{x}_t \mid \mathbf{d}, \mathbf{c}, \mathbf{z}) q_{\omega}(\mathbf{d}, \mathbf{c} \mid \mathbf{x}_t) q_{\xi}(\mathbf{z} \mid \mathbf{x}_t)$. Then we define the model log-likelihood as :

$$\log p_{\psi}(\mathbf{x}_t) = \log \mathbb{E}_{q_{\omega,\xi}(\mathbf{d},\mathbf{c},\mathbf{z}|\mathbf{x}_t)} \left[\frac{p_{\psi}(\mathbf{x}_t, \mathbf{d}, \mathbf{c}, \mathbf{z})}{q_{\omega,\xi}(\mathbf{d}, \mathbf{c}, \mathbf{z} \mid \mathbf{x}_t)} \right]. \tag{19}$$

According to the Jensen inequality, we have :

$$\begin{aligned}\log p_{\psi}(\mathbf{x}_t) \geq & \mathbb{E}_{q_{\omega,\xi}(\mathbf{d},\mathbf{c},\mathbf{z}|\mathbf{x}_t)} \left[\log \frac{p_{\psi}(\mathbf{x}_t, \mathbf{d}, \mathbf{c}, \mathbf{z})}{q_{\omega,\xi}(\mathbf{d}, \mathbf{c}, \mathbf{z} \mid \mathbf{x}_t)} \right] \\ = & \mathbb{E}_{q_{\omega,\xi}(\mathbf{d},\mathbf{c},\mathbf{z}|\mathbf{x}_t)} \left[\log \frac{p_{\psi}(\mathbf{d}, \mathbf{c}, \mathbf{z} \mid \mathbf{x}_t) p(\mathbf{d}) p(\mathbf{c}) p(\mathbf{z})}{q_{\omega}(\mathbf{d}|\mathbf{x}_t) q_{\omega}(\mathbf{c} \mid \mathbf{x}_t) q_{\xi}(\mathbf{z}|\mathbf{x}_t)} \right] \\ = & \mathbb{E}_{q_{\omega,\xi}(\mathbf{d},\mathbf{c},\mathbf{z}|\mathbf{x}_t)} \left[\log p_{\psi}(\mathbf{d}, \mathbf{c}, \mathbf{z} \mid \mathbf{x}_t) \right] + \mathbb{E}_{q_{\omega,\xi}(\mathbf{d},\mathbf{c},\mathbf{z}|\mathbf{x}_t)} \left[\log \frac{p(\mathbf{d})}{q_{\omega}(\mathbf{d}|\mathbf{x}_t)} \right] \\ & + \mathbb{E}_{q_{\omega,\xi}(\mathbf{d},\mathbf{c},\mathbf{z}|\mathbf{x}_t)} \left[\log \frac{p(\mathbf{c})}{q_{\omega}(\mathbf{c} \mid \mathbf{x}_t)} \right] + \mathbb{E}_{q_{\omega,\xi}(\mathbf{d},\mathbf{c},\mathbf{z}|\mathbf{x}_t)} \left[\log \frac{p(\mathbf{z})}{q_{\xi}(\mathbf{z} \mid \mathbf{x}_t)} \right] \\ = & \mathbb{E}_{q_{\omega,\xi}(\mathbf{d},\mathbf{c},\mathbf{z}|\mathbf{x}_t)} \left[\log p_{\psi}(\mathbf{d}, \mathbf{c}, \mathbf{z} \mid \mathbf{x}_t) \right] - D_{KL}(q_{\omega}(\mathbf{d} \mid \mathbf{x}_t) \,||\, p(\mathbf{d})) \\ & - D_{KL}(q_{\omega}(\mathbf{c}|\mathbf{x}_t) \,\|\, p(\mathbf{c})) - D_{KL}(q_{\xi}(\mathbf{z}|\mathbf{x}_t) \,\|\, p(\mathbf{z})) \quad \square\end{aligned} \tag{20}$$

5. Mutual information maximization for interpretable representations

In information theory, mutual information (MI) measures the amount of information shared by one random variable when observing another variable. In the proposed SS-AVL model, we aim to transfer the underlying characteristics of the continuous and discrete latent variables during the decoder-generation process.

Let us denote the joint latent variables by $\mathbf{u} = (\mathbf{d}, \mathbf{c})$ and we want to maximize the MI between the joint latent variable $\mathbf{u}$ and the decoder result, $\mathrm{I}(\mathbf{u}, G(\mathbf{z}, \mathbf{u}))$. Similar MI objectives have been adopted in [21,31–35]. According to these studies, it is difficult to optimize the mutual information directly given that it needs to access the information represented by the true posterior $p(\mathbf{u} \mid \mathbf{x})$. In order to address this problem,

we define an auxiliary distribution $W(\mathbf{u}\,|\,\mathbf{x})$ to approximate the true posterior and then derive a lower bound on the mutual information, expressed using the marginal entropy $H(\mathbf{u})$, as well as the conditional entropy, $H(\mathbf{u}\,|\,G(\mathbf{z},\mathbf{u}))$:

$$\begin{aligned}\mathrm{I}(\mathbf{u}, G(\mathbf{z},\mathbf{u})) &= H(\mathbf{u}) - H(\mathbf{u}\mid G(\mathbf{z},\mathbf{u}))\\ &= \iint G(\mathbf{z},\mathbf{u})p(\mathbf{u}\mid\mathbf{x})\log\frac{p(\mathbf{u}\mid\mathbf{x})}{W(\mathbf{u}\mid\mathbf{x})}\,d\mathbf{x}d\mathbf{u}\\ &+ \iint G(\mathbf{z},\mathbf{u})p(\mathbf{u}\mid\mathbf{x})\log W(\mathbf{u}\mid\mathbf{x})d\mathbf{x}d\mathbf{u} + H(\mathbf{u})\\ &= \mathbb{E}_{\mathbf{x}\sim G(\mathbf{z},\mathbf{u})} D_{KL}[p(\mathbf{u}\mid\mathbf{x})\,||\,W(\mathbf{u}\mid\mathbf{x})]\\ &+ \mathbb{E}_{\mathbf{x}\sim G(\mathbf{z},\mathbf{u})}[\mathbb{E}_{\mathbf{u}\sim p(\mathbf{u},\mathbf{x})}[\log W(\mathbf{u}\mid\mathbf{x})]] + H(\mathbf{u})\\ &\geqslant \mathbb{E}_{\mathbf{x}\sim G(\mathbf{z},\mathbf{u})}[\mathbb{E}_{\mathbf{u}\sim p(\mathbf{u},\mathbf{x})}[\log W(\mathbf{u}\mid\mathbf{x})]] + H(\mathbf{u}) = \mathcal{L}_{MI}\,,\end{aligned} \tag{21}$$

where the auxiliary distribution $W(\mathbf{u}\,|\,\mathbf{x})$ is implemented by $q_\omega(\mathbf{d},\mathbf{c}\,|\,\mathbf{x})$. In practice, we sample a pair of latent variables $\mathbf{d},\mathbf{c}$ from $q_\omega(\mathbf{d},\mathbf{c}\,|\,\mathbf{x})$. We estimate the mutual information by means of the lower bound $\mathcal{L}_{MI}$, from (21). The last term, $H(\mathbf{u})$ represents the marginal entropy of the latent variables and is considered as a constant for simplicity.

6. The two-stage algorithm and implementation details

The structure implementing SS-AVL model is shown in Figs. 2 and 1. $q_\omega(\mathbf{d}\,|\,\mathbf{x})$ and $q_\omega(\mathbf{c}\,|\,\mathbf{x})$ are implemented by the same network except for the last layer which is different for the inference of each latent variable, as it can be seen in Fig. 1a. $q_\xi(\mathbf{z}\,|\,\mathbf{x})$ is implemented by a neural network with trainable parameters ξ, as shown in Fig. 1b. We introduce a two-stage algorithm to train the inference and generator, separately. The proposed algorithm has two independent optimization stages, named"wake" and "dreaming".

