Double InfoGAN for Contrastive Analysis

Florence Carton[1] **Robin Louiset**[1,2] **Pietro Gori**[1]
[1]LTCI, Télécom Paris, IPParis, France [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 sub-optimal 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[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 $\mathbf{z}$ and salient $\mathbf{s}$, whereas samples from the control dataset are generated using only the common $\mathbf{z}$ factors. The salient factors $\mathbf{s}$ 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 $\mathbf{z}$ and $\mathbf{s}$ 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

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

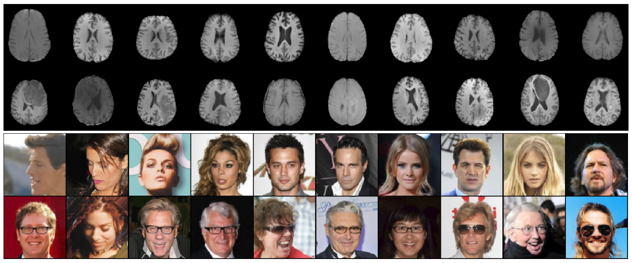

Figure 1: Two examples of datasets for Contrastive Analysis. **First figure**: Brats dataset [Menze et al., 2014]. Top: MRI images of healthy brains (control dataset). Bottom: MRI images of brains with tumor (target dataset). **Second figure**: CelebA dataset. Top: control dataset with regular faces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

based 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

[1] `https://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 Kügelgen 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) $\mathbf{z}$, which is considered as a nuisance and incompressible noise and 2) $\mathbf{c}$, which should model the salient semantic features of the data distribution. The generator of this new model, $G(\mathbf{z}, \mathbf{c})$, takes as input both $\mathbf{z}$ and $\mathbf{c}$ to generate samples $\mathbf{x}$. As shown in [Chen et al., 2016], without regularisation, the generator G may ignore the additional code $\mathbf{c}$ or find a trivial (and useless) solution. To this end, authors propose to regularize the estimate of G by maximizing the mutual information $I(\mathbf{c}; \mathbf{x})$ between $\mathbf{c}$ and $\mathbf{x} \sim G(\mathbf{z}, \mathbf{c})$. Maximum I is obtained when $\mathbf{c}$ and $\mathbf{x}$ are completely dependent and one becomes completely redundant with the knowledge of the other. This should increase the informativeness of $\mathbf{c}$, namely all salient semantic information should be in $\mathbf{c}$ and not in $\mathbf{z}$, which should only account for additional randomness (i.e., noise). Authors propose to maximize a lower bound of $I(\mathbf{c}; \mathbf{x})$ by defining an auxiliary distribution $Q(\mathbf{c}|\mathbf{x})$, parameterized as a neural network, to approximate $P(\mathbf{c}|\mathbf{x})$:

$$I(\mathbf{c}; \mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim P(\mathbf{z}), \mathbf{c}\sim P(\mathbf{c}), \mathbf{x}\sim P(\mathbf{x}|\mathbf{c},\mathbf{z})} \log(Q(\mathbf{c}|\mathbf{x})) + H(\mathbf{c}) \tag{1}$$

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 $X = \{\mathbf{x}_i\}$ and $Y = \{\mathbf{y}_j\}$ be the background (or control) and target data-sets of images respectively. Both $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_j\}$ are assumed to be i.i.d. from two different and unknown distributions ($P(\mathbf{x})$ and $P(\mathbf{y})$) that depend on a pair of latent variables ($\mathbf{z}$, $\mathbf{s}$). Here, $\mathbf{s}$ is assumed to capture the salient generative factors proper only to Y whereas $\mathbf{z}$ should describe the common generative factors between X and Y. The generative models (i.e. same decoder with parameters θ) are: $\mathbf{x}_i \sim P_\theta(\mathbf{x}|\mathbf{z}_i, \mathbf{s}_i = s')$ and $\mathbf{y}_j \sim P_\theta(\mathbf{y}_j|\mathbf{z}_j, \mathbf{s}_j)$, where the salient factors $\mathbf{s}_i$ of X are fixed to a constant value s' (e.g., $s' = 0$), thus enforcing $\mathbf{z}$ 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, $Q_\phi(\mathbf{z}_i, \mathbf{s}_i|\mathbf{x}_i)$ and $Q_\phi(\mathbf{z}_j, \mathbf{s}_j|\mathbf{y}_j)$, which are usually assumed to be conditional independent: $Q_\phi(\mathbf{z}, \mathbf{s}|\cdot) = Q_\phi(\mathbf{z}|\cdot)Q_\phi(\mathbf{s}|\cdot)$. The latent generative factors ($\mathbf{z}$, $\mathbf{s}$) are also usually assumed to be independent (i.e., $P_\cdot(\mathbf{z}, \mathbf{s}) = P_\cdot(\mathbf{z})P_\cdot(\mathbf{s})$). The common factor $\mathbf{z}$ should follow the same prior distribution in X and Y (e.g., $P_x(\mathbf{z}) = P_y(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, \mathcal{I})$). The salient factor $\mathbf{s}$ follows instead a different prior distribution between X and Y, such as $P_y(\mathbf{s}) = \mathcal{N}(\mathbf{s}; 0, \mathcal{I})$ and $P_x(\mathbf{s}) = \delta(\mathbf{s} = s')$, the Dirac distribution centered at s'. Based on this generative latent variable model, one can derive a lower bound of the marginal log likelihood:

$$\begin{aligned}\log P(\mathbf{x}) \geq & \mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})Q_\phi(\mathbf{s}|\mathbf{x})} \log P_\theta(\mathbf{x}|\mathbf{z}, \mathbf{s}) - \\ & KL(Q_\phi(\mathbf{z}|\mathbf{x})||p_{\mathbf{x}}(z)) - KL(Q_\phi(\mathbf{s}|\mathbf{x})||p_{\mathbf{x}}(s))\end{aligned} \tag{2}$$

and similarly for $\log P(\mathbf{y})$. 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 factors[2] and most of them ignore the KL divergence

[2] [Abid and Zou, 2019] proposed to minimize the total correlation (TC) between $q_{\phi_z,\phi_s}(z, s|x)$ and $q_{\phi_z}(z|x)q_{\phi_s}(s|x)$ via the density-ratio trick

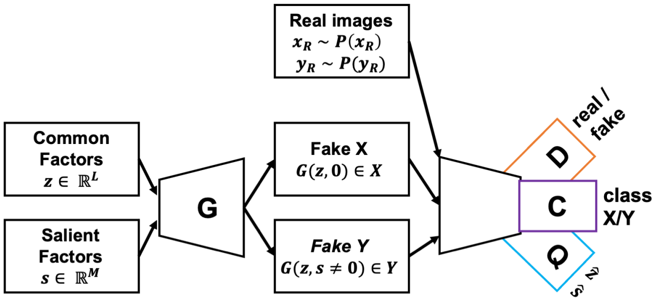

Figure 2: **Double InfoGAN**. Our model takes two inputs: $\mathbf{z}$ (common factors) and $\mathbf{s}$ (salient factors). The generator G produces fake images that, together with the real images, are passed to a discriminator and encoder. The discriminator has two modules: D for detecting real from fake images, and C for classyfing images in the correct domain (i.e., X or Y). The encoder Q has two modules, Q_z and Q_s, to reconstruct the latent factors $(\mathbf{z}, \mathbf{s})$. D, C and Q share all layers but the last one.

term about $p_{\mathbf{y}}(s)$ (except [Choudhuri et al., 2019] and [Weinberger et al., 2022]), thus allowing a possible information leakage between salient and common factors, as discussed in [Weinberger et al., 2022]. Furthermore, the quality of the generated images is rather poor.

