Double InfoGAN for Contrastive Analysis

Florence Carton\textsuperscript{1} Robin Louiset\textsuperscript{1,2} Pietro Gori\textsuperscript{1}
\textsuperscript{1}LTCI, Télecom Paris, IPParis, France \textsuperscript{2}NeuroSpin, CEA, Universite Paris-Saclay, France

Abstract

Contrastive Analysis (CA) deals with the discovery of what is common and what is distinctive of a target domain compared to a background one. This is of great interest in many applications, such as medical imaging. Current state-of-the-art (SOTA) methods are latent variable models based on VAE (CA-VAEs). However, they all either ignore important constraints or they don’t enforce fundamental assumptions. This may lead to suboptimal solutions where distinctive factors are mistaken for common ones (or viceversa). Furthermore, the generated images have a rather poor quality, typical of VAEs, decreasing their interpretability and usefulness. Here, we propose Double InfoGAN, the first GAN based method for CA that leverages the high-quality synthesis of GAN and the separation power of InfoGAN. Experimental results on four visual datasets, from simple synthetic examples to complex medical images, show that the proposed method outperforms SOTA CA-VAEs in terms of latent separation and image quality. Datasets and code are available online\textsuperscript{1}.

1 INTRODUCTION

Learning disentangled generative factors in an unsupervised way has gathered much attention lately since it is of interest in many domains, such as medical imaging. Most approaches look for factors that capture distinct, noticeable and semantically meaningful variations in one dataset (e.g., presence of hat or glasses in CelebA). Authors usually propose well adapted regularizations, which may promote, for instance, “uncorrelatedness” (FactorVAE [Kim and Mnih, 2018]) or “informativeness” (InfoGAN [Chen et al., 2016]).

In this paper, we focus on a related but different problem, that has been named Contrastive Analysis (CA) [Zou et al., 2013, Abid et al., 2018, Weinberger et al., 2022]. We wish to discover in an unsupervised way what is added or modified on a target dataset compared to a control (or background) dataset, as well as what is common between the two domains. For example, in medical imaging, one would like to discover the salient variations characterizing a pathology that are only present in a population of patients and not in a population of healthy controls. Both the target (patients) and the background (healthy) datasets are supposed to share uninteresting (healthy) variations. The goal is thus to identify and separate the generative factors common to both populations from the ones distinctive (i.e., specific) only of the target dataset.

The most recent CA methods are based on the Variational AutoEncoders (VAE) [Kingma and Welling, 2014] model and they are called Contrastive VAE (CA-VAE). These methods assume that samples from the target dataset are generated using two sets of latent factors, common $z$ and salient $s$, whereas samples from the control dataset are generated using only the common $z$ factors. The salient factors $s$ should therefore model the specific patterns of variations of the target dataset. All these methods share the same general mathematical formulation, which derives from the standard VAE. However, they all either ignore a term of the proposed loss (e.g., KL loss in [Abid and Zou, 2019, Ruiz et al., 2019]) or they don’t enforce important assumptions (e.g., independence between $z$ and $s$ in [Weinberger et al., 2022]), which may lead to sub-optimal solutions where salient factors are mistaken for common ones (or viceversa). Furthermore, they all share a typical downside of VAEs: a blurry and poor quality image generation.

For these reasons, we propose Double InfoGAN: a novel Contrastive method which leverages the high-quality synthesis of Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] and the separation power of InfoGAN [Chen et al., 2016]. To the best of our knowledge, this is the first GAN...
Double InfoGAN for Contrastive Analysis

Figure 1: Two examples of datasets for Contrastive Analysis. **First figure:** Brats dataset [Menze et al., 2014]. Top: MRI images of healthy brains (control dataset). Bottom: MRI images of brains with tumor (target dataset). **Second figure:** CelebA dataset. Top: control dataset with regular faces (no smile, no glasses). Bottom: target dataset that contains smiling faces with glasses.

Based on method proposed in the context of Contrastive Analysis. The main contributions of this paper are:

- The first GAN based method for Contrastive Analysis (CA) which allows high-quality synthesis.
- A new regularization term for CA, inspired by InfoGAN.
- Two new losses for an accurate separation and estimate of the common and salient generative factors.
- Extensive experimental results on four visual datasets, from synthetic to complex ones, show that the proposed method outperforms SOTA CA-VAE methods in terms of latent separation and image quality. Datasets and code are available online.¹

2 RELATED WORK

Separating common from distinctive latent representations has become an active research area in several fields, such as domain adaptation (DA) [Ganin et al., 2017, Hoffman et al., 2018] and image-to-image translation (IMI) [Zhu et al., 2017, Isola et al., 2017, Liu et al., 2017, Lee et al., 2018, Huang et al., 2018].

DA seeks to transfer a classifier from a source domain, with many labelled samples, to a different target domain, which has few or no labelled data. As shown in [Ganin et al., 2017], an effective classifier should use shared features that cannot discriminate between the two domains. The goal of IMI is instead to estimate a transformation that maps images from the source domain to the target one by disentangling and controlling high-level visual attributes (style, gender, objects) [Lee et al., 2018]. The main difference between these methods and the proposed one is the objective. Our goal is to statistically analyze two domains (e.g., healthy and patients) looking for latent representations that generate the background (e.g., healthy) and target (e.g., pathological) content. We do not seek to transfer a classifier or to map an image to a different distribution. We wish, for instance, to generate new images and not only to translate them to another domain. Another important difference is that we do not want to encode only a particular distinctive attribute (e.g., style [Ma et al., 2019], gender) but all distinctive variations of the target domain with respect to the background one. Furthermore, we do not plan to use a weight sharing constraint [Lee et al., 2018, Liu et al., 2017], or other architectural constraints, which assume that the main differences are, for instance, only in the low-level features (color, texture, edges, etc.).

Our work is also close to unsupervised anomaly detection [Guillon et al., 2021, Baur et al., 2021, PANG et al., 2022, Vétil et al., 2022], which is usually composed of two steps. First, the distribution of the background (control) domain is learned, using deep generative models. Then, or at the same time, a discriminator is optimized to detect the target (anomalous) samples. By looking at the reconstruction errors [Guillon et al., 2021], attention scores [Venkataramanan et al., 2020], visual saliency [Kimura et al., 2020] or other features, one can understand which are the salient patterns of the target (anomalous) domain. Even if this strategy can be highly interpretable, the goal is to spot an anomalous sample and not to model the latent factors that generate the anomalous patterns.

Another class of methods, mainly used in the fields of data integration and data fusion, are the projection based latent variables approaches, such as 2B-PLS, 02PLS, DISCO-SCA, GSVD, JIVE [Feng et al., 2018, Rohlf and Corti, 2000, Deun et al., 2012, Yu et al., 2017, Trygg, 2002, Smilde et al., 2017]. Contrary to these methods, we do not use only linear transformations, but we leverage the capacity of deep learning to estimate non-linear mappings.