In the "wake" phase, we optimize the discriminator and generator by the following loss functions :

$$\mathcal{L}_D = \nabla_\varphi \frac{1}{m}\sum_{i=1}^{m} -[D(\mathbf{x}_i) - D(G(\mathbf{z}_i,\mathbf{c}_i,\mathbf{d}_i))], \tag{22}$$

$$\mathcal{L}_G = -\nabla_\psi \frac{1}{m}\sum_{i=1}^{m} D(G(\mathbf{z}_i,\mathbf{c}_i,\mathbf{d}_i)), \tag{23}$$

where m represents the number of data samples used in the batch during SGD learning.

In the "dreaming" phase, we maximize the MI between latent representations and observed data by optimizing the parameters of the generator, discriminator and inference model with respect to their gradients :

$$\begin{aligned}\mathcal{L}_{MI} = \nabla_{\varphi,\omega,\psi}\Big(&\mathbb{E}_{\mathbf{x}\sim G(\mathbf{z},\mathbf{u})}[\mathbb{E}_{\mathbf{d}\sim p(\mathbf{d},\mathbf{x})}[\log q_\omega(\mathbf{d}\mid\mathbf{x}_i)]]\\ &+ \mathbb{E}_{\mathbf{x}\sim G(\mathbf{z},\mathbf{u})}[\mathbb{E}_{\mathbf{c}\sim p(\mathbf{c},\mathbf{x})}[\log q_\omega(\mathbf{c}\mid\mathbf{x})]]\Big)\,.\end{aligned} \tag{24}$$

It can be seen from Figs. 2 and 1, that unlike the approach from BiGAN [15], where the inference model is trained with respect to gradient signals from the discriminator, the proposed algorithm optimizes the parameters of the inference model in the "dreaming" phase by using two different objective functions, whose parameters are updated by maximizing the marginal log-likelihood objective functions over the observed samples drawn during the "wake" phase from the current generator distribution. The proposed algorithm can allow the inference models to be trained independently in order to enforce disentanglement between $\mathbf{z}$ and $(\mathbf{c},\mathbf{d})$:

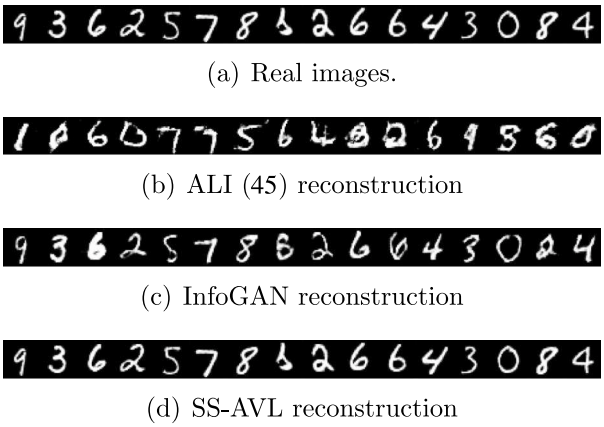
$$\begin{aligned}&\nabla_\omega\Big(\mathbb{E}_{\mathbf{d},\mathbf{c}\sim q_\omega(\mathbf{d},\mathbf{c}|\mathbf{x}'),\tilde{\mathbf{z}}\sim p(\tilde{\mathbf{z}})}[\log p_\psi(\mathbf{x}'\mid\mathbf{d},\mathbf{c},\tilde{\mathbf{z}})] - D_{KL}(q_\omega(\mathbf{d}\mid\mathbf{x}')\,||\,p(\mathbf{d}))\\ &- D_{KL}(q_\omega(\mathbf{c}\mid\mathbf{x}')\,||\,p(\mathbf{c}))\Big),\end{aligned} \tag{25}$$

$$\nabla_\xi\Big(\mathbb{E}_{\mathbf{z}\sim q_\omega(\mathbf{z}|\mathbf{x}'),\tilde{\mathbf{d}}\sim p(\tilde{\mathbf{d}}),\tilde{\mathbf{c}}\sim p(\tilde{\mathbf{c}})}[\log p_\psi(\mathbf{x}'\mid\tilde{\mathbf{d}},\tilde{\mathbf{c}},\mathbf{z})] - D_{KL}(q_\xi(\mathbf{z}\mid\mathbf{x}')\,||\,p(\mathbf{z}))\Big)\,. \tag{26}$$

The pseudo-code of the supervised training algorithm is provided in Algorithm 1.

Algorithm 1: The supervised learning for SS-AVL

Input: Training database
Output: Model's parameters
for $epoch < Epoch^{max}$ **do**
  for $j < batchCount$ **do**
    $\mathbf{x}_{batch} \sim pd(\mathbf{x})$ Sample from the data distribution ;
    $\mathbf{z}\sim p(\mathbf{z}), \mathbf{c}\sim p(\mathbf{c}), \mathbf{d}\sim p(\mathbf{d})$ Latent random vectors drawn from the prior distributions ;
    $\mathbf{x}_g = G(\mathbf{z},\mathbf{c},\mathbf{d})$ Generate images for prior distributions ;
    $\mathbf{z}' \sim q_\xi(\mathbf{z}\,|\,\mathbf{x}_g), \mathbf{c}', \mathbf{d}' = q_\omega(\mathbf{c},\mathbf{d}\,|\,\mathbf{x}_g)$ Infer latent variables from generated images ;
    $\mathbf{x}' = G(\mathbf{z},\mathbf{c}',\mathbf{d}'), \tilde{\mathbf{x}} = G(\mathbf{z}',\mathbf{c},\mathbf{d})$ Reconstruct images ;
    Wake phase ;
    Update discriminator network by $\mathcal{L}_D$;
    Update generator by $\mathcal{L}_G$;
    Dreaming phase ;
    Update all components by $\mathcal{L}_{MI}$;
    Update the encoder by the two loss functions (Eq. (25) and Eq. (26)) ;
  end
end



(a) Real images.

(b) ALI (45) reconstruction

(c) InfoGAN reconstruction

(d) SS-AVL reconstruction

Fig. 3. Reconstruction results on MNIST.

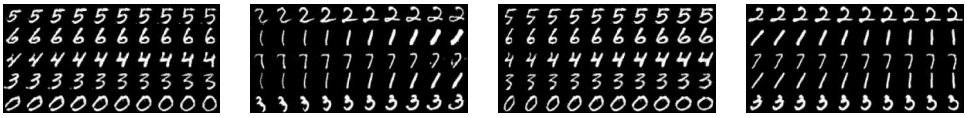
7. Discussion

One downside of the proposed SS-AVL is that the mutual information maximization from Eq. (21) requires updating jointly the inference model and generator on real training samples to encourage learning interpretable representations. However, when real training samples are not available, mutual information optimization will not be applicable. In addition, in a more realistic learning environment where the past datasets are unavailable when learning a new dataset [37], the proposed SS-AVL would quickly forget its previously learnt knowledge, leading to significant performance degeneration on past datasets. A simple approach to relieving forgetting is to employ the generation property of SS-AVL, which can produce past knowledge that is incorporated with new samples for learning a new task. This approach will be investigated in a future study.