4 METHOD - Double InfoGAN

Model In Double InfoGAN, we use a generative model similar to the one proposed in CA-VAE but within the framework of InfoGAN. We suppose that the background images $\{\mathbf{x}_i\} \overset{\text{iid}}{\sim} P(\mathbf{x})$ and the target images $\{\mathbf{y}_j\} \overset{\text{iid}}{\sim} P(\mathbf{y})$, where $P(\mathbf{x})$ and $P(\mathbf{y})$ are unknown and depend on a pair of latent variables $(\mathbf{z} \in \mathbb{R}^L, \mathbf{s} \in \mathbb{R}^M)$. Differently from InfoGAN, and similarly to CA-VAE, $\mathbf{z}$ should now capture the generative factors common to both X and Y whereas $\mathbf{s}$ the salient factors proper only to Y. As in GAN [Goodfellow et al., 2014], we introduce a generator G and a discriminator. The generator G should generate samples that are indistinguishable from the true ones, whereas the discriminator is divided into two modules. The first (and standard) one D is trained to discriminate between fake and real samples. The second module C is trained to correctly classify real samples (i.e., X or Y). As in InfoGAN, we also use one encoder, divided into two modules, Q_z and Q_s, to reconstruct the latent factors $\mathbf{z}$ and $\mathbf{s}$. The discriminator, D and C, and the encoder, Q_z and Q_s, are parametrized as neural networks, that share all layers but the output one.

Let $\mathbf{x} = G(\mathbf{z}, \mathbf{s} = s')$ and $\mathbf{y} = G(\mathbf{z}, \mathbf{s})$ be the generated samples. We suppose, and force it in practice, that the latent variables $\mathbf{z} = \{z_1, ..., z_L\}$ and $\mathbf{s} = \{s_1, ..., s_M\}$ are independent and follow a factorized distribution: $P(\mathbf{z}) = \prod_{i=1}^{L} P(c_z)$ and $P(\mathbf{s}) = \prod_{j=1}^{M} P(s_j)$, for X and Y. The total cost function is:

$$\min_{G,Q_z,Q_sC} \max_{D} \quad w_{Adv}\mathcal{L}_{Adv}(G,D) + w_{Class}\mathcal{L}_{cl}(G,C) - w_{Info}\mathcal{L}_{Info}(G,Q_z,Q_s) + w_{Im}\mathcal{L}_{Im}(G,Q_z,Q_s) \tag{3}$$

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:

$$\mathcal{L}_{Adv}(D,G) = w_{bg}\Big(-\mathbb{E}_{\mathbf{x}_R \sim P(\mathbf{x}_R)}\big[\log(D(\mathbf{x}_R)\big] - \mathbb{E}_{z \sim P_x(\mathbf{z})}\big[\log(1 - (D(G(\mathbf{z}, 0))))\big]\Big) + w_t\Big(-\mathbb{E}_{\mathbf{y}_R \sim P(\mathbf{y}_R)}\big[\log(D(\mathbf{y}_R)\big] - \mathbb{E}_{\mathbf{z},\mathbf{s} \sim P_y(\mathbf{z},\mathbf{s})}\big[\log(1 - (D(G(\mathbf{z}, \mathbf{s}))))\big]\Big) \tag{4}$$

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

$$\begin{aligned}\mathcal{L}_{cl}(C) =& \mathbb{E}_{\mathbf{x}_R \sim P(\mathbf{x}_R)}\big[\mathcal{B}(C(\mathbf{x}_R), 0)\big] \\ &+ \mathbb{E}_{\mathbf{y}_R \sim P(\mathbf{y}_R)}\big[\mathcal{B}(C(\mathbf{y}_R), 1)\big] \\ \mathcal{L}_{cl}(G) =& \mathbb{E}_{\mathbf{z} \sim P_x(\mathbf{z})}[\mathcal{B}(C(G(\mathbf{z}, 0)), 0)] \\ &+ \mathbb{E}_{\mathbf{z},\mathbf{s} \sim P_y(\mathbf{z},\mathbf{s})}[\mathcal{B}(C(G(\mathbf{z}, \mathbf{s})), 1)]\end{aligned} \tag{5}$$

Info Loss Similarly to InfoGAN, we propose two regularization terms based on mutual information, $I((\mathbf{z}, \mathbf{s}); \mathbf{y})$ and $I((\mathbf{z}, \mathbf{s} = s'); \mathbf{x})$, to encourage informative latent codes. However, in our case, these two

[Kim and Mnih, 2018], but their implementation is inaccurate since they don't use an independent optimizer.

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 $\mathbf{z}$ to fully encode X and at the same time to be informative for the generation of Y. In parallel, $\mathbf{s}$ 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 $\mathbf{z}$ and $\mathbf{s}$ are independent *by construction*, the mutual information $I((\mathbf{z}, \mathbf{s}); \cdot)$ can be decomposed into the sum of the two mutual information $I(\mathbf{z}; \cdot) + I(\mathbf{s}; \cdot)$. 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_z and Q_s, as factorized distributions. Beside a factorized Gaussian distribution with identity covariance, we have also tested a factorized Laplace distribution $\mathbf{L}(\mu, b)$ 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 $\mathbf{s}$ should be equal to 0 for real images of domain X (i.e., $\mathbf{x}_R \sim P(\mathbf{x}_R)$), we also add as regularization the lower bound of the mutual information $I(\mathbf{s}; \mathbf{x}_R)$. As before, we fix $P_x(\mathbf{s}) = \delta(\mathbf{s} = 0)$. The sum of these five lower bounds defines the $\mathcal{L}_{Info}$ loss:

$$\begin{aligned}
&\mathcal{L}_{Info}(G, Q_z, Q_s) = w_{bg}\mathbb{E}_{\mathbf{z}\sim P_y(\mathbf{z})}\big[w^z_{Info}|(Q_z(G(z,0)) - z| \\
&+ w^s_{Info}|Q_s(G(z,0)) - 0|\big] \\
&+ w_t\mathbb{E}_{\mathbf{z},\mathbf{s}\sim P_y(\mathbf{z},\mathbf{s})}\big[w^z_{Info}|(Q_z(G(z,s)) - z| \\
&+ w^s_{Info}|Q_s(G(z,s)) - s|\big] \\
&+ w^{real}_{Info}\mathbb{E}_{\mathbf{x}_R\sim P(\mathbf{x}_R)}\big[|(Q_s(\mathbf{x}_R)) - 0|\big]
\end{aligned} \tag{6}$$