In parallel, research on disentanglement has been developed, making it possible to modify a single and semantically meaningful pattern of the image (e.g., person’s smile, gender), by varying only one component of the latent representation [Kim and Mnih, 2018]. As shown in [Locatello et al., 2019], the unsupervised learning of disentangled representations is theoretically impossible from i.i.d. samples without inductive biases [Higgins et al., 2017, Chen et al., 2016], weak labels [Shu et al., 2020, Locatello et al., 2020], or supervision [Lample et al., 2017, Choi et al., 2018, He et al., 2019, Shi et al., 2021, Joy et al., 2021]. These methods have

¹https://github.com/Florence-C/Double_InfoGAN.git
all focused on the latent generative factors of a single dataset, and their goal is thus different from ours. With a different perspective, methods stemming from the recent Contrastive Analysis (CA) setting [Zou et al., 2013, Abid et al., 2018, Tu et al., 2021, Ruiz et al., 2019, Zou et al., 2022, Abid and Zou, 2019, Choudhuri et al., 2019, Severson et al., 2019, Weinberger et al., 2022] mainly use variational autoencoders (VAE) to model latent variations only present among target samples and not in the background dataset. Similarly, in [Benaim et al., 2019], authors used standard autoencoders to estimate common latent patterns between two domains as well as patterns unique to each domain. Being based on auto-encoders, this method cannot sample in the latent space (i.e., no new image generation) and its goal is to map sample images from one domain to the other, as in IMI. Another related method is NestedVAE [Vowels et al., 2020], whose goal is bias reduction by estimating common factors between visual domains using paired data. Here, we wish to use unpaired datasets.

Lastly, CA is different from style vs. content separation and style transfer. In particular, in recent works [Kazemi et al., 2019, von Kugelgen et al., 2021], content usually refers to the invariant generative factors across samples and views (i.e., transformations/augmentations of a sample), while style refers to the varying factors. Content and style thus depend on the chosen semantic-invariant transformations, and they are defined for a single dataset. In CA, we do not necessarily need transformations or views, and we jointly analyze two different datasets.

3 BACKGROUND

InfoGAN In [Chen et al., 2016], differently from standard GAN [Goodfellow et al., 2014], authors propose a new method, called InfoGAN, where they decompose the input noise vector of GANs into two parts: 1) z, which is considered as a nuisance and incompressible noise and 2) e, which should model the salient semantic features of the data distribution. The generator of this new model, G(z, e), takes as input both z and e to generate samples x. As shown in [Chen et al., 2016], without regularisation, the generator G may ignore the additional code e or find a trivial (and useless) solution. To this end, authors propose to regularize the estimate of G by maximizing the mutual information I(c; x) between c and x ∼ G(z, e). Maximum I is obtained when c and x are completely dependent and one becomes completely redundant with the knowledge of the other. This should increase the informativeness of c, namely all salient semantic information should be in c and not in z, which should only account for additional randomness (i.e., noise). Authors propose to maximize a lower bound of I(c; x) by defining an auxiliary distribution Q(c|x), parameterized as a neural network, to approximate P(c|x):

$$I(c; x) \geq E_{z \sim P(z), c \sim P(c), x \sim P(x|c, z)} \log (Q(c|x)) + H(e)$$

(1)

More mathematical details in the Supplementary.

Contrastive VAE (CA-VAE) In this section, we present the CA-VAE models [Choudhuri et al., 2019, Severson et al., 2019, Abid and Zou, 2019, Ruiz et al., 2019, Zou et al., 2022, Weinberger et al., 2022]. Let $$X = \{x_i\}$$ and $$Y = \{y_j\}$$ be the background (or control) and target data-sets of images respectively. Both $$\{x_i\}$$ and $$\{y_j\}$$ are assumed to be i.i.d. from two different and unknown distributions ($$P(x)$$ and $$P(y)$$) that depend on a pair of latent variables (z, s). Here, s is assumed to capture the salient generative factors proper only to Y whereas z should describe the common generative factors between X and Y. The generative models (i.e. same decoder with parameters θ) are: $$x_i \sim P_θ(x; z_i, s_i = s')$$ and $$y_j \sim P_θ(y; z_j, z_j),$$ where the salient factors $$s_i$$ of X are fixed to a constant value s′ (e.g., s′ = 0), thus enforcing z to fully encode alone X. The conditional posterior distributions are approximated using another neural network (i.e. encoder with parameters φ) shared between X and Y, $$Q_φ(z; s, x)$$ and $$Q_φ(z; y, y),$$ which are usually assumed to be conditional independent: $$Q_φ(z; s) = Q_φ(z|s).$$ The latent generative factors (z, s) are also usually assumed to be independent (i.e., $$P(z,s) = P(z)P(s).$$) The common factor z should follow the same prior distribution in X and Y (e.g., $$P_z(z) = P(y) = N(0,0, I).$$) The salient factor s follows instead a different prior distribution between X and Y, such as $$P_p(s) = N(0,0, I)$$ and $$P_p(s) = δ(s = s'),$$ the Dirac distribution centered at s'. Based on this generative latent variable model, one can derive a lower bound of the marginal log likelihood:

$$\log P(x) \geq E_{Q_φ(x|z)}Q_φ(s|x) \log P_θ(x|z, s) - KL(Q_φ(z|x)||p_φ(z)) - KL(Q_φ(s|x)||p_φ(s))$$

(2)

and similarly for $$\log P(y).$$ All existing CA-VAE methods share this mathematical framework. They mainly differ for optimization or architectural choices and new added losses. However, none of these methods explicitly enforces the independence between common and salient latent factors2 and most of them ignore the KL divergence.

2[Abid and Zou, 2019] proposed to minimize the total correlation (TC) between $$q_{φ, θ, ϕ}(z, s|x)$$ and $$q_{φ, ϕ}(z|x)q_{φ, ϕ}(s|x)$$ via the density-ratio trick.
The second module $C$ is trained to produce images correctly classified by $y$ in the correct domain (i.e., $C$ is trained to produce images correctly classified by $y$ in the correct domain). We suppose that the back-end image reconstruction is similar to the one proposed in CA-VAE but within the framework of InfoGAN. We suppose, and force it in practice, that the latent variables $z = \{z_1, ..., z_L\}$ and $s = \{s_1, ..., s_M\}$ are independent and follow a factorized distribution: $P(z) = \prod_{i=1}^{L} P(z_i)$ and $P(s) = \prod_{j=1}^{M} P(s_j)$, for $X$ and $Y$. The total cost function is:

$$
\min_{G,Q,D} \max_{\theta_{G},\theta_{Q},\theta_{D}} \mathcal{L}_{Adv}(G,D) + w_{class} \mathcal{L}_{cl}(G,C) - \mathcal{L}_{If}(G,s,\mathcal{Q}) \\
+ w_{Im} \mathcal{L}_{Im}(G,Q,s) 
$$

In the following, we will describe each term.

**Adversarial GAN Loss** As in [Goodfellow et al., 2014], $G$ and $D$ are trained together in a min-max game using the original nonsaturating GAN (NSGAN) formulation:

$$
\mathcal{L}_{Adv}(D,G) = \mathbb{E}_{x \sim P(x)} [\log(D(x))] \\
- \mathbb{E}_{z \sim P(z)} [\log(1 - D(G(z, 0)))] + w_{r} [\mathbb{E}_{y \sim P(y)} \log(1 - D(G(z, y)))]
$$

where $D(I)$ indicates the probability that $I$ is real or fake and $x \sim P(x)$ and $y \sim P(y)$ are real images. Furthermore, we choose the same factorized prior distribution $P(z)$ for both $X$ and $Y$ (i.e., $P_z(z) = P_y(z) = P(z)$), namely a Gaussian $\mathcal{N}(0,1)$. We also tested a uniform distribution $U[-1,1]$ but the results were slightly worse. Instead, about $P(s)$, it should be different between $X$ and $Y$. We use a Dirac delta distribution centered at 0 for $X$ (i.e., $P_z(s) = \delta(s = 0)$) and we have tested several distributions for $P_y(s)$. Depending on the data and related assumptions, one could use, for instance, a factorized uniform distribution, $U[0,1]$, or a factorized Gaussian $\mathcal{N}(0,1)$ (ignoring the samples equal to 0). In our experiments, results were slightly better when using $\mathcal{N}(0,1)$.