Another downside of this proposed SS-AVL methodology is that it cannot always learn optimal latent representations when the generator does not approximate the real data distribution exactly. When the generator is trained on a more complex data distribution, the model could easily suffer from the mode collapse [14], resulting in reproducing images only from certain categories. In order to address this drawback, an appropriate hyperparameter configuration can be adopted. In addition, several procedures have been lately proposed to

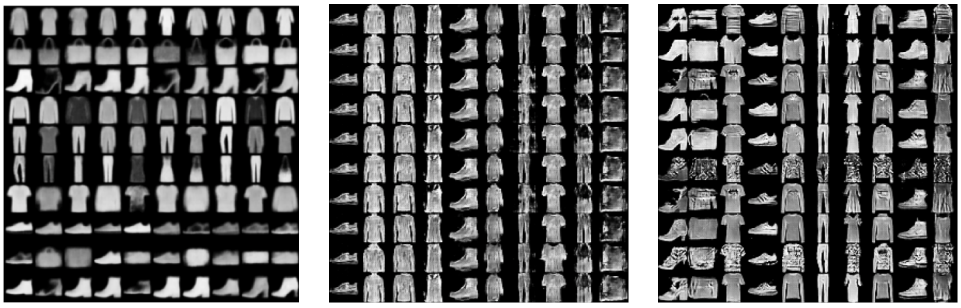
(a) JointVAE (36). (b) InfoGAN (31). (c) SS-AVL.

Fig. 4. Generation results on MNIST when the discrete variable is manipulated and the continuous variable is fixed. The visual results of JointVAE are those from [36].



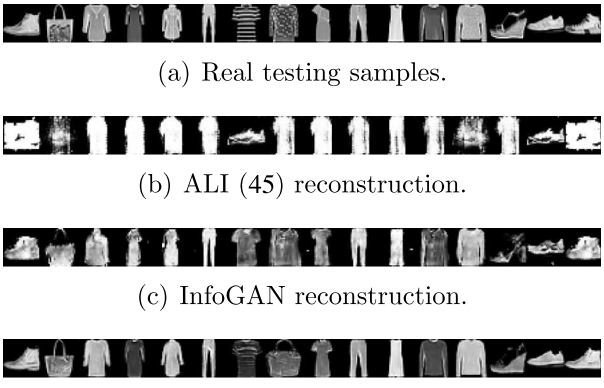
(a) InfoGAN c_1. (b) SS-AVL c_1. (c) InfoGAN c_2. (d) SS-AVL c_2.

Fig. 5. Generation results when changing continuous variables c_1 and c_2 from −1 to 1.



(a) JointVAE (36). (b) InfoGAN (31). (c) SS-AVL.

Fig. 6. Generation results when training on the Fashion dataset and considering changing the discrete variable while the continuous variable is fixed. The visual results of JointVAE are those from [36].



(a) Real testing samples.

(b) ALI (45) reconstruction.

(c) InfoGAN reconstruction.

(d) SS-AVL reconstruction.

Fig. 7. Reconstruction results when training on Fashion dataset.

stabilize the GAN training [14,38], which can relieve the mode collapse issue. Since the proposed SS-AVL does not interfere with the adversarial learning, SS-AVL can be smoothly combined with these approaches to further enhance the performance, which will be investigated in our future work.

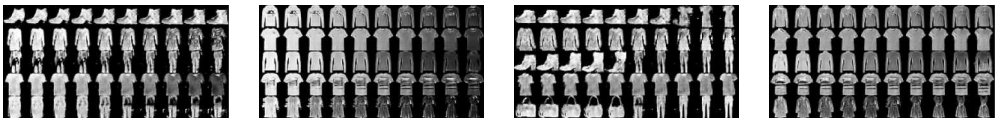
8. Experimental results

In the following we test and evaluate the performance and properties of SS-AVL on a variety of datasets while comparing it with the state-of-the-art.

8.1. Results on MNIST and MNIST-fashion databases

We firstly consider the MNIST dataset [39], representing images of handwritten digits grouped in 10 classes. In order to learn the discrete latent variable which captures different styles of the handwritten digits, we choose a categorical vector sampled from $\mathbf{c} \sim Cat(K = 10, p = 0.1)$ and two continuous variables sampled from a uniform distribution $\mathbf{z}, \mathbf{d} \sim U(-1, 1)$, where we consider each entry of the latent feature vectors being sampled from the corresponding distribution for the proposed Self-Supervised Adversarial Variational Learning (SS-VAL) as well as for InfoGAN [31], which is used for comparison.

We provide the reconstruction results on MNIST, achieved by the proposed SS-AVL in Fig. 3d, where the discrete latent variables are sampled from the Gumble-softmax distribution while the continuous latent variables are sampled from a Gaussian distribution, whose mean and diagonal covariance are parameterized by the encoder, and then take the latent variables as the input for the generator. In Fig. 3b and Fig. 3c we provide the reconstruction results for the Adversarially Learned Inference (ALI) [45] and InfoGAN [31], respectively. From



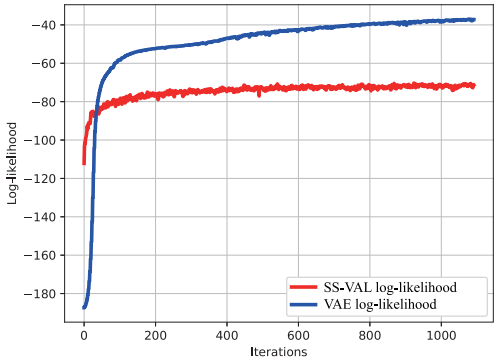
(a) InfoGAN c_1. (b) SS-AVL c_1. (c) InfoGAN c_2. (d) SS-AVL c_2.

Fig. 8. Generation results when changing continuous variables c_1 and c_2 from −1 to 1.

Table 1
Unsupervised classification results. M represents the number of different runs and K is the number of mixtures' components.

MNIST				
Method	K	M	Mean Accuracy	Best Accuracy
SS-AVL	1	4	95.42	96.15
JointVAE [36]	1	4	71.53	87.32
SubGAN [40]	20	4	89.72	90.81
InfoGAN [31]	1	4	91.98	93.35
GMVAE [41]	30	4	87.45	89.27
GMVAE [41]	16	4	87.12	87.82
AAE [13]	16	4	88.82	90.45
CatGAN [42]	30	4	93.67	95.73
DEC [43]	10	4	83.26	84.30
PixelGAN [44]	30	4	93.57	94.73
MNIST-Fashion				
SS-AVL	1	4	51.62	52.53
JointVAE [36]	1	4	48.57	49.83
InfoGAN	1	4	38.63	39.80

(a) Testing samples. (b) Reconstructions.



(c) Log-likelihood estimation during training.

Fig. 9. Results produced by a fixed GAN considering a trained inference model.

these results, it can be observed that SS-AVL provides better digit image reconstructions than either ALI or InfoGAN.

In another experiment, we change the discrete latent variable from 0 to 9 and sample the continuous variables **c** from the uniform distribution $U(-1, 1)$. The results generated when considering MNIST database for training are presented in Figs. 4a, 4b and 4c, for JointVAE, InfoGAN and the proposed SS-AVL, respectively. From Fig. 4c we observe that SS-AVL can generate digits showing different handwritten style characteristics, which cannot be said about the InfoGAN results, which sometimes would generate digit images which do not correspond to what is expected, as it can be observed from Fig. 4b. In the following we sample the continuous codes c_1, c_2 within $[-1, 1]$ and fix the other latent variables. The results generated when considering training with MNIST data are shown in Figs. 5b and 5d for SS-AVL, while those for InfoGAN are provided in Figs. 5a and 5c, when changing either c_1 or c_2. It can be observed that by varying the latent codes in SS-AVL, we generate images showing meaningful characteristics such as digits being rotated or displaying various handwriting styles.