Image reconstruction loss Differently from usual GAN models, we also propose to maximize the log-likelihood $\log(P(\mathbf{y}))$ (and $\log(P(\mathbf{x}))$) 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 $\mathbf{y}_R$ (or $\mathbf{x}_R$), we can use the auxiliary encoder $Q = (Q_s, Q_z)$ to estimate the latent factors $\hat{z}$ and $\hat{s}$ that should generate $\mathbf{y}_R$ (or $\mathbf{x}_R$) and then maximize (an approximation) of the log-likelihood of the generated images $\mathbf{y} = G(\hat{z}, \hat{s})$ (or $\mathbf{x} = G(\hat{z}, 0)$):

$$\begin{aligned}
\log P(\mathbf{y}) \geq\ & \mathbb{E}_{\mathbf{y}_R\sim P(\mathbf{y}_R),(\mathbf{z},\mathbf{s})\sim Q(\mathbf{z},\mathbf{s}|\mathbf{y}_R)} \log P(\mathbf{y}|\mathbf{z},\mathbf{s},\mathbf{y}_R) \\
& - \mathbb{E}_{\mathbf{y}_R\sim P(\mathbf{y}_R)} KL(Q(\mathbf{z},\mathbf{s}|\mathbf{y}_R)||P(\mathbf{z},\mathbf{s}|\mathbf{y}_R))
\end{aligned} \tag{7}$$

We notice that the second term should tend towards 0 during training thanks to the previous Info Loss.[3] We can thus approximate $\log P(\mathbf{y})$ by computing only the left term and modeling $P(\mathbf{y}|\mathbf{z}, \mathbf{s}, \mathbf{y}_R)$ as a Laplace distribution $\mathbf{L}(\mu, b)$ 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 $\log P(\mathbf{x})$. We define $\mathcal{L}_{Im}(G, Q_z, Q_s) = \log P(\mathbf{x}) + \log P(\mathbf{y})$:

$$\begin{aligned}
\mathcal{L}_{Im}(G, Q_z, Q_s) = w_{bg} & \mathop{\mathbb{E}}_{\substack{\mathbf{x}_R\sim P(\mathbf{x}_R) \\ \hat{z}=Q_z(\mathbf{x}_R)}} \big[|G(\hat{z}, 0) - \mathbf{x}_R|\big] \\
+ w_t & \mathop{\mathbb{E}}_{\substack{\mathbf{y}_R\sim P(\mathbf{y}_R) \\ \hat{z},\hat{s}=Q(\mathbf{y}_R)}} \big[|G(\hat{z}, \hat{s}) - \mathbf{y}_R|\big]
\end{aligned} \tag{8}$$

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 $\mathbf{x}_R \in X$ and $\mathbf{y}_R \in Y$, we can first estimate the latent factors $(\widehat{z_X}, \widehat{s_X})$ and $(\widehat{z_y}, \widehat{s_Y})$, that should have generated $\mathbf{x}_R$ and $\mathbf{y}_R$, using Q. Then, we can swap the estimated salient features $\widehat{s_X}$ and $\widehat{s_Y}$, and re-generate the images $G(\widehat{z_X}, \widehat{s_Y})$ and $G(\widehat{z_Y}, \widehat{s_X})$.

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.

	$\hat{s}_y \uparrow$			$\hat{z}_y \downarrow$		
	Best	Average	Worst	Best	Average	Worst
cVAE*	0.84	0.82	0.81	0.78	0.80	0.81
MM-cVAE*	0.85	0.82	*nan*	0.72	0.76	*nan*
double InfoGAN	**0.95**	**0.95**	**0.94**	**0.69**	**0.71**	**0.73**

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

*: Results are different from [Weinberger et al., 2022] where no external test set is used.

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 $\hat{s}_y = Q_s(\mathbf{s}|\mathbf{y})$ and $\hat{z}_y = Q_z(\mathbf{z}|\mathbf{y})$. 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 ± 0.03 for background images and 2.66 ± 0.02 for target images, whereas MM-cVAE obtains 1.43 ± 0.03 and 1.44 ± 0.01 for background and target images respectively. Similar results were obtained using the Fréchet inception distance (FID).

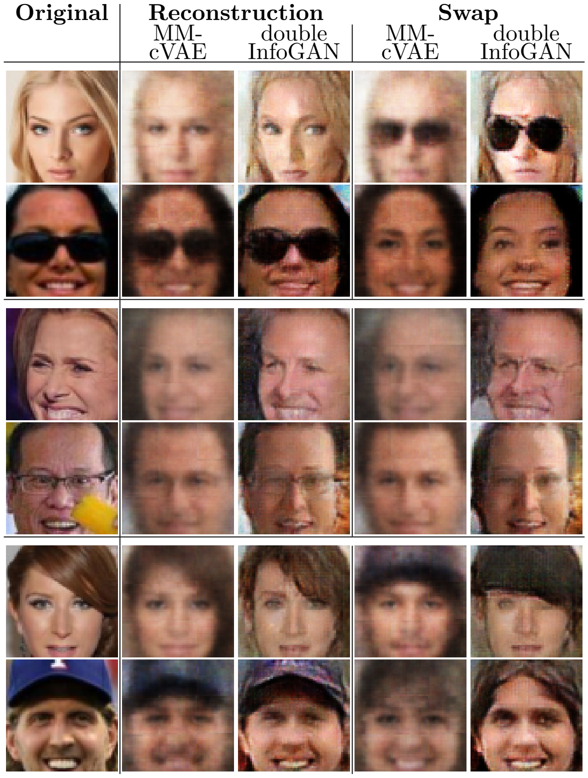

Figure 3: Image reconstruction and swap with the CelebA with accessories dataset.

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 $\mathbf{z}$ in each row and using different $\mathbf{s}$ (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

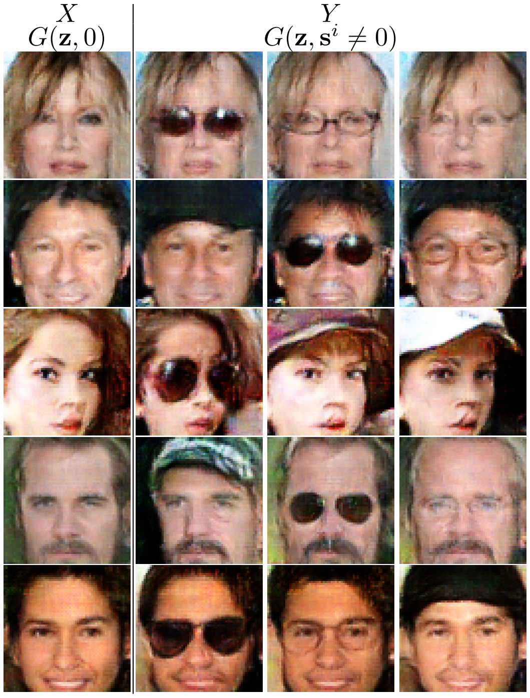

Figure 4: Fake images generated by our model. In each row, we use the same common feature $\mathbf{z}$ for all images, $\mathbf{s} = 0$ for X and different salient features $\mathbf{s} \neq 0$ for Y.

latent space $\mathbf{z}_y$, and the MNIST variability (*i.e.,* digits) only in the salient space $\mathbf{s}_y$. A perfect classifier would have 100% accuracy on MNIST when using $\mathbf{s}_y$ and 10% (which corresponds to randomness) when using $\mathbf{z}_y$. Conversely, it should have 100% accuracy on Cifar-10 when trained on $\mathbf{z}_y$ and 10% when trained on $\mathbf{s}_y$.