**Class Loss** To make sure that generated images belong to the correct class, we propose to add a second discriminator module $C$. It is trained on real images to predict the correct class: $X$ or $Y$. At the same time, $G$ is trained to produce images correctly classified by $C$. We (arbitrarily) assign 0 (resp. 1) for class $X$ (resp. $Y$) and use the binary cross entropy ($B$). The loss is:

$$
\mathcal{L}_{cl}(C) = \mathbb{E}_{x \sim P(x)} [B(C(x, 0), 0)] \\
+ \mathbb{E}_{y \sim P(y)} [B(C(y, 1), 1)] \\
\mathcal{L}_{cl}(G) = \mathbb{E}_{z \sim P_z(z)} [B(C(G(z, 0)), 0)] \\
+ \mathbb{E}_{y \sim P_y(s)} [B(C(G(z, y), 1))] 
$$

**Info Loss** Similarly to InfoGAN, we propose two regularization terms based on mutual information, $I((z,s);y)$ and $I((z,s) = s');x)$, to encourage informative latent codes. However, in our case, these two
terms are not added to disentangle between informative and nuisance generative factors, but to enforce the separation between common and salient factors. Indeed, the maximization of these two regularity terms should enforce $z$ to fully encode $X$ and at the same time to be informative for the generation of $Y$. In parallel, $s$ should only encode distinctive semantic information of $Y$. Please note that the inclusion of two other nuisance factors, similarly to InfoGAN, describing the incompressible noise of $X$ and $Y$, would make the analysis more complex (i.e., additional regularity terms) since they should not model the common nor the salient generative factors.

Since $z$ and $s$ are independent by construction, the mutual information $I((z, s); :)$ can be decomposed into the sum of the two mutual information $I(z; :)$ + $I(s; :)$.

Thus, similarly to InfoGAN (see Eq. 1), we can retrieve four lower bounds. As in [Chen et al., 2016, Lin et al., 2020b], to promote stability and efficiency, we model the two auxiliary distributions, $Q_x$ and $Q_s$, as factorized distributions. Beside a factorized Gaussian distribution with identity covariance, we have also tested a factorized Laplace distribution $L(\mu, b)$ with $b = 1$. This brings to a $l1$ reconstruction loss instead of a standard $l2$, and showed better performance in practice.

Finally, to better train $Q_s$, and since we know that $s$ should be equal to 0 for real images of domain $X$ (i.e., $x_R \sim P(x_R))$, we also add as regularization the lower bound of the mutual information $I(s; x_R)$. As before, we fix $P_s(s) = \delta(s = 0)$. The sum of these five lower bounds defines the $L_{Inf_o}$ loss:

$$L_{Inf_o}(G, Q_z, Q_s) = w_{bg} \mathbb{E}_{x_R \sim P(x_R)}[w_{Inf_o}](Q_z(G(z, 0)) - z) + w_{Inf_o}Q_z(G(z, 0)) - 0$$

$$+ w_{Inf_o}Q_s(G(z, s) - s) + w_{Inf_o}Q_s(G(z, 0) - 0)$$

$$+ w_{Inf_o}Q_s(G(z, s) - s) + w_{Inf_o}Q_x(G(z, s) - 0)$$

$$+ w_{Inf_o}Q_x(G(z, 0) - 0)$$

(6)

**Image reconstruction loss**

Differently from usual GAN models, we also propose to maximize the log-likelihood $\log(P(y))$ (and $\log(P(x))$) of the generated images based on the proposed model. Indeed, no likelihood is generally available for optimizing the generator $G$ in a GAN model [Goodfellow et al., 2014]. However, here, given a real image $y_R$ (or $x_R$), we can use the auxiliary encoder $Q = (Q_z, Q_s)$ to estimate the latent factors $\hat{z}$ and $\hat{s}$ that should generate $y_R$ (or $x_R$) and then maximize (an approximation) of the log-likelihood of the generated images $y = G(\hat{z}, \hat{s})$ (or $x = G(\hat{z}, 0)$):

$$\log P(y) \geq \mathbb{E}_{y_R \sim P(y_R), (z, s) \sim Q(z, s)(y_R)} \log P(y|z, s, y_R) - \mathbb{E}_{y_R \sim P(y_R)} KL(Q(z, s|y_R) || P(z, s|y_R))$$

(7)

We notice that the second term should tend towards 0 during training thanks to the previous Info Loss.\(^3\) We can thus approximate $\log P(y)$ by computing only the left term and modeling $P(y|z, s, y_R)$ as a Laplace distribution $L(\mu, b)$ with $b = 1$. We use a Laplace distribution, instead of a Gaussian one, since it has been shown, for instance in [Isola et al., 2017], that a $l1$-loss encourages sharper and better image reconstructions than a $l2$-loss. Similar computations can be done for $\log P(x)$.

We define $L_{Inf}(G, Q_s, Q_s) = \log P(x) + \log P(y)$:

$$L_{Inf}(G, Q_z, Q_s) = w_{bg} \mathbb{E}_{x_R \sim P(x_R)}[\log G(\hat{z}, 0) - x_R]$$

$$+ w_{inf} \mathbb{E}_{y_R \sim P(y_R)}[\log G(\hat{z}, \hat{s}) - y_R]$$

(8)

5 RESULTS

In this section, we present the results of our model on four different visual datasets. Three of them (CelebA with accessories [Weinberger et al., 2022], Cifar-10-MNIST and dSprites-MNIST) have been conceived for the CA setting, giving us the possibility to qualitatively and quantitatively evaluate the performance of our model. We compare it with two SOTA Contrastive VAE algorithms (cVAE [Abid and Zou, 2019] and MM-cVAE [Weinberger et al., 2022]) that had the best results in [Weinberger et al., 2022].\(^4\) The fourth dataset, Brats [Menze et al., 2014], comprises T1-w MR brain images of healthy subject and patient with brain tumors, and is used for qualitative evaluation.

For quantitative evaluation, we use the fact that the information about attributes (e.g. glasses/hats in CelebA, MNIST digits, Cifar objects) should be present either in the common or in the salient space. Given a test set of images, we first use $Q$ to reconstruct $\hat{z}$ and $\hat{s}$ and then train a classifier on them to predict the attribute presence. By evaluating the discriminative power of the classifier, we can understand whether the information about the attributes has been put in the common or salient latent space by the method.

Qualitatively, the model can be evaluated by: 1) looking at the image reconstruction, 2) generating new images (sampling different salient features) and 3) swapping salient features. Given two real images $x_R \in X$ and $y_R \in Y$, we can first estimate the latent factors $(\hat{z}_X, \hat{s}_X)$ and $(\hat{z}_Y, \hat{s}_Y)$, that should have generated $x_R$ and $y_R$, using $Q$. Then, we can swap the estimated salient features $s_X$ and $s_Y$, and re-generate the images $G(\hat{z}_X, s_Y)$ and $G(\hat{z}_Y, s_X)$.

Implementation details about the architectures and hyper-parameters used in the different experiments can be found in the Supplementary.

\(^3\)Lower bounds become tight as $Q$ resembles the true $P$.