The Fashion dataset, which contains images of clothing items, is a more challenging database for training than MNIST and we consider the same hyperparameters as for MNIST when training SS-AVL and InfoGAN for comparison. The generation results across all Fashion classes are presented in Fig. 6, where we change the discrete latent variable from 0 to 9. The original images are provided in Fig. 7a and the reconstructions by ALI, InfoGAN and SS-AVL are shown in Figs. 7b, 7c and 7d, respectively. We also sample the continuous codes c_1, c_2 within $[-1, 1]$ and fix the other latent variables. The results for InfoGAN and SS-AVL are shown in Fig. 6e and 6f, respectively. We find that both SS-AVL and InfoGAN are able to generate the right image classes for a given discrete code. However, while SS-AVL is usually able to yield various styles of shapes and textures for the clothing items shown in images, InfoGAN tends to output images displaying a rather fixed style. We also change two continuous latent variables c_1, c_2 within $[-1, 1]$ and the results are shown in Fig. 8. We observe from Figs. 8b and 8d that the continuous latent variables in SS-AVL can capture well the variation in items' shape as well as in the lighting. Meanwhile, InfoGAN produces rather unexpected image variations, as it can be seen from Figs. 8a and 8c.

In Table 1 we provide the classification results when considering unsupervised learning on both MNIST and Fashion datasets, where we infer the class from the discrete variables **d**, and compare with the real class label. Most existing unsupervised learning methods adopt mixture models, which have higher complexity and significantly more parameters, so we refer to K as the number of components in such mixing models from Table 1. From these results we observe that the proposed approach achieves higher accuracy than InfoGAN and than most other models.

From Table 1 of the paper, we find that the GAN-based methods such as InfoGAN and AAE outperform other types of methods for the unsupervised classification task. Such results demonstrate that by combining a GAN model with the inference model we can capture discretely-defined variations in images and a GAN-based model performs better than the VAE-based methods. Compared with the baselines, the proposed SS-AVL achieves the best performance. In addition, the proposed SS-AVL does not require many components and still outperforms mixture models such as GMVAE and SubGAN.

8.2. Learning representations from a trained GAN model

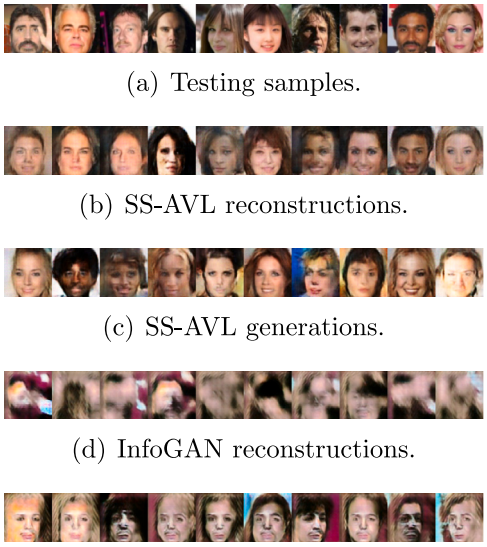
In this section, we investigate how the proposed approach can provide inference mechanisms for a trained GAN model. In certain applications, because of the data privacy requirements or due to memory limitations, we cannot access real samples and then we would consider the data generated by a GAN which can provide an appropriate solution. In the following experiments, we consider a GAN model which was trained on a certain dataset, considering a single latent vector **z**. So we can rewrite the data-free log-likelihood objective function from Eq. (16) by considering β-VAE [9]:

$$\log p_{\psi^*}(\mathbf{x}^*) \geq \mathbb{E}_{q_\xi(\mathbf{z}|\mathbf{x}^*)}[\log p_{\psi^*}(\mathbf{x}^* \mid \mathbf{z})] - \beta D_{KL}[q_\xi(\mathbf{z} \mid \mathbf{x}^*) \,||\, p(\mathbf{z})] \tag{27}$$

(a) Testing samples. (b) SS-AVL Reconstructions. (c) VAE Reconstructions.

Fig. 10. Reconstructions on CelebA data, after training.

Fig. 11. Interpolation results on CelebA testing samples.



(a) Testing samples.

(b) SS-AVL reconstructions.

(c) SS-AVL generations.

(d) InfoGAN reconstructions.

(e) InfoGAN generations.

Fig. 12. Generation and reconstructions results by SS-AVL and InfoGAN on Celeba dataset.

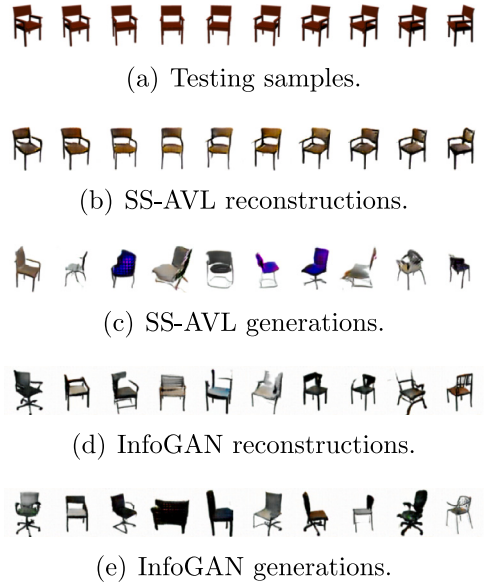
where we train a single inference model $q_\xi(\mathbf{z} \mid \mathbf{x}^*)$ by using the above objective function while $\mathbf{x}^*$ is produced by a GAN generator. We provide the reconstruction results in Fig. 9b for the images from Fig. 9a. These results are produced by the GAN's generator (fixed during the inference training) taking inputs from the latent variable $\mathbf{z}$ estimated by the inference model. This result shows that the proposed approach can provide an accurate inference model with a fixed GAN model without considering any real data. We also plot the log-likelihood in Fig. 9c, where we use the negative reconstruction error as the first term calculated for all testing samples, considering a fixed GAN model and a learned inference model, where we also compare with a VAE trained on all training samples. From these results, we observe that the proposed approach can allow the inference model to adapt well to the fixed GAN model by only considering a few training iterations.

In the following we train WGAN [26] on CelebA [46] dataset, which also receives a single latent vector $\mathbf{z}$. After training, we fix the WGAN model and train the inference model with the proposed data-free likelihood objective function from Eq. (27). In this case we optimize a reduced ELBO, where β is set as a small value, weighting the Kullback–Leibler divergence, aiming to produce higher-quality reconstructions. We also train a VAE model on CelebA for comparison. The generation and reconstructions are shown in Fig. 10. We observe that the proposed approach can produce better reconstruction results than VAEs.

We also perform image interpolation experiments and the results are presented in Fig. 11. From these results, we can observe a smooth transition between two faces during interpolation. These results demonstrate manifold continuity showing that the proposed approach is able to learn an accurate inference model from a fixed GAN model without requiring any real data.

8.3. Results on databases containing complex images

We evaluate the proposed SS-AVL approach on databases containing more complex images, such as CelebA [46] and 3D Chair [47]. We consider a 10-dimensional vector only for the continuous latent variables for modelling the underlying changing factors using SS-AVL. When considering the face images from Fig. 12a, the reconstructions and generations by SS-AVL when inferring the continuous and discrete latent variables from the testing samples and then combining the



(a) Testing samples.

(b) SS-AVL reconstructions.

(c) SS-AVL generations.

(d) InfoGAN reconstructions.

(e) InfoGAN generations.

Fig. 13. Generation and reconstruction results by SS-AVL and InfoGAN on 3D-Chair dataset.

Fig. 14. Manipulating latent codes on CelebA dataset. We change a single latent variable in the latent space from −1 to 1 while fixing the others.

resulting variables as inputs for the generator are provided in Figs. 12b and 12c. Meanwhile, reconstructions and generations by InfoGAN are shown in Figs. 12d and 12e. From Figs. 12b and 12d it can be observed that SS-AVL provides reasonable reconstructions, while InfoGAN yields blurred reconstruction results, according to the results from Figs. 12c and 12e.