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., nan*). 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-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.

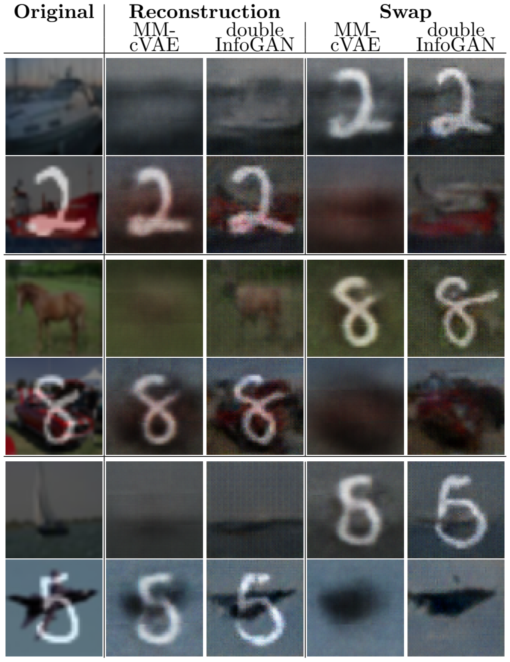

Figure 5: Image reconstruction and swap with Cifar-10-MNIST.

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 (128×128) 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 ($\mathbf{s} = 0$), and on the right images with tumor ($\mathbf{s} \neq 0$). Images in the same row are generated using the same $\mathbf{z}$. 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 $\mathbf{z}$ (i.e. row), we can also notice that the model seems to have correctly encoded in $\mathbf{z}$ the general anatomical variability of the brain.

In Fig.7, we generate healthy counterparts of target images with tumor, setting $\mathbf{s} = 0$. 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

	Mnist (salient)						Cifar (background)					
	$\mathbf{s}_y \uparrow$			$\mathbf{z}_y \downarrow$			$\mathbf{s}_y \downarrow$			$\mathbf{z}_y \uparrow$		
	Best	Avg.	Worst	Best	Avg.	Worst	Best	Avg.	Worst	Best	Avg.	Worst
MM-cVAE (size 128)	0.81	0.76	*nan*	0.43	0.48	*nan*	0.14	0.18	*nan*	0.36	0.35	*nan*
MM-cVAE (size 200)	0.82	0.63	0.13	0.43	0.58	0.82	**0.12**	**0.17**	0.27	0.37	0.36	0.34
double InfoGAN (size 128)	0.87	**0.87**	**0.86**	**0.25**	**0.26**	**0.28**	0.17	0.18	**0.19**	0.43	0.42	0.41
double InfoGAN (size 200)	**0.88**	**0.87**	**0.86**	0.32	0.32	0.32	0.20	0.21	0.23	**0.44**	**0.44**	**0.43**

Table 2: MNIST-Cifar10 classification. Digits information should only be encoded in $\mathbf{s}_y$ and not in $\mathbf{z}_y$, whereas the contrary should be true for Objects information. Std ≤ 0.01. Best results in **bold**.

	Mnist (salient)		Cifar (bg)	
	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$
$-L_{Class}$	0.48	0.83	0.23	0.37
$- L_{Class}$ - L_{Im}	0.54	0.72	0.22	0.38
$- L_{Info}$ - L_{Class} - L_{Im}	0.70	0.70	**0.18**	0.18
$- L_{Info}$	0.85	0.60	0.30	0.36
$- L_{Info}$ - L_{Im}	0.59	0.59	0.20	0.20
$- L_{Class}$ - L_{Info}	0.74	0.67	0.29	0.35
$- L_{Im}$	0.86	**0.25**	0.20	**0.42**
Full	**0.87**	0.26	**0.18**	**0.42**

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

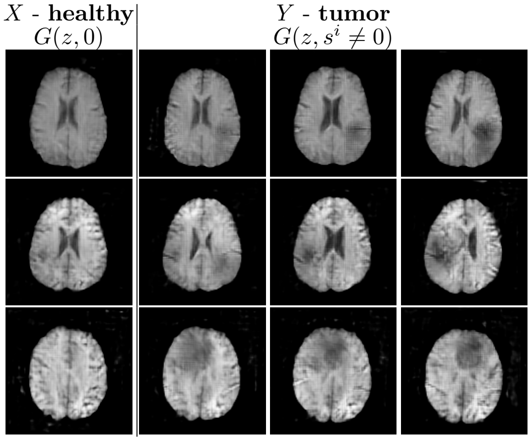

Figure 6: Fake images generated by our model. In each row, we use the same $\mathbf{z}$ for all images with $\mathbf{s} = 0$ for X and different $\mathbf{s} \neq 0$ for each exemple of Y.

[Liu et al., 2015a, Roux et al., 2019], where tumor images have to be registered to healthy templates. Please note that here we use 2D slices with a small architecture (DCGAN), and a small (and biased) dataset (Brats). Indeed, we have noticed that most of the slices containing a tumor are in the central part of the brain (greater size) whereas slices from the higher or lower part of the brain (smaller size) have less frequently a tumor. This might thus entail structural changes during the generation of the healthy counterpart (swap), such as the one in size in the third row of Fig. 7. This could be solved by directly working with 3D data, more powerful networks and debiasing strategies.

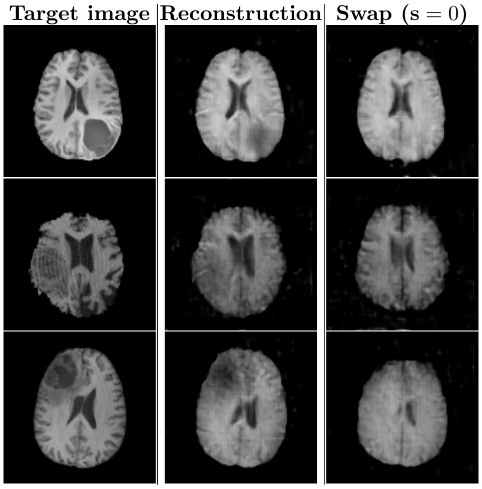

Figure 7: Reconstruction (middle) and generation of an healthy counterpart (swap, on the right) of a target image with brain tumor (on the left) by setting $\mathbf{s} = 0$ and keeping the same $\mathbf{z}$.