\(^4\)We use the code provided by the authors of MM-cVAE.
CelebA with accessories We use the dataset based on CelebA [Liu et al., 2015b] presented in [Weinberger et al., 2022], where background images $X$ are faces with neither hat of glasses, and target images $Y$ are faces with hat or glasses. We use 20,000 images for training, 10,000 background and 10,000 target, equally divided between glasses and hat. To evaluate the target class separation, we create a test set with images (5,000 with glasses and 5,000 with hat) never seen during training and compute the accuracy of a logistic regression (with 5-fold cross validation) on the reconstructed latent factors $\hat{s}_y = Q_s(s|y) \text{ and } \hat{z}_y = Q_z(z|y)$. Results are available in Table 1. Please note that the evaluation protocol in [Weinberger et al., 2022] was different since authors did not use an external test set. For a fair comparison, we run all methods 5 times (with different random seeds) for 500 epochs, and reported the highest (best), average and lowest (worst) scores. Extensive results are presented in the Supplementary. It is interesting to underline that MM-cVAE [Weinberger et al., 2022] does not converge at every run. We have observed a divergence of the KL loss in about 10% of the trainings, which led to a convergence failure. We have used the original architecture of the MM-cVAE paper [Weinberger et al., 2022] to reproduce their results. We provide qualitative results in Fig. 3 with image reconstruction and salient feature swap. Please note that this would not be possible with SOTA IMI methods, such as CycleGAN [Zhu et al., 2017] and MUNIT [Huang et al., 2018], not conceived for the CA setting. First of all, we observe that our model produces images of better quality than MM-cVAE, although this could probably be improved using larger GAN architectures, such as BigGAN [Brock et al., 2019] or StyleGAN [Karras et al., 2019]. From a quantitative point of view, our model obtains an average Inception Score (IS) equal to $1.63 \pm 0.03$ for background images and $2.66 \pm 0.02$ for target images, whereas MM-cVAE obtains $1.43 \pm 0.03$ and $1.44 \pm 0.01$ for background and target images respectively. Similar results were obtained using the Fréchet inception distance (FID). It is interesting to notice that our model, contrarily to MM-cVAE, preserves the characteristics of the salient elements, such as the opacity of the glasses. Both models struggle to preserve the shape of the original hat, although our method tends to generate a better hat but based on the hairstyle of the person. In Fig. 4, we present qualitative results where we generate images fixing a $z$ in each row and using different $s$ (0 for $X$, $\neq 0$ for $Y$). We can see that there is indeed a change of domain, and that the model generates a wide variety of images. When switching from background $X$ to target $Y$, the characteristics of the person are well preserved, and a salient feature is added, here glasses or hat. Furthermore, we can also notice that our model, being more accurate, is also more sensitive to dataset biases. For instance, we have noticed that in our dataset people with thin, transparent glasses are usually old men. This bias is clearly visible in the second row of Fig.3 and Fig.4. Removing such bias, as in [Barbano et al., 2023], is left as future work.

Cifar-10-MNIST dataset We create a new dataset based on Cifar-10 [Krizhevsky, 2009] and MNIST [LeCun, ]. Background images $X$ are Cifar-10 images, and target images $Y$ are also CIFAR-10 with a random MNIST digit overlaid on it. We use 50k training images, equally divided between $X$ and $Y$, and 10k test images, equally divided among the MNIST digits. Our model should successfully capture the background variability (i.e., CIFAR objects) only in the common

### Table 1: 5-fold average accuracy on Target CelebA (glasses vs hat). Std is always $\leq 0.01$, so we don’t report it for clarity. Best results in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Average $s_y \uparrow$</th>
<th>Worst $s_y \downarrow$</th>
<th>Average $z_y \uparrow$</th>
<th>Worst $z_y \downarrow$</th>
</tr>
</thead>
<tbody>
<tr>
<td>cVAE*</td>
<td>0.84</td>
<td>0.82</td>
<td>0.81</td>
<td>0.81</td>
</tr>
<tr>
<td>MM-cVAE*</td>
<td>0.85</td>
<td>0.82</td>
<td>0.78</td>
<td>0.80</td>
</tr>
<tr>
<td>double InfoGAN</td>
<td>0.95</td>
<td>0.95</td>
<td>0.94</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Figure 3: Image reconstruction and swap with the CelebA with accessories dataset.
We can notice that our method either outperforms MM-cVAE. Since we used $z$ when trained on Cifar-10 when trained on $z_y$ and 10% when trained on $s_y$.

We compare our model with MM-cVAE. Since we used the same image size as for CelebA ($64 \times 64 \times 3$), we kept the same network architecture. We tested several hyper-parameters for both methods and used the best configuration in our experiments. Results using two different latent space size are shown in Table 2 (for MM-cVAE, we use: $\lambda_1 = 10^2$, $\lambda_2 = 10^3$). As before, we run both methods 5 times (with different random seeds) for 500 epochs, and report the highest, average and lowest scores. More results in the Suppl.

We can notice that our method either outperforms MM-cVAE or obtains comparable results. Moreover, during our numerous trainings, we noticed that the results obtained with our method are very stable, while those obtained with MM-cVAE, as before with CelebA, are more variable and may diverge (i.e., non). Visual examples are presented in Fig. 5, with image reconstruction and salient feature swap (more in Supplementary). Our model offers sharper images than MM-cVAE and is able to better extract salient features.

**Ablation study** We present in Table 3 a detailed ablation study on the proposed losses using the Cifar-10-MNIST dataset and the architecture with a latent space of size 128 (since it obtained the best results in Table 2). We can notice that the proposed combination of losses obtains the best results.

**Brats dataset** In this section, we present qualitative results on the Brats dataset [Menze et al., 2014]. Background data $X$ contains T1-w MR brain images of healthy subject whereas the target dataset $Y$ has images of patients with brain tumors. Since images are bigger ($128 \times 128$) than the other datatsets, we use a different architecture. More details can be found in the Supplementary. Please note that here there are no sub-categories (as in previous datasets) that can be exploited to compute quantitative metrics (subgroup classification).

Fig. 6 shows fake images generated by our model trained on Brats. On the left are healthy images ($s = 0$), and on the right images with tumor ($s \neq 0$). Images in the same row are generated using the same $z$. We can see that the general anatomy of the brain is preserved when changing domain, and that tumors with different size and position are generated. By changing $z$ (i.e. row), we can also notice that the model seems to have correctly encoded in $z$ the general anatomical variability of the brain.

In Fig. 7, we generate healthy counterparts of target images with tumor, setting $s = 0$. This is very valuable in a clinical setting for multi-modal fusion [François et al., 2022, Maillard et al., 2022], where images from different modalities can exhibit a different topology due to the tumor, and atlas construction
Table 2: MNIST-Cifar10 classification. Digits information should only be encoded in $s_y$ and not in $z_y$, whereas the contrary should be true for Objects information. Std $\leq$ 0.01. Best results in **bold**.

<table>
<thead>
<tr>
<th></th>
<th>Mntst (salient)</th>
<th>Cifar (bg)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$s_y \uparrow$</td>
<td>$z_y \downarrow$</td>
</tr>
<tr>
<td>MM-cVAE (size 128)</td>
<td>0.81</td>
<td>0.76</td>
</tr>
<tr>
<td>MM-cVAE (size 200)</td>
<td>0.82</td>
<td>0.63</td>
</tr>
<tr>
<td>double InfoGAN (size 128)</td>
<td><strong>0.87</strong></td>
<td><strong>0.87</strong></td>
</tr>
<tr>
<td>double InfoGAN (size 200)</td>
<td>0.88</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Table 3: Ablation study of the different losses on the Cifar-MNIST dataset. For every configuration, 3 trainings were launched. We report average values.