In the following we consider a 10-dimensional vector for the continuous latent variables in order to model images from the 3D Chair dataset [47]. We set the dimension of the noise generation vector $\mathbf{z}$ as 100. The reconstruction and generation results for the images from Fig. 13a by SS-AVL are provided in Figs. 13b and 13c, while the reconstruction and generation results by InfoGAN [31] are provided in Figs. 13d and 13e. We can see that SS-AVL is able to provide accurate image reconstructions and realistic image generations of face and 3D chair from Figs. 12 and 13.

We also manipulate in turns a single latent variable from the given latent space, while fixing the others, when considering CelebA dataset, and the results are provided in Fig. 14. We can observe that the proposed approach can represent eight different kinds of disentangled representations modelling the following changes: modifying hair bangs, adding or not having glasses, modifying hair colour, modifying hair style, varying makeup, narrowing the face, changing facial expression such as smiling, or changing gender.

We repeat the same experiment as above for 3D Chair dataset, by changing a single variable while keeping the others fixed. The generation results for SS-AVL are shown in Figs. 15a–e while those by β−TCVAE [48] are displayed in Figs. 15f–j. From these results it can be observed that the proposed approach can discover a variety of feature variations in 3D Chair images representing chair orientation, style of chairs' backrest, leg style. It can be observed from these results that when the image is generated using SS-AVL and the chair size is increasing, other features, such as for example the backrest of the chair, are changed proportionally as well. Meanwhile, such changes are not properly synchronized in the images generated by β−TCVAE from Figs. 15f–j.

We also compare the proposed approach with other baselines on the 3D-Face dataset [49]. In Fig. 16a we provide the testing images and their reconstructions by the proposed SS-AVL are shown in Fig. 16b. In Figs. 16c [36] and 16d we show reconstructions of the 3D-Face dataset images by JointVAE [36] and β-VAE [9]. We consider 200 training epochs for each method. These visual results show that the proposed SS-AVL achieves slightly better reconstructions than other baselines on the challenging 3D-Face dataset.

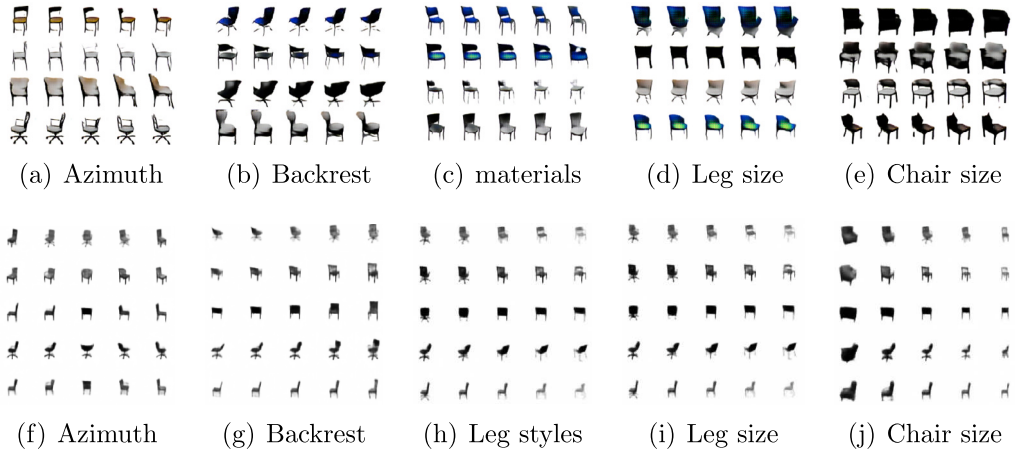


Fig. 15. Manipulating latent codes on the 3D chair dataset. We change a single latent variable in the latent space from −1 to 1 while fixing the others. The first row results are provided by the proposed SS-AVL approach. The second-row results are for β−TCVAE [48].

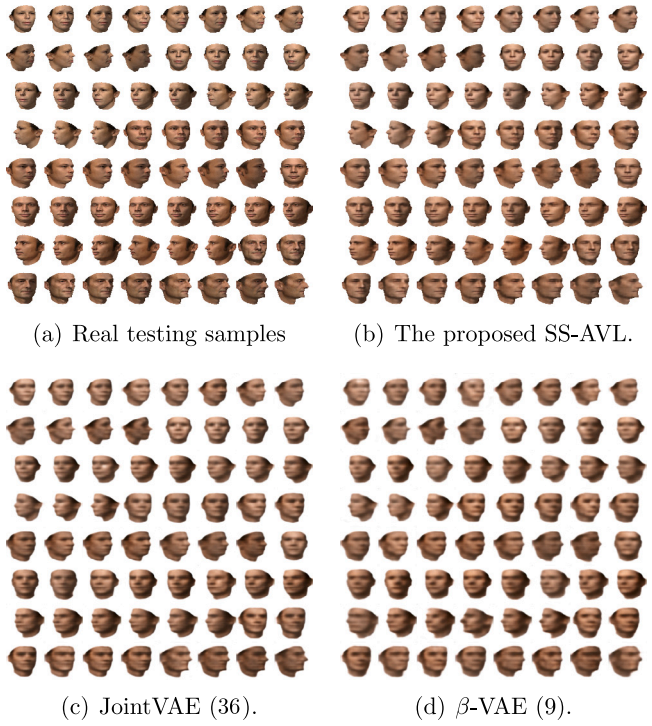


Fig. 16. The reconstructions results on the 3D-Face dataset [49].

Table 2
Disentanglement evaluation on the dSprites, where L is the dimension of the latent space.

Methods	L	Score
SS-AVL	10	0.79
Beta-VAE [9]	10	0.73
FactorVAE [10]	10	0.82
JointVAE [36]	10	0.69

8.4. Numerical evaluation

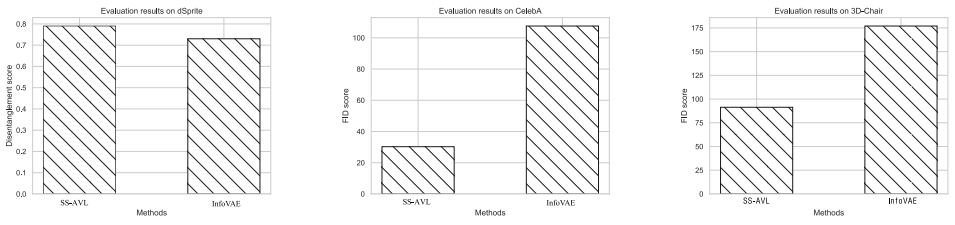
In this section, we investigate the disentanglement ability of the proposed approach by using the metric from [10] on the benchmark dataset dSprites [50]. We consider 6 continuous latent variables and one three-dimensional discrete vector when training the model. We perform the experiments following the JointVAE analysis [36]. The results are reported in Table 2 and all other results than ours are referred from [36]. The proposed SS-AVL approach achieves a competitive disentanglement score when compared with those of the state of art. We provide visual results of the dSprites in Fig. 19a and we compare with β-VAE and FactorVAE, shown in Fig. 19b and 19c, respectively. These results indicate that the proposed SS-AVL achieves competitive results when compared to the two VAE-based methods. In particular, SS-AVL produces variations for only a single image characteristic. For example, SS-AVL only changes the y-axis of an object in the image, while the other methods would also change the object's shape, as can be seen in the first row of the images shown in Fig. 19.