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, elipse 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 $\mathbf{s}_y$, while keeping $\mathbf{z}_y$ 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. This will widen its fields of application to, for instance, clinically valuable and challenging tasks, such as computer aided-diagnosis. 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[5]. 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.

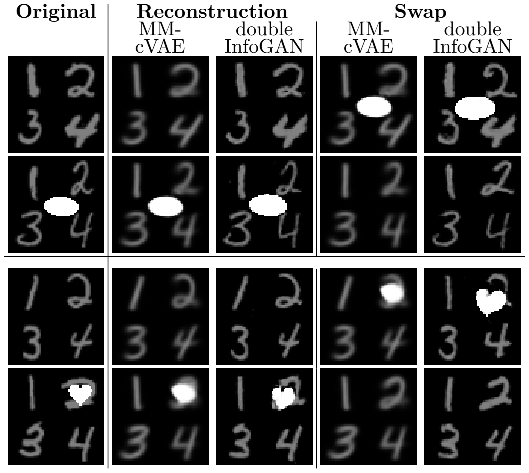

Figure 8: Image reconstruction and swap of salient features on the dSprites-MNIST dataset.

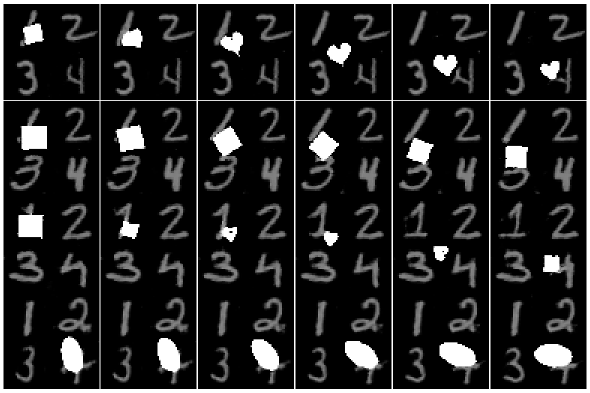

Figure 9: Each row represents the variation of only one element of the salient factor $\mathbf{s}_y$, while keeping $\mathbf{z}_y$ fixed. We can see a certain entanglement, with several parameters changing at the same time: shape and position (line 1), position and orientation (line 2). Only the last line shows a disentanglement, with only the orientation of the ellipse changing.

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

[5] The dataset label could be considered as an auxiliary variable but it does not make c and s independent

References

[Abid et al., 2018] Abid, A., Zhang, M. J., Bagaria, V. K., and Zou, J. (2018). Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nat Commun*, 9(1):2134. Number: 1 Publisher: Nature Publishing Group.

[Abid and Zou, 2019] Abid, A. and Zou, J. (2019). Contrastive Variational Autoencoder Enhances Salient Features. arXiv:1902.04601 [cs, stat].

[Barbano et al., 2023] Barbano, C. A., Dufumier, B., Tartaglione, E., Grangetto, M., and Gori, P. (2023). Unbiased Supervised Contrastive Learning. In *ICLR*. arXiv:2211.05568 [cs, stat].

[Baur et al., 2021] Baur, C., Denner, S., Wiestler, B., Navab, N., and Albarqouni, S. (2021). Autoencoders for unsupervised anomaly segmentation in brain MR images: A comparative study. *Medical Image Analysis*, 69(8):1–16.

[Benaim et al., 2019] Benaim, S., Khaitov, M., Galanti, T., and Wolf, L. (2019). Domain intersection and domain difference. *Proceedings of the IEEE International Conference on Computer Vision*, pages 3444–3452.

[Brock et al., 2019] Brock, A., Donahue, J., and Simonyan, K. (2019). Large scale GaN training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*.

[Chen et al., 2016] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

[Choi et al., 2018] Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S., and Choo, J. (2018). StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 8789–8797.

[Choudhuri et al., 2019] Choudhuri, A., Makkuva, A. V., Rana, R., Oh, S., Chowdhary, G., and Schwing, A. (2019). Towards Principled Objectives for Contrastive Disentanglement.

[Deun et al., 2012] Deun, K. V., Mechelen, I. V., Thorrez, L., Schouteden, M., Moor, B. D., Werf, M. J. v. d., Lathauwer, L. D., Smilde, A. K., and Kiers, H. A. L. (2012). DISCO-SCA and Properly Applied GSVD as Swinging Methods to Find Common and Distinctive Processes. *PLOS ONE*, 7(5):e37840.

[Feng et al., 2018] Feng, Q., Jiang, M., Hannig, J., and Marron, J. S. (2018). Angle-based joint and individual variation explained. *Journal of Multivariate Analysis*, 166:241–265.

[François et al., 2022] François, A., Maillard, M., Oppenheim, C., Pallud, J., Bloch, I., Gori, P., and Glaunès, J. (2022). Weighted Metamorphosis for Registration of Images with Different Topologies. In Hering, A., Schnabel, J., Zhang, M., Ferrante, E., Heinrich, M., and Rueckert, D., editors, *Biomedical Image Registration*, Lecture Notes in Computer Science, pages 8–17, Cham. Springer International Publishing.

[Ganin et al., 2017] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2017). Domain-Adversarial Training of Neural Networks. In Csurka, G., editor, *Domain Adaptation in Computer Vision Applications*, Advances in Computer Vision and Pattern Recognition, pages 189–209. Springer International Publishing, Cham.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 27.

[Guillon et al., 2021] Guillon, L., Cagna, B., Dufumier, B., Chavas, J., Rivière, D., and Mangin, J.-F. (2021). Detection of Abnormal Folding Patterns with Unsupervised Deep Generative Models. In Abdulkadir, A., Kia, S. M., Habes, M., Kumar, V., Rondina, J. M., Tax, C., and Wolfers, T., editors, *Machine Learning in Clinical Neuroimaging*, Lecture Notes in Computer Science, pages 63–72, Cham. Springer International Publishing.

[He et al., 2019] He, Z., Zuo, W., Kan, M., Shan, S., and Chen, X. (2019). AttGAN: Facial Attribute Editing by only Changing What You Want. *IEEE Transactions on Image Processing*, 28(11):5464–5478.

[Higgins et al., 2017] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). β-VAE: learning basic visual concepts with a constrained variational framework. In *ICLR*.

[Hoffman et al., 2018] Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T. (2018). CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1989–1998. PMLR. ISSN: 2640-3498.

[Huang et al., 2018] Huang, X., Liu, M.-Y., Belongie, S., and Kautz, J. (2018). Multimodal Unsupervised Image-to-Image Translation. In *ECCV*.

[Hyvarinen et al., 2019] Hyvarinen, A., Sasaki, H., and Turner, R. E. (2019). Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning. In *AISTATS*.