<table>
<thead>
<tr>
<th></th>
<th>Mntst (salient)</th>
<th>Cifar (bg)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$s_y \uparrow$</td>
<td>$z_y \downarrow$</td>
</tr>
<tr>
<td>$-L_{\text{Class}}$</td>
<td>0.48</td>
<td>0.83</td>
</tr>
<tr>
<td>$-L_{\text{Class}} - L_{1\text{lin}}$</td>
<td>0.54</td>
<td>0.72</td>
</tr>
<tr>
<td>$-L_{\text{Info}} - L_{\text{Class}} - L_{1\text{lin}}$</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td>$-L_{\text{Info}}$</td>
<td>0.85</td>
<td>0.60</td>
</tr>
<tr>
<td>$-L_{\text{Info}} - L_{1\text{lin}}$</td>
<td>0.59</td>
<td>0.59</td>
</tr>
<tr>
<td>$-L_{\text{Class}} - L_{\text{Info}}$</td>
<td>0.74</td>
<td>0.67</td>
</tr>
<tr>
<td>$-L_{1\text{lin}}$</td>
<td>0.86</td>
<td><strong>0.25</strong></td>
</tr>
<tr>
<td>Full</td>
<td><strong>0.87</strong></td>
<td>0.26</td>
</tr>
</tbody>
</table>

**Figure 6:** Fake images generated by our model. In each row, we use the same $z$ for all images with $s = 0$ for $X$ and different $s \neq 0$ for each example of $Y$.

**dSprites-MNIST dataset** A new toy dataset is proposed for evaluating CA methods. The background dataset $X$ consists of 4 MNIST digits (1, 2, 3 and 4) regularly placed in a square. In the target dataset $Y$, dSprites element [Matthey et al., 2017] are added on top of the same 4 MNIST digits. Image reconstruction and salient feature swap are presented in Fig. 8. As before, we can see that, compared to MM-cVAE, image reconstructions are more accurate and sharp and, when exchanging salient features, the dSprites elements are better preserved.

**Disentanglement** As in [Higgins et al., 2017, Lin et al., 2020b], we also use dSprites to evaluate the disentanglement of our method in the salient space. Indeed, dSprites elements only exhibit 5 possible variations, making it easy to evaluate the disentanglement. Possible variations are: 1) shape (heart, ellipse and square), 2) size, 3) position in $X$, 4) position in $Y$ and 5) orientation (i.e. rotation). As metric, we use the FactorVAE (fvae) score [Kim and Mnih, 2018]. Initial results using the proposed method showed a very poor disentanglement. To further improve it, we adapted for our model the Contrastive Regularizer (CR) module of...
InfoGAN-CR [Lin et al., 2020b] (more details in the Supplementary), obtaining a maximum fvae score of 0.47. For comparison, InfoGAN-CR achieves a fvae score of 0.88 on the dsprite dataset alone. This shows that disentangling salient (or common) factors is much more difficult in our case than when using a single data-set. Exploring disentanglement regularizations more suited for a CA setting is left as future work.

In Fig. 9, we show target images generated by our model when varying only one dimension (from -1.5 to 1.5) of $s_y$, while keeping $z_y$ fixed. We clearly see a high entanglement among the dSprites factors of variation. For completeness, we also checked whether the CR module helped the separation between common and salient information, and found similar quantitative results (see Supplementary).

6 CONCLUSIONS AND PERSPECTIVES

We propose the first GAN-based model for Contrastive Analysis (CA) that estimates and separates in an unsupervised way all common and distinctive generative factors of a target dataset with respect to a background dataset. Compared to current SOTA CA-VAE models, we demonstrate superior performance on 4 visual datasets of increasing complexity and ranging from simple toy examples to real medical data. Our method manages to better separate common from salient factors, shows a better image generation quality and a greater stability during training. Furthermore, it allows the generation of multiple counterparts between domains by fixing the common factors and adding/removing the salient ones. We believe that the proposed method will benefit from more powerful GAN models and future progress in disentanglement, increasing its accuracy and interpretability. A last interesting research avenue could be the extension to the recent diffusion based models, as [Song et al., 2021, Rombach et al., 2022].

Limitations Recent works have shown that generative models, such as VAE and GAN, are in general not identifiable [Locatello et al., 2019]. To obtain identifiability, two different solutions have been proposed: 1) either regularizing [Kivva et al., 2022] / constraining (e.g., making it linear) the encoder or 2) introducing an auxiliary variable so that the latent factors are conditionally independent given the auxiliary variable [Hyvarinen et al., 2019, Khemakhem et al., 2020]. Unfortunately, in Contrastive Analysis, neither of these solutions may be used. While all losses proposed here, and in the related works, are needed to effectively separate common from salient factors, they do not assure that all true generative factors have been identified. This is the main limitation of this work, and actually of all concurrent CA-VAE models, and is left as future work. Inspired by [Wyner, 1975], a possible research direction would be adding an information-theoretic loss that quantifies the common and salient information content so that, under realistic assumptions, the model could be identifiable.

Acknowledgments This work was supported by the IDS department of Télécom Paris and by the L’association Télécom Paris Alumni.
References


Locatello et al., 2019 | Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and


Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]
A ETHICAL STATEMENT

This paper presents the Double InfoGAN method, which aims to improve the accuracy and interpretability of Contrastive Analysis in various fields, including medical imaging. We acknowledge that the use of medical imaging datasets raises ethical concerns that we discuss in the following. Firstly, we ensured that all medical imaging datasets used in our experiments were publicly available, anonymized and ethically sourced. Secondly, we aim at developing a method that improves the quality and interpretability of medical images, potentially leading to better diagnosis and treatment. We recognize that the accuracy and reliability of the generated medical images are crucial for clinical decision-making and that, as already mentioned in the article, more powerful GAN architectures with specific anatomical regularizations will be explored in future. Another ethical concern is the potential for biased outcomes, particularly when the datasets are not representative of the entire population. Biases in the data can lead to inaccurate and unfair results, which could exacerbate healthcare disparities or bring to wrong diagnosis and treatments. Therefore, it is important to ensure that datasets are diverse, inclusive and not biased, representing different demographics, hospitals, conditions, etc. In summary, while contrastive analysis can lead to significant improvements in healthcare and in other domains, it is essential to address potential ethical concerns, such as privacy and bias, to ensure accurate and trustworthy results.

B ARCHITECTURES

We present here the architectures used to obtain the results of this paper. These are inspired by DCGAN [Radford et al., 2016], with the addition of Gaussian Noise in the Discriminator. This prevents it from converging too fast and improves our performances. For the Brats dataset, we also added the spectral norm in the discriminator (instead of the batchnorm), and an attention block in the generator.

All computations are run on a server with one NVIDIA A100 GPU card and 64 AMD EPYC 7302 16-Core Processors.