In the following we compare the proposed SS-AVL approach with InfoVAE [51]. InfoVAE also enforces learning good representations by maximizing the mutual information between representations and data. However, InfoVAE incorporates the mutual information term in the ELBO as a single optimization objective function, which is different from the proposed approach that maximizes the mutual information separately from the optimization of generator and inference models. We use the Maximum-Mean Discrepancy as the divergence $D_{MMD}(p \parallel q)$ in InfoVAE. We provide the disentanglement results, calculated as in [36], on dSprites in Fig. 17a while evaluating the Fréchet Inception Distance (FID) [52] on CelebA and 3D-Chair databases in Fig. 17b and Fig. 17c, respectively. According to these results we can observe that the proposed approach not only that it achieves a better disentanglement score but also it can generate better images than InfoVAE [51].

We also compare SS-AVL with more recent works on the challenging dataset ImageNet [53]. The Root Mean Square Error (RMSE) and Inception Score (IS) results are reported in Table 3. From these results we can observe that the proposed SS-AVL outperforms other recent works except for MVAE, which is a mixture model using many more parameters. These results demonstrate that the proposed SS-AVL achieves good results while also enjoying some good properties such as being able to learn interpretable representations and can be used for representation learning from a pre-trained GAN model.

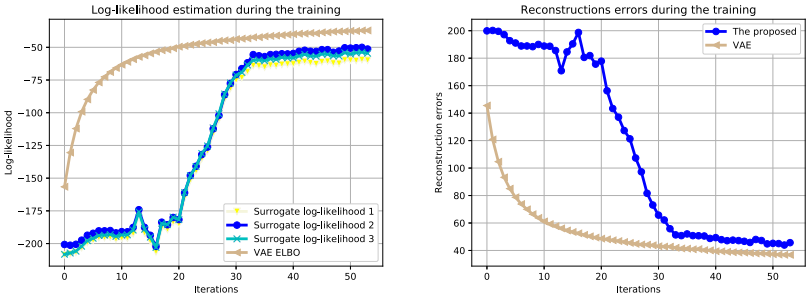
8.5. Analysis results

In the following we evaluate the model's likelihood during the training in order to investigate how the inference model can adapt to the generator. We also provide numerical results for the theoretical framework described in Section 4. We randomly select 1,000 testing samples $\mathbf{x}_t$, from the MNIST dataset in order to estimate the surrogate log-likelihoods $\mathcal{L}(\psi, \xi, \omega; \mathbf{x}_t)$, $\mathcal{L}(\psi, \omega; \mathbf{x}_t)$, $\mathcal{L}(\psi, \xi; \mathbf{x}_t)$, respectively. The results are shown in Fig. 18a, where the log-likelihood surrogates 1, 2, 3 represent $\mathcal{L}(\psi, \xi, \omega; \mathbf{x}_t)$ from Eq. (18), $\mathcal{L}(\psi, \omega; \mathbf{x}_t)$ from Eq. (13) and $\mathcal{L}(\psi, \xi; \mathbf{x}_t)$ from Eq. (16). For comparison we train a VAE, that shares



(a) Disentanglement score. (b) FID on CelebA. (c) FID on 3D-Chairs.

Fig. 17. Disentanglement score and Fréchet Inception Distance (FID) evaluations.



(a) Log-likelihood. (b) MSE.

Fig. 18. Log-likelihood estimation and reconstruction errors on MNIST testing samples during the training.

Table 3
RMSE and Inception score (*IS*) on ImageNet database. The results of other baselines are taken from [23] except ControlVAE [54] and NCP [55].

Model	RMSE	*IS*
MVAE-Gau [23]	**19.44**	**6.84**
MVAE-Gau fixed K [23]	20.87	6.30
MVAE-GS [23]	20.45	6.52
MSVI [56]	22.29	6.12
InfoVAE [51]	22.73	6.14
β-VAE [9]	31.47	5.05
VAE	28.44	5.46
Wasserstein–Wasserstein Auto-Encoders [57]	25.63	5.79
MAE [58]	23.25	5.87
ControlVAE [54]	20.78	6.18
NCP [55]	21.23	6.10
SS-AVL	20.15	6.29

the same network architecture with the proposed model and treat the VAE's results as the ELBO on the real sample log-likelihood. We can observe that $\mathcal{L}(\psi, \xi, \omega; \mathbf{x}_t)$, $\mathcal{L}(\psi, \omega; \mathbf{x}_t)$ and $\mathcal{L}(\psi, \xi; \mathbf{x}_t)$ are bounded on the real sample log-likelihood, as described in Section 4, "Theoretical framework". We also provide the reconstruction error calculated on 1,000 testing images for each iteration in Fig. 18b, which is gradually reduced when the number of iterations is increased. We observe that the quality of the generator, the surrogate log-likelihood on testing data samples increases towards the results provided by the VAE.

The log-likelihood and reconstruction results across all MNIST and Fashion testing samples are provided in Table 4, where SS-AVL-R denotes the model in which all inference networks are retrained by using the log-likelihood objective function. From these results, we observe that retraining inference models can improve the log-likelihood and reconstruction ability. These results show that the performance of the proposed approach is very close to that of a VAE trained on the real training samples. However, the inference model in SS-AVL does not actually see any real data samples.

Table 4
Log-likelihood (LL) estimation and MSE reconstructions on MNIST and Fashion datasets.

Dataset	Methods	LL	MSE
MNIST	SS-AVL	−59.01	35.39
MNIST	SS-AVL-R	−53.14	34.04
Fashion	SS-AVL	−46.39	29.41
Fashion	SS-AVL-R	−41.20	27.91

8.6. Ablation study

In this section, we investigate the importance of each module making up the SS-AVL model and how this is addressing important questions.

Is only one inference model enough? First, we evaluate the performance of the proposed approach when using only one inference model. Therefore, we create two different baselines, one without using the inference model $q_{\xi}(\mathbf{z} \mid \mathbf{x})$, namely SS-AVL1, while a second baseline does not use the inference model $q_{\omega}(\mathbf{d}, \mathbf{c} \mid \mathbf{x})$, namely SS-AVL2. We consider the same hyperparameter setting for the proposed approach and for the two baselines during the training. The reconstruction results for MNIST data from Fig. 20a, by SS-AVL1 and SS-AVL2 are shown in Figs. 20b and 20c, respectively. Meanwhile, the results by the proposed SS-AVL model are provided in Fig. 20d. We observe that the two baselines are generating digits with different styles when comparing to the real samples. However, by using a model with two inference models defining two log-likelihood objective functions, as proposed in this research study, we can generate digits displaying handwriting styles which are clearer and consistent with those from the input images, when comparing to the results provided by the baselines. The reason for this is that the proposed two log-likelihood functions are encouraged to capture different details of the data while when having only one inference model we can only capture a limited set of feature variations.

Assessment of the disentanglement between z and $\{\mathbf{c}, \mathbf{d}\}$. In this section, we investigate the disentanglement ability between $\mathbf{z}$ and $\{\mathbf{c}, \mathbf{d}\}$ in the proposed methodology. In Section 3.2, we have shown that the

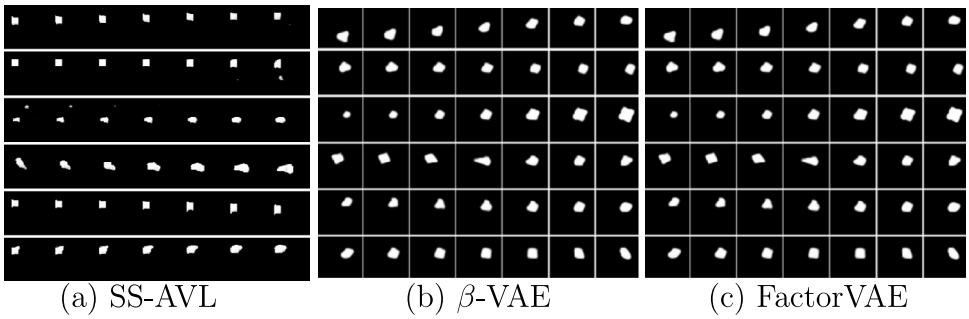


Fig. 19. Generation results for the dSprites dataset by changing a single latent variable from −3.0 to 3.0. The results from β-VAE and FactorVAE are from [36].