[Isola et al., 2017] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, Honolulu, HI. IEEE.

[Joy et al., 2021] Joy, T., Schmon, S. M., Torr, P. H. S., Siddharth, N., and Rainforth, T. (2021). Capturing Label Characteristics in VAEs. In *ICLR*. arXiv:2006.10102 [cs, stat].

[Karras et al., 2019] Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:4396–4405.

[Kazemi et al., 2019] Kazemi, H., Iranmanesh, S. M., and Nasrabadi, N. (2019). Style and Content Disentanglement in Generative Adversarial Networks. pages 848–856.

[Khemakhem et al., 2020] Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A. (2020). Variational Autoencoders and Nonlinear ICA: A Unifying Framework. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR. ISSN: 2640-3498.

[Kim and Mnih, 2018] Kim, H. and Mnih, A. (2018). Disentangling by Factorising. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2649–2658. PMLR. ISSN: 2640-3498.

[Kimura et al., 2020] Kimura, D., Chaudhury, S., Narita, M., Munawar, A., and Tachibana, R. (2020). Adversarial Discriminative Attention for Robust Anomaly Detection. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2161–2170, Snowmass Village, CO, USA. IEEE.

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. In *ICLR*.

[Kivva et al., 2022] Kivva, B., Rajendran, G., Ravikumar, P., and Aragam, B. (2022). Identifiability of deep generative models without auxiliary information. In *NeurIPS*.

[Krizhevsky, 2009] Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. Technical report.

[Lample et al., 2017] Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., and Ranzato, M. (2017). Fader networks: Manipulating images by sliding attributes. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):5968–5977.

[LeCun,] LeCun, Y. The mnist database of handwritten digits. Technical report.

[Lee et al., 2018] Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. (2018). Diverse Image-to-Image Translation via Disentangled Representations. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, volume 11205, pages 36–52. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.

[Lin et al., 2020a] Lin, Z., Thekumparampi, K. K., Fanti, G., and Oh, S. (2020a). InfoGAN-CR and modelcentrality: Self-supervised model training and selection for disentangling gans. *37th International Conference on Machine Learning, ICML 2020*, pages 6083–6095.

[Lin et al., 2020b] Lin, Z., Thekumparampil, K., Fanti, G., and Oh, S. (2020b). InfoGAN-CR and ModelCentrality: Self-supervised Model Training and Selection for Disentangling GANs. In *Proceedings of the 37th International Conference on Machine Learning*.

[Liu et al., 2017] Liu, M.-Y., Breuel, T., and Kautz, J. (2017). Unsupervised Image-to-Image Translation Networks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Liu et al., 2015a] Liu, X., Niethammer, M., Kwitt, R., Singh, N., McCormick, M., and Aylward, S. (2015a). Low-Rank Atlas Image Analyses in the Presence of Pathologies. *IEEE Transactions on Medical Imaging*, 34(12):2583–2591. Conference Name: IEEE Transactions on Medical Imaging.

[Liu et al., 2015b] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015b). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

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

Bachem, O. (2019). Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4114–4124. PMLR. ISSN: 2640-3498.

[Locatello et al., 2020] Locatello, F., Poole, B., Raetsch, G., Schölkopf, B., Bachem, O., and Tschannen, M. (2020). Weakly-Supervised Disentanglement Without Compromises. In *Proceedings of the 37th International Conference on Machine Learning*, pages 6348–6359. PMLR. ISSN: 2640-3498.

[Ma et al., 2019] Ma, L., Jia, X., Georgoulis, S., Tuytelaars, T., and Van Gool, L. (2019). Exemplar Guided Unsupervised Image-to-Image Translation with Semantic Consistency. In *ICLR*. arXiv. arXiv:1805.11145 [cs].

[Maillard et al., 2022] Maillard, M., François, A., Glaunès, J., Bloch, I., and Gori, P. (2022). A Deep Residual Learning Implementation of Metamorphosis. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pages 1–4. ISSN: 1945-8452.

[Matthey et al., 2017] Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. (2017). dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/.

[Menze et al., 2014] Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al. (2014). The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024.

[PANG et al., 2022] PANG, G., SHEN, C., CAO, L., and HENGEL, A. V. D. (2022). Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54(2):1–38.

[Radford et al., 2016] Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pages 1–16.

[Rohlf and Corti, 2000] Rohlf, F. J. and Corti, M. (2000). Use of Two-Block Partial Least-Squares to Study Covariation in Shape. *Systematic Biology*, 49(4):740–753. Publisher: [Oxford University Press, Society of Systematic Biologists].

[Rombach et al., 2022] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*, pages 10674–10685.

[Roux et al., 2019] Roux, A., Roca, P., Edjlali, M., Sato, K., Zanello, M., Dezamis, E., Gori, P., Lion, S., Fleury, A., Dhermain, F., Meder, J.-F., Chrétien, F., Lechapt, E., Varlet, P., Oppenheim, C., and Pallud, J. (2019). MRI Atlas of IDH Wild-Type Supratentorial Glioblastoma: Probabilistic Maps of Phenotype, Management, and Outcomes. *Radiology*, 293(3):633–643. Publisher: Radiological Society of North America.

[Ruiz et al., 2019] Ruiz, A., Martinez, O., Binefa, X., and Verbeek, J. (2019). Learning Disentangled Representations with Reference-Based Variational Autoencoders. In *ICLR workshop on Learning from Limited Labeled Data*, pages 1–17, New Orleans, United States.

[Severson et al., 2019] Severson, K., Ghosh, S., and Ng, K. (2019). Unsupervised learning with contrastive latent variable models. In *AAAI*. arXiv. arXiv:1811.06094 [cs, stat].

[Shi et al., 2021] Shi, Y., Yang, X., Wan, Y., and Shen, X. (2021). SemanticStyleGAN: Learning Compositional Generative Priors for Controllable Image Synthesis and Editing.

[Shu et al., 2020] Shu, R., Chen, Y., Kumar, A., Ermon, S., and Poole, B. (2020). Weakly Supervised Disentanglement with Guarantees. In *ICLR*. arXiv. arXiv:1910.09772 [cs, stat].

[Smilde et al., 2017] Smilde, A. K., Måge, I., Næs, T., Hankemeier, T., Lips, M. A., Kiers, H. A. L., Acar, E., and Bro, R. (2017). Common and distinct components in data fusion. *Journal of Chemometrics*, 31(7):e2900.

[Song et al., 2021] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-Based Generative Modeling through Stochastic Differential Equations. In *ICLR*.

[Trygg, 2002] Trygg, J. (2002). O2-PLS for qualitative and quantitative analysis in multivariate calibration. *Journal of Chemometrics*, 16(6):283–293.

[Tu et al., 2021] Tu, R., Foss, A. H., and Zhao, S. D. (2021). Capturing patterns of variation unique to a specific dataset. arXiv:2104.08157 [cs, stat].