<table>
<thead>
<tr>
<th>Discriminator D / C / Q</th>
<th>Generator G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input $64 \times 64 \times c$; $c=3$ for RGB, $c=1$ for gray/binary</td>
<td>Input $z \in \mathbb{R}^n$, $s \in \mathbb{R}^m$</td>
</tr>
<tr>
<td>Gaussian noise, $4 \times 4$ conv 64, stride 2, lReLU</td>
<td>concat and reshape $(z + s, 1, 1)$</td>
</tr>
<tr>
<td>Gaussian noise, $4 \times 4$ conv 128, stride 2, batchnorm, lReLU</td>
<td>$4 \times 4$ convtranspose 512, batchnorm, ReLU</td>
</tr>
<tr>
<td>Gaussian noise, $4 \times 4$ conv 256, stride 2, batchnorm, lReLU</td>
<td>$4 \times 4$ convtranspose 256, stride 2, batchnorm, ReLU</td>
</tr>
<tr>
<td>Gaussian noise, $4 \times 4$ conv 512, stride 2, batchnorm, lReLU (*)</td>
<td>$4 \times 4$ convtranspose 128, stride 2, batchnorm, ReLU</td>
</tr>
<tr>
<td>From *: Gaussian noise, $4 \times 4$ conv 1, sigmoid (output layer for D)</td>
<td>$4 \times 4$ convtranspose 64, stride 2, batchnorm, ReLU</td>
</tr>
<tr>
<td>From *: Gaussian noise, $4 \times 4$ conv 1, sigmoid (output layer for C)</td>
<td>$4 \times 4$ convtranspose 128, stride 2, batchnorm, ReLU</td>
</tr>
<tr>
<td>From *: Gaussian noise, $4 \times 4$ conv n (output layer for $Q_z$)</td>
<td>From *: Gaussian noise, $4 \times 4$ conv m (output layer for $Q_s$)</td>
</tr>
</tbody>
</table>

Table 4: Architecture for celeba, cifar/mnist and dsprite/mnist datasets. Gaussian Noise is added at every layer, with standard deviation of 0.2. LeakyReLU has a negative slope of 0.2. Batch size of 128. When a CR module is added, it has the same architecture as $Q_s$. 
Table 5: Architecture for Brats. Gaussian noise is additive, with standard deviation of 0.2. LeakyReLU has a negative slope of 0.2. Batch size of 32.

### C MATHEMATICAL DEVELOPMENTS

#### C.1 InfoGAN and InfoGAN-CR

The regularization loss proposed in InfoGAN [Chen et al., 2016] is:

\[
I(c; x) = -H(c|x) + H(c) = \mathbb{E}_{x \sim P(x), c \sim P(c|x)} \log P(c|x) + H(c) =
\]

\[
= \int P(x) \int P(c|x) \log(P(c|x)) dc + H(c) =
\]

\[
= \int P(x) \int P(c'|x) \log(P(c'|x)) dc + H(c') = (\text{change of variables between } c \text{ and } c')
\]

\[
= \int \int \int P(x, c, z) dz dc \int P(c'|x) \log(P(c'|x)) dc + H(c') =
\]

\[
= \int P(z) \int P(c|x) \int P(x|c, z) \int P(c'|x) \log(P(c'|x)) dz dc dc + H(c') =
\]

\[
= \int P(z) \int P(c|x) \int P(x|c, z) \int P(c'|x) \log(Q(c'|x)) dz dc dc + H(c') = (\text{identity trick})
\]

\[
= \mathbb{E}_{z \sim P(z), c \sim P(c, x \sim P(x|c, z))} KL(P(c'|x)||Q(c'|x)) + \mathbb{E}_{z \sim P(z), c \sim P(c, x \sim P(x|c, z), z \sim P(z|c,x))} \log Q(c'|x) + H(c')
\]

\[
\geq \mathbb{E}_{z \sim P(z), c \sim P(c, x \sim P(x|c, z))} \log(Q(c'|x)) + H(c')
\]

where we have introduced an auxiliary distribution \(Q(c|x)\), parameterized as a neural network, to approximate the posterior \(P(c|x)\) (which is difficult to compute) and we have made the hypothesis that \(c\) does not depend on \(z\) (i.e., \(P(c|z) = P(c)\)). To further remove also the need to sample from \(P(c|x)\) (which would be impossible in most cases), authors propose a simple, yet effective, modification of the previous variational lower bound. In their algorithm, they actually compute and maximize: \(\mathcal{L}(G, Q) = \mathbb{E}_{z \sim P(z), c \sim P(c, x \sim P(x|c, z))} \log(Q(c|x)) + H(c)\), which is equivalent to the previous lower bound:
\[ L(G, Q) = \mathbb{E}_{z \sim P(z), c \sim P(c), x \sim P(x|c, z)} \log(Q(c|x)) + H(c) = \\
= \int P(z) \int P(c) \int P(x|c, z) \log(Q(c|x)) dz dc dx + H(c) = \\
= \int P(z) \int P(c) \int P(x, c|z) \log(Q(c|x)) dz dc dx + H(c) = \\
= \int P(z) \int \int P(x, c'|z) \log(Q(c'|x)) dz dc dx + H(c') = \quad \text{(change of variable between c and c')} \\
= \int P(z) \int \int P(x, c|z) P(c'|x) \log(Q(c'|x)) dz dc dx + H(c') = \quad \text{(c' doesn’t depend on z, re-introduce c)} \\
= \int P(z) \int \int P(c) P(x|c, z) P(c'|x) \log(Q(c'|x)) dz dc dx + H(c') = \\
= \mathbb{E}_{z \sim P(z), c \sim P(c), x \sim P(x|c, z)} \mathbb{E}_{c' \sim P(c'|x)} \log(Q(c'|x)) + H(c') \quad \text{(10)} \\
\]

In [Chen et al., 2016], authors proposed to model the auxiliary conditional distribution \( Q(c|x) \) as a factorized Gaussian with identity covariance \( Q(c|x) = \prod_i (c_i|x_i) = \prod_i N(\mu_i(x_i), 1) \). As shown in [Lin et al., 2020a], this is fundamental for stability and efficiency. Furthermore, in [Lin et al., 2020a], authors also showed that informativeness alone does not necessarily encourage disentanglement. To this end, they propose a new regularizer, called Contrastive Regularizer (CR), which enforces distinguishable visual changes in the images created using different latent codes. More specifically, they propose to fix a latent code \( c_i \), draw the others \( \{c_j\}_{j \neq i} \) uniformly at random, and then sample two or more images \( x_i \) from the resulting distribution \( (x_i) \sim Q^i \). By repeating this process for all \( k \) latent codes \( c_i \), one can obtain an estimate of all distributions \( Q^i \). The goal, following the usual definition of disentanglement, is then to maximize the difference between the distributions \( Q^i \), so that each latent code \( c_i \) should encode a specific visual variation in the created images that should be noticeable and easy to distinguish from the patterns encoded by the other latent codes \( \{c_j\}_{j \neq i} \). Authors propose to maximize the following loss:

\[ L_c = d_{JS}(Q^1, ..., Q^k) : = \frac{1}{k} \sum_i KL(Q^i||\frac{\sum_j Q^j}{k}) \quad \text{(11)} \]

They propose to approximate this regularization term using a discriminator \( H \) that performs multi-way hypothesis testing. Given two or more images \( (x_i) \), created by fixing only one latent code \( c_i \), the discriminator \( H \) needs to identify the latent dimension \( i \) shared between the images. Authors claim, and experimentally demonstrate, that by updating both discriminator \( H \) and generator \( G \) to maximize \( L_c \), it “should encourage each latent code \( c_i \) to make distinct and noticeable changes, hence promoting disentanglement”.

We adapted the CR module for our model, as explained in Fig.10.