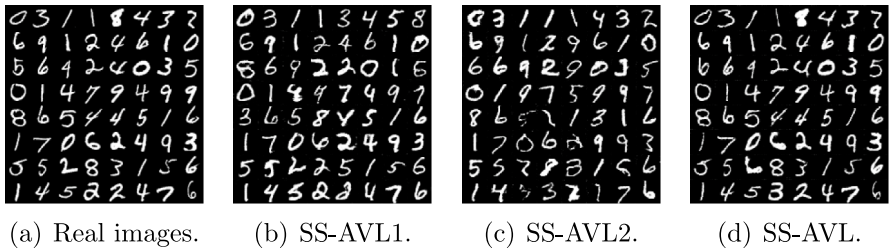


Fig. 20. Reconstruction results on MNIST testing samples by changing the inference encoders structure.

Fig. 21. Manipulated results when changing one of the dimensions of $\mathbf{z}$ from −3 to 3, while fixing the others.

latent variables $\{\mathbf{c}, \mathbf{d}\}$ are able to capture both discrete and continuous variations of data, which is evident from Fig. 4 for MNIST database. In the following, we change one dimension of $\mathbf{z}$ while fixing the other variables. From the results presented in Fig. 21, we can see that after changing a dimension from the continuous latent space vector $\mathbf{z} = (z1\ \ z2\ \ z3\ \ z4\ \ z5)^T$, we can change neither the digital type nor the handwriting style. The reason for this is that the proposed approach separately trains two inference models by using the proposed data-free likelihood objective functions. This mechanism helps enforce the disentanglement between the random variable $\mathbf{z}$ and the continuous and categorical variables $\{\mathbf{c}, \mathbf{d}\}$. At the same time, the mutual information maximization is only optimized on one inference model, which helps learning better interpretable representations than InfoGAN and other methods.

9. Limitation and conclusion

In this paper, we introduce a new approach, Self-Supervised Adversarial Variational Learning (SS-AVL), for jointly learning discrete and continuous interpretable representations. Different from other existing hybrid models, SS-AVL separately optimizes the inference and the generator models by using a two-stage optimization approach. SS-AVL is able to learn data representations from a trained (fixed) GAN model without using any real data and has many advantages over other VAE-GAN hybrid methods. This shows that the proposed approach represents a tool which can provide inference mechanisms for arbitrarily chosen GAN models without using any real data. In addition, SS-AVL is shown to be enabled with disentangled and interpretable representation learning mechanisms. Finally, experiments show that SS-AVL can provide high-quality diverse interpretable results of data variations. A challenge with the proposed approach is that the inference models cannot learn interpretable representations of the real data if the generator does not approximate the empirical data distribution well. This weakness can be addressed by configuring the hyper-parameters appropriately.

Data availability

Data used is public.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proc. Advances in Neural Inf. Proc. Systems, NIPS, 2014, pp. 2672–2680.
[2] Y. Sun, J. Chen, Q. Liu, G. Liu, Learning image compressed sensing with sub-pixel convolutional generative adversarial network, Pattern Recognit. 98 (2020).
[3] X. Jin, Z. Chen, W. Li, AI-GAN: Asynchronous interactive generative adversarial network for single image rain removal, Pattern Recognit. 100 (2020).
[4] X. Zhang, X. Wang, C. Shi, Z. Yan, X. Li, B. Kong, S. Lyu, B. Zhu, J. Lv, Y. Yin, et al., DE-GAN: Domain embedded gan for high quality face image inpainting, Pattern Recognit. 124 (2022).
[5] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, 2013, arXiv preprint arXiv:1312.6114.
[6] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[7] K. Ridgeway, A survey of inductive biases for factorial representation-learning, 2016, arXiv preprint arXiv:1612.05299.
[8] A. Alemi, I. Fischer, J. Dillon, K. Murphy, Deep variational information bottleneck, in: Proc. Int. Conf. of Learning Representation, ICLR, 2017, arXiv preprint arXiv:1612.00410.
[9] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, β-VAE: Learning basic visual concepts with a constrained variational framework, in: Proc. Int. Conf. on Learning Representations, ICLR, 2017.
[10] H. Kim, A. Mnih, Disentangling by factorising, in: Proc. Int. Conf. on Machine Learning (ICML), Vol. PMLR 80, 2018, pp. 2649–2658.
[11] S. Watanabe, Information theoretical analysis of multivariate correlation, IBM J. Res. Dev. 4 (1) (1960) 66–82.
[12] Z. Lin, K.K. Thekumparampil, G. Fanti, S. Oh, InfoGAN-CR: Disentangling generative adversarial networks with contrastive regularizers, in: Proc. of Int. Conf. on Machine Learning (ICML), Vol. PMLR 119, 2020, pp. 6127–6139.