[Venkataramanan et al., 2020] Venkataramanan, S., Peng, K.-C., Singh, R. V., and Mahalanobis, A. (2020). Attention Guided Anomaly Localization in Images. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, volume 12362, pages 485–503. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.

[von Kügelgen et al., 2021] von Kügelgen, J., Sharma, Y., Gresele, L., Brendel, W., Schölkopf, B., Besserve, M., and Locatello, F. (2021). Self-Supervised Learning with Data Augmentations Provably Isolates Content from Style. In *Advances in Neural Information Processing Systems*, volume 34, pages 16451–16467.

[Vowels et al., 2020] Vowels, M. J., Cihan Camgoz, N., and Bowden, R. (2020). NestedVAE: Isolating Common Factors via Weak Supervision. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9199–9209, Seattle, WA, USA. IEEE.

[Vétil et al., 2022] Vétil, R., Abi-Nader, C., Bône, A., Vullierme, M.-P., Rohé, M.-M., Gori, P., and Bloch, I. (2022). Learning Shape Distributions from Large Databases of Healthy Organs: Applications to Zero-Shot and Few-Shot Abnormal Pancreas Detection. In Wang, L., Dou, Q., Fletcher, P. T., Speidel, S., and Li, S., editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, Lecture Notes in Computer Science, pages 464–473, Cham. Springer Nature Switzerland.

[Weinberger et al., 2022] Weinberger, E., Beebe-Wang, N., and Lee, S.-I. (2022). Moment Matching Deep Contrastive Latent Variable Models. In *AISTATS*. arXiv. arXiv:2202.10560 [cs].

[Wyner, 1975] Wyner, A. (1975). The common information of two dependent random variables. *IEEE Trans. Inform. Theory*, 21(2):163–179.

[Yu et al., 2017] Yu, Q., Risk, B. B., Zhang, K., and Marron, J. S. (2017). JIVE integration of imaging and behavioral data. *NeuroImage*, 152:38–49.

[Zhu et al., 2017] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, Venice. IEEE.

[Zou et al., 2013] Zou, J. Y., Hsu, D. J., Parkes, D. C., and Adams, R. P. (2013). Contrastive Learning Using Spectral Methods. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

[Zou et al., 2022] Zou, K., Faisan, S., Heitz, F., and Valette, S. (2022). Joint Disentanglement of Labels and Their Features with VAE. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1341–1345, Bordeaux, France. IEEE.

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]

Supplementary Materials

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, anonimized 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.

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

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

Discriminator D / C / Q	Generator G
Input $128 \times 128 \times 1$	Input $z \in \mathbb{R}^n, s \in \mathbb{R}^m$
Gaussian noise, 4×4 conv 64, stride 2, spectral norm, lReLU	concat + FC 8192 +reshape $(512, 4, 4)$
Gaussian noise, 4×4 conv 128, stride 2, spectral norm, lReLU	upsample, 3×3 conv 1024, batchnorm, ReLU
Gaussian noise, 4×4 conv 256, stride 2, spectral norm, lReLU	upsample, 3×3 conv 512, batchnorm, ReLU
Gaussian noise, 4×4 conv 512, stride 2, spectral norm, lReLU	upsample, 3×3 conv 256, batchnorm, ReLU
Gaussian noise, 4×4 conv 512, stride 2, spectral norm, lReLU (*)	Self-Attention Block
From *: Gaussian noise, FC 1, sigmoid (output layer for D)	upsample, 3×3 conv 256, batchnorm, ReLU
From *: Gaussian noise, FC 1, sigmoid (output layer for C)	3×3 conv 128, batchnorm, ReLU
From *: Gaussian noise, FC 128 , spectral norm, lReLU	upsample, 3×3 conv 64, batchnorm, ReLU
Gaussian noise, FC n , spectral norm, lReLU (output layer for Q_z)	3×3 conv 1, tanh
From *: Gaussian noise , FC 128 , SN, lReLU	
Gaussian noise, FC m , spectral norm, lReLU (output layer for 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:

$$\begin{aligned}
&I(\mathbf{c};\mathbf{x}) = -H(\mathbf{c}|\mathbf{x}) + H(\mathbf{c}) = \mathbb{E}_{\mathbf{x}\sim P(\mathbf{x}),\mathbf{c}\sim P(\mathbf{c}|\mathbf{x})} \log P(\mathbf{c}|\mathbf{x}) + H(\mathbf{c}) = \\
&= \int P(\mathbf{x}) \int P(\mathbf{c}|\mathbf{x}) \log(P(\mathbf{c}|\mathbf{x})) dxdc + H(\mathbf{c}) = \\
&= \int P(\mathbf{x}) \int P(\mathbf{c}'|\mathbf{x}) \log(P(\mathbf{c}'|\mathbf{x})) dxdc' + H(\mathbf{c}') = \quad \text{(change of variables between c and c')} \\
&= \int\int\int P(\mathbf{x},\mathbf{c},\mathbf{z}) dcdz \int P(\mathbf{c}'|\mathbf{x}) \log(P(\mathbf{c}'|\mathbf{x})) dxdc' + H(\mathbf{c}') = \\
&= \int P(\mathbf{z}) \int P(\mathbf{c}) \int P(\mathbf{x}|\mathbf{c},\mathbf{z}) \int P(\mathbf{c}'|\mathbf{x}) \log(P(\mathbf{c}'|\mathbf{x})) dzdxdc'dc + H(\mathbf{c}') = \\
&= \int P(\mathbf{z}) \int P(\mathbf{c}) \int P(\mathbf{x}|\mathbf{c},\mathbf{z}) \int P(\mathbf{c}'|\mathbf{x}) \log(P(\mathbf{c}'|\mathbf{x}) \frac{Q(\mathbf{c}'|\mathbf{x})}{Q(\mathbf{c}'|\mathbf{x})}) dzdxdc'dc + H(\mathbf{c}') = \quad \text{(identity trick)} \\
&= \underbrace{\mathbb{E}_{\mathbf{z}\sim P(\mathbf{z}),\mathbf{c}\sim P(\mathbf{c}),\mathbf{x}\sim P(\mathbf{x}|\mathbf{c},\mathbf{z})} KL(P(\mathbf{c}'|\mathbf{x})||Q(\mathbf{c}'|\mathbf{x}))}_{\geq 0} + \mathbb{E}_{\mathbf{z}\sim P(\mathbf{z}),\mathbf{c}\sim P(\mathbf{c}),\mathbf{x}\sim P(\mathbf{x}|\mathbf{c},\mathbf{z})} \mathbb{E}_{c'\sim P(\mathbf{c}'|\mathbf{x})} \log Q(\mathbf{c}'|\mathbf{x}) + H(\mathbf{c}') \\
&\geq \mathbb{E}_{\mathbf{z}\sim P(\mathbf{z}),\mathbf{c}\sim P(\mathbf{c}),\mathbf{x}\sim P(\mathbf{x}|\mathbf{c},\mathbf{z})} \mathbb{E}_{\mathbf{c}'\sim P(\mathbf{c}'|\mathbf{x})} \log(Q(\mathbf{c}'|\mathbf{x})) + H(\mathbf{c}')
\end{aligned} \tag{9}$$