![Figure 10: Contrastive regularizer. Two images are generated with the following constraints: the background latent space \( z \) is the same, and all the salient latent code are different except one (named \( k \) here). From these two images, the CR module is then trained to predict which one of the latent code was identical. The CR module outputs a vector of probabilities of the same size as \( s \).](image-url)
C.2 Double InfoGAN

Since \( z \) and \( s \) are supposed to be independent, the mutual information \( I((z, s); y) \) can be decomposed into the sum of the two mutual information \( I(z; y) + I(s; y) \)

\[
I((z, s); y) = -H((z, s)|y) + H(z, s) = -H((z, s)|y) + H(z) + H(s) =
\]

\[
= \int \int P(y) \int \int P(z, s, y) \log \left( \frac{P(z, s, y)}{P(y)} \right) dz ds dy + H(z) + H(s) =
\]

\[
= \int P(y) \int \int P(s|y) P(z|y) (\log(P(s|y)P(z|y))) dz ds dy + H(z) + H(s) =
\]

\[
= \int P(y) \int \int (P(s|y) + \log(P(z|y))) dz ds dy + H(z) + H(s) =
\]

\[
E_{y \sim P(y), s \sim P(s|y)} \log(P(s|y)) + E_{z \sim P(z), z \sim P(z|y)} \log(P(z|y)) + H(z) + H(s) =
\]

\[
= -H(s|y) - H(z|y) + H(z) + H(s) = I(z; y) + I(s; y)
\]

(12)

The log-likelihood \( \log(P(y)) \) of the generated images based on the proposed model is:

\[
\log P(y) = \log \int \int P(y, z, s, y_R) dz ds dy R = \log \int \int P(y, z, s, y_R) P(z, s|y_R) P(y_R) \frac{Q(z, s|y_R)}{Q(z, s|y_R)} dz ds dy R
\]

\[
= \log E_{y_R \sim P(y_R), (z, s) \sim Q(z, s|y_R)} P(y|z, s, y_R) \frac{P(z, s|y_R)}{Q(z, s|y_R)} \geq E_{y_R \sim P(y_R), (z, s) \sim Q(z, s|y_R)} \log P(y|z, s, y_R) \frac{P(z, s|y_R)}{Q(z, s|y_R)} =
\]

\[
= E_{y_R \sim P(y_R), (z, s) \sim Q(z, s|y_R)} \log P(y|z, s, y_R) - E_{y_R \sim P(y_R)} KL(Q(z, s|y_R)||P(z, s|y_R))
\]

(13)

D LOSSES EQUATIONS

In this section, we detail the different losses used in our paper as functions of the involved modules: discriminator, generator, encoder. Notation are as follows: from an image \( I, D(I) \) will be the standard adversarial output of the GAN discriminator (namely the probability that an image is real or fake), \( C(I) \) will be the class predicted by the discriminator from image \( I \), and \( Q(I) \) will be the predicted \( \hat{z} \) and \( \hat{s} \). In practice, \( Q, C \) and \( D \) share most of the layers (see detailed architecture).

Reminder of the global loss:

\[
\min_{G, Q, H, C} \max_D \quad w_{Adv} \cdot L_{Adv}(G, D) + w_{Class} \cdot L_{Class}(G, C) + w_{Inf} \cdot L_{Inf}(G, Q) + w_{Im} \cdot L_{Im}(G, Q) + w_{CR} \cdot L_{CR}(G, H)
\]

(14)

D.1 Adversarial GAN Loss

Similarly to [Goodfellow et al., 2014], the adversarial loss used here is:

\[
L_{Adv}(D, G) = w_g \left( -E_{x_R \sim P(x_R)} [\log(D(x_R))] - E_{z \sim P(z)} [\log(1 - (D(G(z, 0))))] \right)
\]

\[
+ w_l \left( -E_{y_R \sim P(y_R)} [\log(D(y_R))] - E_{z, s \sim P(z, s)} [\log(1 - (D(G(z, s))))] \right)
\]

(15)

D.2 Info Loss

The weight \( w_{Inf} \) is actually divided in three components : \( w_{Inf}^s, w_{Inf}^z \) and \( w_{Inf}^{real} \)

\[
L_{Inf}(G, Q) = w_g E_{z \sim P(z)} [w_{Inf}^s Q_z(G(z, 0)) - z] + w_{Inf}^z Q_z(G(z, 0)) - z] + w_{Inf}^{real} E_{x_R \sim P(x_R)} [(Q_s(x_R)) - 0]
\]

(16)
### D.3 Image reconstruction loss

We also use an image reconstruction loss, which depends on $G$ and $Q$:

$$
\mathcal{L}_{Im}(G, Q) = w_{bg} \mathbb{E}_{x_R \sim P(x_R), \hat{z} = Q(z)} \left[ |G(\hat{z}, 0) - x_R| \right] + w_t \mathbb{E}_{y_R \sim P(y_R), \hat{z}, \hat{s} = Q(y_R)} \left[ |G(\hat{z}, \hat{s}) - y_R| \right] 
$$

#### (17)

### D.4 CR Loss

The Contrastive Regularization loss is computed to improve the disentanglement of $s$. The module $H$ and the generator $G$ are trained using a Cross Entropy loss to find the salient feature $k$ in common between two images generated with two salient factors that have only one factor in common (i.e., $s^1_k = s^2_k$) and all other factors different (i.e., $s^1_j \neq s^2_j$ with $\forall j \neq k$):

$$
\mathcal{L}_{CR}(G, H) = \mathbb{E}_{k \in \mathbb{N}, z \sim P(z), s^1, s^2 \sim P(s), s^1_k = s^2_k, s^1_j \neq s^2_j, \forall j \neq k} \left[ CE(H(G(z, s^1), G(z, s^2)), k) \right] 
$$

#### (18)

### D.5 Weights and ratio between the losses

Different weights are used to balance the different losses, as well as other hyper-parameters (learning rate, number of epochs, etc.). Table 6 summarizes the hyper-parameters used in our experiments and their values for each dataset employed.

---

**E EXTENSIVE RESULTS**

**Figure 13:** Swap using the cVAE method with the CelebA with accessories dataset. We can clearly see that personal traits (which should be encoded in the common space) are lost during the swap and accessories (which should be encoded in the salient space) are not always correctly added. This could explain the poor quantitative performance of cVAE.
Figure 11: Image reconstruction and swap with CelebA. In every block, first row refers to $X$ and second row to $Y$. It’s interesting to notice that rare attributes, such as the glasses in the first two blocks, are correctly reconstructed by our method and not by MM-cVAE. However, they are changed towards more "common" glasses after swapping.
Table 7: Target Dataset separation on CelebA - glasses vs hat - for the 5 trainings of the three methods at different epochs. For clarity, we don’t report the standard deviations, whose values are between 0.00 and 0.01. We can notice that the trainings of MMc-VAE are less stable than the ones of our method.
Table 8: Target Dataset separation on Cifar-10-MNIST for MM-cVAE and our method, with and without CR.