[13] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders, in: Proc. ICLR-Workshop, 2016, ArXiv Preprint arXiv:1511.05644.
[14] A. Srivastava, L. Valkov, C. Russell, M. Gutmann, C. Sutton, VEEGAN: Reducing mode collapse in GANs using implicit variational learning, in: Proc. Advances in Neural Inf. Proc. Systems, NIPS, 2017, pp. 3308–3318.
[15] J. Donahue, P. Krähenbühl, T. Darrell, Adversarial feature learning, in: Proc. Int. Conf. on Learning Rep., ICLR, 2017, arXiv preprint arXiv:1605.09782.
[16] H. Huang, R. He, Z. Sun, T. Tan, Introvae: Introspective variational autoencoders for photographic image synthesis, in: Proc. Advances in Neural Inf. Proc. Systems, NIPS, 2018, pp. 52–63.
[17] D. Ulyanov, A. Vedaldi, V. Lempitsky, It takes (only) two: Adversarial generator-encoder networks, in: Proc. AAAI Conf. on Artificial Intelligence, 2018, pp. 1250–1257.
[18] L. Mescheder, S. Nowozin, A. Geiger, Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks, in: Proc. Int. Conf. on Machine Learning, Vol. PMLR 70, ICML, 2017, pp. 2391–2400.
[19] Y. Pu, W. Wang, R. Henao, C. L., Z. Gan, C. Li, L. Carin, Adversarial symmetric variational autoencoder, in: Proc. Advances in Neural Inf. Proc. Systems, NIPS, 2017, pp. 4333–4342.
[20] A.B.L. Larsen, S.K. Sønderby, H. Larochelle, O. Winther, Autoencoding beyond pixels using a learned similarity metric, in: Proc. Int. Conf. on Machine Learning (ICML), Vol. PMLR 48, 2016, pp. 1558–1566.
[21] F. Ye, A.G. Bors, Learning joint latent representations based on information maximization, Inform. Sci. 567 (2021) 216–236.
[22] T. Che, Y. Li, R. Zhang, D. Hjelm, W. Li, Y. Song, Y. Bengio, Maximum-likelihood augmented discrete generative adversarial networks, 2017, arXiv preprint arXiv:1702.07983.
[23] F. Ye, A.G. Bors, Deep mixture generative autoencoders, IEEE Trans. Neural Netw. Learn. Syst. 33 (10) (2022) 5789–5803.
[24] A. Kolesnikov, X. Zhai, L. Beyer, Revisiting self-supervised visual representation learning, in: Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 1920–1929.
[25] X. Zhai, A. Oliver, A. Kolesnikov, L. Beyer, S4L: Self-supervised semi-supervised learning, in: Proc. of the IEEE/CVF Int. Conf. on Computer Vision, 2019, pp. 1476–1485.
[26] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: Proc. Int. Conf. on Machine Learning (ICML), Vol. PMLR 70, 2017, pp. 214–223.
[27] C. Villani, Optimal Transport: Old and New, Springer, 2008.
[28] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of Wasserstein GANs, in: Advances in Neural Inf. Proc. Systems (NIPS), 2017, pp. 5767–5777.
[29] D.J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in: Proc. Int. Conf. on Machine Learning (ICML), Vol. PMLR 32, 2014, pp. II–1278–II–1286.
[30] E. Jang, S. Gu, B. Poole, Categorical reparameterization with Gumbel-Softmax, in: Proc. Int. Conf. on Learning Representations, ICLR, 2016, arXiv preprint arXiv:1611.01144.
[31] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets, in: Advances in Neural Inf. Proc. Systems (NIPS), 2016, pp. 2172–2180.
[32] F. Barber, D. Agakov, The IM algorithm: a variational approach to information maximization, Adv. Neural Inf. Process. Syst. (2003) 201–208.
[33] J. Bridle, A. Heading, D. MacKay, Unsupervised classifiers, mutual information and'phantom targets, in: Proc. Advances in Neural Inf. Proc. Systems, NIPS, 1992, pp. 1096–1101.
[34] A. Krause, P. Perona, R. Gomes, Discriminative clustering by regularized information maximization, in: Proc. Advances in Neural Inf. Proc. Systems, NIPS, 2010, pp. 775–783.
[35] F. Ye, A.G. Bors, Lifelong learning of interpretable image representations, in: Proc. Int. Conf. on Image Processing Theory, Tools and Applications, IPTA, 2020, pp. 1–6.
[36] E. Dupont, Learning disentangled joint continuous and discrete representations, in: Advances in Neural Inf. Proc. Systems (NIPS), 2018, pp. 710–720.
[37] C. Zhuang, S. Huang, G. Cheng, J. Ning, Multi-criteria selection of rehearsal samples for continual learning, Pattern Recognit. 132 (2022) 108907.
[38] I. Skorokhodov, S. Tulyakov, M. Elhoseiny, Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3626–3636.
[39] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recog, Proc. IEEE 86 (11) (1998) 2278–2324.
[40] J. Liang, J. Yang, H.-Y. Lee, K. Wang, M.-H. Yang, Sub-GAN: An unsupervised generative model via subspaces, in: Proc. of the European Conf. on Computer Vision (ECCV), Vol. LNCS 11215, 2018, pp. 698–714.
[41] N. Dilokthanakul, P. Mediano, M. Garnelo, M. Lee, H. Salimbeni, K. Arulkumaran, M. Shanahan, Deep unsupervised clustering with Gaussian mixture variational autoencoders, 2016, arXiv preprint arXiv:1611.02648.
[42] J.T. Springenberg, Unsupervised and semi-supervised learning with categorical generative adversarial networks, in: Proc. Int. Conf. on Learning Representations, ICLR, 2016, arXiv preprint arXiv:1511.06390.
[43] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: Proc. Int. Conf. on Machine Learning (ICML), Vol. PMLR 48, 2016, pp. 478–487.
[44] A. Makhzani, B. Frey, PixelGAN autoencoders, in: Advances in Neural Information Processing Systems (NIPS), 2017, pp. 1972–1982.
[45] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, A. Courville, Adversarially learned inference, in: Proc. Int. Conf. on Learning Rep., ICLR, 2017, arXiv preprint arXiv:1606.00704.
[46] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proc. of IEEE Int. Conf. on Computer Vision, ICCV, 2015, pp. 3730–3738.
[47] M. Aubry, D. Maturana, A.A. Efros, B.C. Russell, J. Sivic, Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, CVPR, 2014, pp. 3762–3769.
[48] R. Chen, X. Li, R. Grosse, D. Duvenaud, Isolating sources of disentanglement in VAEs, in: Proc. Advances in Neural Inf. Proc. Systems (NIPS), 2018, pp. 2615–2625.
[49] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, T. Vetter, A 3D face model for pose and illumination invariant face recognition, in: IEEE Int. Conf. on Advanced Video and Signal Based Surveillance, AVSS, 2009, pp. 296–301.
[50] L. Matthey, I. Higgins, D. Hassabis, A. Lerchner, dSprites: Disentanglement testing sprites dataset, 2017, https://github.com/deepmind/dsprites-dataset/.
[51] S. Zhao, J. Song, S. Ermon, InfoVAE: Balancing learning and inference in variational autoencoders, in: Proc. of the AAAI Conf. on Artificial Intelligence, Vol. 33, 2019, pp. 5885–5892.
[52] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local Nash equilibrium, in: Advances in Neural Inf. Proc. Syst. (NIPS), 2017, pp. 6626–6637.
[53] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proc. Advances in Neural Inf. Proc. Systems, NIPS, 2012, pp. 1097–1105.
[54] H. Shao, Y. Yang, H. Lin, L. Lin, Y. Chen, Q. Yang, H. Zhao, Rethinking controllable variational autoencoders, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 19250–19259.
[55] J. Aneja, A. Schwing, J. Kautz, A. Vahdat, A contrastive learning approach for training variational autoencoder priors, Adv. Neural Inf. Process. Syst. 34 (2021) 480–493.
[56] R. Kurle, S. Günnemann, P. van der Smagt, Multi-source neural variational inference, in: Proc. of AAAI Conf. on Artificial Intelligence, Vol. 33, 2019, pp. 4114–4121.
[57] S. Zhang, Y. Gao, Y. Jiao, J. Liu, Y. Wang, C. Yang, Wasserstein-Wasserstein auto-encoders, 2019, arXiv preprint arXiv:1902.09323.
[58] X. Ma, C. Zhou, E. Hovy, MAE: Mutual posterior-divergence regularization for variational AutoEncoders, in: Proc. Int. Conf. on Learning Representations, ICLR, 2019, arXiv preprint arXiv:1901.01498.

Fei Ye is currently a Ph.D. candidate in the Department of Computer Science at the University of York, UK. He received the bachelor degree from Chengdu University of Technology, China, in 2014 and the master degree in computer science and technology from Southwest Jiaotong University, China, in 2018. His research topics includes deep generative image models, lifelong learning and mixture models.

Adrian G. Bors received the Ph.D. degree in informatics from the Aristotle University of Thessaloniki, Greece, in 1999 and the M.Sc. degree in electronics engineering from the Polytechnic University of Bucharest, Romania, in 1992. In 1999 he joined the Department of Computer Science, Univ. of York, UK, where he is currently an Associate Professor. He was a Research Scientist at Tampere Univ. of Technology, Finland, a Visiting Scholar at the Univ. of California at San Diego (UCSD), and an Invited Professor at the Univ. of Montpellier, France. Dr. Bors has authored and co-authored more than 160 research papers, including 41 in journals. His research interests include computer vision, computational intelligence, pattern recognition, and image processing. He was a member of the organizing committees for IEEE WIFS 2021, IPTA 2020, IEEE ICIP 2018, BMVC 2016, IPTA 2014, CAIP 2013, and IEEE ICIP 2001. He was an Associate Editor of the IEEE Transactions on Image Processing from 2010 to 2014 and the IEEE Transactions on Neural Networks from 2001 to 2009. He was a Co-Guest Editor for a special issue for the International Journal for Computer Vision in 2018 as well as for the Journal of Pattern Recognition in 2015. He is a Senior IEEE member.