<table>
<thead>
<tr>
<th>Training</th>
<th>epoch 100</th>
<th>epoch 300</th>
<th>epoch 500</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mnist (salient)</td>
<td>Cifar (bg)</td>
<td>Mnist (salient)</td>
</tr>
<tr>
<td>Training 1</td>
<td>0.68 0.57</td>
<td>0.16 0.36</td>
<td>0.68 0.51</td>
</tr>
<tr>
<td>Training 2</td>
<td>0.80 0.51</td>
<td>0.15 0.36</td>
<td>0.80 0.41</td>
</tr>
<tr>
<td>Training 3</td>
<td>0.81 0.56</td>
<td>0.27 0.33</td>
<td>0.81 0.52</td>
</tr>
<tr>
<td>Training 4</td>
<td>0.81 0.53</td>
<td>0.31 0.25</td>
<td>nan   nan   nan   nan</td>
</tr>
<tr>
<td>Training 5</td>
<td>0.71 0.57</td>
<td>0.16 0.35</td>
<td>0.73 0.51</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training</th>
<th>epoch 100</th>
<th>epoch 300</th>
<th>epoch 500</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mnist (salient)</td>
<td>Cifar (bg)</td>
<td>Mnist (salient)</td>
</tr>
<tr>
<td>Training 1</td>
<td>0.19 0.82</td>
<td>0.12 0.35</td>
<td>0.14 0.82</td>
</tr>
<tr>
<td>Training 2</td>
<td>0.80 0.55</td>
<td>0.17 0.36</td>
<td>0.80 0.44</td>
</tr>
<tr>
<td>Training 3</td>
<td>0.81 0.64</td>
<td>0.28 0.34</td>
<td>0.82 0.57</td>
</tr>
<tr>
<td>Training 4</td>
<td>0.66 0.67</td>
<td>0.18 0.36</td>
<td>0.64 0.63</td>
</tr>
<tr>
<td>Training 5</td>
<td>0.76 0.60</td>
<td>0.19 0.36</td>
<td>0.78 0.51</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training</th>
<th>epoch 100</th>
<th>epoch 300</th>
<th>epoch 500</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mnist (salient)</td>
<td>Cifar (bg)</td>
<td>Mnist (salient)</td>
</tr>
<tr>
<td>Training 1</td>
<td>0.89 0.28</td>
<td>0.19 0.42</td>
<td>0.89 0.30</td>
</tr>
<tr>
<td>Training 2</td>
<td>0.89 0.29</td>
<td>0.21 0.41</td>
<td>0.88 0.31</td>
</tr>
<tr>
<td>Training 3</td>
<td>0.90 0.28</td>
<td>0.19 0.42</td>
<td>0.88 0.32</td>
</tr>
<tr>
<td>Training 4</td>
<td>0.90 0.29</td>
<td>0.20 0.41</td>
<td>0.88 0.30</td>
</tr>
<tr>
<td>Training 5</td>
<td>0.89 0.28</td>
<td>0.19 0.41</td>
<td>0.88 0.30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training</th>
<th>epoch 100</th>
<th>epoch 300</th>
<th>epoch 500</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mnist (salient)</td>
<td>Cifar (bg)</td>
<td>Mnist (salient)</td>
</tr>
<tr>
<td>Training 1</td>
<td>0.88 0.21</td>
<td>0.16 0.40</td>
<td>0.88 0.27</td>
</tr>
<tr>
<td>Training 2</td>
<td>0.87 0.24</td>
<td>0.16 0.39</td>
<td>0.88 0.27</td>
</tr>
<tr>
<td>Training 3</td>
<td>0.88 0.22</td>
<td>0.16 0.39</td>
<td>0.87 0.26</td>
</tr>
<tr>
<td>Training 4</td>
<td>0.87 0.25</td>
<td>0.17 0.40</td>
<td>0.88 0.25</td>
</tr>
<tr>
<td>Training 5</td>
<td>0.88 0.24</td>
<td>0.15 0.39</td>
<td>0.87 0.24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training</th>
<th>epoch 100</th>
<th>epoch 300</th>
<th>epoch 500</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mnist (salient)</td>
<td>Cifar (bg)</td>
<td>Mnist (salient)</td>
</tr>
<tr>
<td>Training 1</td>
<td>0.88 0.25</td>
<td>0.18 0.42</td>
<td>0.87 0.28</td>
</tr>
<tr>
<td>Training 2</td>
<td>0.87 0.23</td>
<td>0.16 0.40</td>
<td>0.88 0.24</td>
</tr>
<tr>
<td>Training 3</td>
<td>0.88 0.21</td>
<td>0.18 0.40</td>
<td>0.88 0.26</td>
</tr>
<tr>
<td>Training 4</td>
<td>0.88 0.26</td>
<td>0.17 0.39</td>
<td>0.88 0.27</td>
</tr>
<tr>
<td>Training 5</td>
<td>0.89 0.26</td>
<td>0.17 0.40</td>
<td>0.88 0.25</td>
</tr>
</tbody>
</table>
Figure 12: More examples of image reconstruction and swap with Cifar-10-MNIST dataset.
### Table 9: Ablation study of CR Loss on Mnist-dsprite dataset. Score indicated is fvae score

<table>
<thead>
<tr>
<th>Training</th>
<th>epoch 50</th>
<th>epoch 100</th>
<th>epoch 200</th>
<th>epoch 300</th>
<th>epoch 400</th>
<th>epoch 500</th>
<th>epoch 600</th>
<th>epoch 700</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training 1</td>
<td>0.334</td>
<td>0.366</td>
<td>0.366</td>
<td>0.376</td>
<td>0.394</td>
<td>0.396</td>
<td>0.398</td>
<td>0.4</td>
</tr>
<tr>
<td>Training 2</td>
<td>0.414</td>
<td>0.37</td>
<td>0.338</td>
<td>0.328</td>
<td>0.314</td>
<td>0.314</td>
<td>0.282</td>
<td>0.294</td>
</tr>
<tr>
<td>Training 3</td>
<td>0.258</td>
<td>0.294</td>
<td>0.25</td>
<td>0.258</td>
<td>0.222</td>
<td>0.222</td>
<td>0.218</td>
<td>0.216</td>
</tr>
<tr>
<td>Training 4</td>
<td>0.35</td>
<td>0.346</td>
<td>0.346</td>
<td>0.402</td>
<td>0.43</td>
<td>0.454</td>
<td>0.49</td>
<td><strong>0.494</strong></td>
</tr>
<tr>
<td>Training 5</td>
<td>0.202</td>
<td>0.202</td>
<td>0.206</td>
<td>0.216</td>
<td>0.22</td>
<td>0.222</td>
<td>0.226</td>
<td>0.24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training</th>
<th>epoch 50</th>
<th>epoch 100</th>
<th>epoch 200</th>
<th>epoch 300</th>
<th>epoch 400</th>
<th>epoch 500</th>
<th>epoch 600</th>
<th>epoch 700</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training 1</td>
<td>0.348</td>
<td>0.358</td>
<td>0.364</td>
<td>0.376</td>
<td>0.39</td>
<td>0.392</td>
<td>0.406</td>
<td><strong>0.424</strong></td>
</tr>
<tr>
<td>Training 2</td>
<td>0.348</td>
<td>0.354</td>
<td>0.372</td>
<td>0.37</td>
<td>0.378</td>
<td>0.396</td>
<td>0.398</td>
<td>0.398</td>
</tr>
<tr>
<td>Training 3</td>
<td>0.326</td>
<td>0.35</td>
<td>0.354</td>
<td>0.342</td>
<td>0.334</td>
<td>0.324</td>
<td>0.306</td>
<td>0.306</td>
</tr>
<tr>
<td>Training 4</td>
<td>0.292</td>
<td>0.292</td>
<td>0.288</td>
<td>0.306</td>
<td>0.316</td>
<td>0.358</td>
<td>0.348</td>
<td>0.386</td>
</tr>
<tr>
<td>Training 5</td>
<td>0.306</td>
<td>0.32</td>
<td>0.318</td>
<td>0.292</td>
<td>0.304</td>
<td>0.324</td>
<td>0.334</td>
<td>0.34</td>
</tr>
</tbody>
</table>