where we have introduced an auxiliary distribution $Q(\mathbf{c}|\mathbf{x})$, parameterized as a neural network, to approximate the posterior $P(\mathbf{c}|\mathbf{x})$ (which is difficult to compute) and we have made the hypothesis that $\mathbf{c}$ does not depend on $\mathbf{z}$ (i.e., $P(\mathbf{c}|\mathbf{z}) = P(\mathbf{c})$). To further remove also the need to sample from $P(\mathbf{c}|\mathbf{x})$ (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: $\mathcal{L}(G, Q) = \mathbb{E}_{\mathbf{z}\sim P(\mathbf{z}),\mathbf{c}\sim P(\mathbf{c}),x\sim P(\mathbf{x}|\mathbf{c},\mathbf{z})} \log(Q(\mathbf{c}|\mathbf{x})) + H(\mathbf{c})$, which is equivalent to the previous lower bound:

$$\begin{aligned}
&\mathcal{L}(G,Q) = \mathbb{E}_{\mathbf{z}\sim P(\mathbf{z}),\mathbf{c}\sim P(\mathbf{c}),\mathbf{x}\sim P(\mathbf{x}|\mathbf{c},\mathbf{z})} \log(Q(\mathbf{c}|\mathbf{x})) + H(\mathbf{c}) = \\
&= \int P(\mathbf{z}) \int P(\mathbf{c}) \int P(\mathbf{x}|\mathbf{c},\mathbf{z}) \log(Q(\mathbf{c}|\mathbf{x})) dzdcdx + H(\mathbf{c}) = \\
&= \int P(\mathbf{z}) \int\int P(\mathbf{x},\mathbf{c}|\mathbf{z}) \log(Q(\mathbf{c}|\mathbf{x})) dzdcdx + H(\mathbf{c}) = \\
&= \int P(\mathbf{z}) \int\int P(\mathbf{x},\mathbf{c}'|\mathbf{z}) \log(Q(\mathbf{c}'|\mathbf{x})) dzdc'dx + H(\mathbf{c}') = \quad \text{(change of variable between c and c')} \\
&= \int P(\mathbf{z}) \int\int P(\mathbf{x}|\mathbf{z}) P(\mathbf{c}'|\mathbf{x},\mathbf{z}) \log(Q(\mathbf{c}'|\mathbf{x})) dzdc'dx + H(\mathbf{c}') = \\
&= \int P(\mathbf{z}) \int\int\int P(\mathbf{x},\mathbf{c}|\mathbf{z}) P(\mathbf{c}'|\mathbf{x}) \log(Q(\mathbf{c}'|\mathbf{x})) dzdc'dxdc + H(\mathbf{c}') = \quad \text{(c' doesn't depend on z, re-introduce c)} \\
&= \int P(\mathbf{z}) \int\int\int P(\mathbf{c}) P(\mathbf{x}|\mathbf{c},\mathbf{z}) P(\mathbf{c}'|\mathbf{x}) \log(Q(\mathbf{c}'|\mathbf{x})) dzdc'dxdc + H(\mathbf{c}') = \\
&= \mathbb{E}_{\mathbf{z}\sim P(\mathbf{z}),\mathbf{c}\sim P(\mathbf{c}),\mathbf{x}\sim P(\mathbf{x}|\mathbf{c},\mathbf{z})} \mathbb{E}_{\mathbf{c}'\sim P(\mathbf{c}'|\mathbf{x})} \log(Q(\mathbf{c}'|\mathbf{x})) + H(\mathbf{c}')
\end{aligned} \tag{10}$$

In [Chen et al., 2016], authors proposed to model the auxiliary conditional distribution $Q(\mathbf{c}|\mathbf{x})$ as a *factorized Gaussian* with *identity covariance* $Q(\mathbf{c}|\mathbf{x}) = \prod_i (c_i|\mathbf{x}) = \prod_i \mathcal{N}(\mu_i(\mathbf{x}), 1)$. 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:

$$\mathcal{L}_c = d_{JS}(Q^1, ..., Q^k) := \frac{1}{k} \sum_i KL(Q^i || \frac{\sum_j Q^j}{k}) \tag{11}$$

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 $\mathcal{L}_c$ 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.

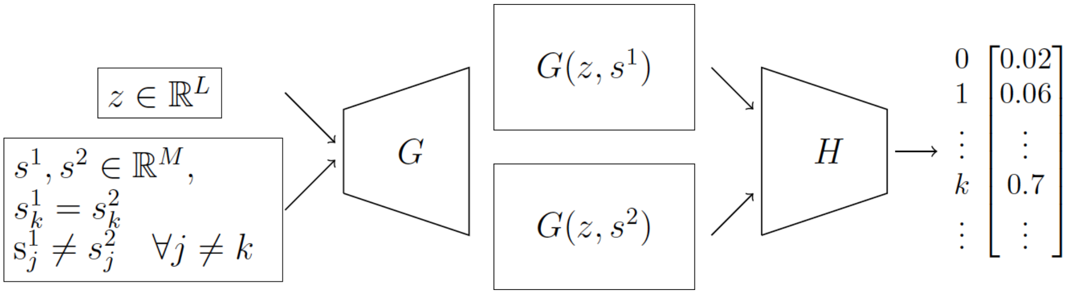
$z \in \mathbb{R}^L$
$s^1, s^2 \in \mathbb{R}^M,$
$s^1_k = s^2_k$
$\mathrm{s}^1_j \neq s^2_j \quad \forall j \neq k$
G
$G(z, s^1)$
$G(z, s^2)$
H
$\begin{array}{c c} 0 & 0.02 \\ 1 & 0.06 \\ \vdots & \vdots \\ k & 0.7 \\ \vdots & \vdots \end{array}$

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.

C.2 Double InfoGAN

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

$$
\begin{aligned}
& I((\mathbf{z},\mathbf{s});\mathbf{y}) = -H((\mathbf{z},\mathbf{s})|\mathbf{y}) + H(\mathbf{z},\mathbf{s}) = -H((\mathbf{z},\mathbf{s})|\mathbf{y}) + H(\mathbf{z}) + H(\mathbf{s}) = \\
& = \int\int\int P(\mathbf{z},\mathbf{s},\mathbf{y}) \log(\frac{P(\mathbf{z},\mathbf{s},\mathbf{y})}{P(\mathbf{y})}) dzdsdy + H(\mathbf{z}) + H(\mathbf{s}) = \\
& = \int P(\mathbf{y}) \int\int P(\mathbf{s}|\mathbf{y})P(\mathbf{z}|\mathbf{y}) \log(P(\mathbf{s}|\mathbf{y})P(\mathbf{z}|\mathbf{y})) dzdsdy + H(\mathbf{z}) + H(\mathbf{s}) = \quad \text{(suppose } P(\mathbf{s},\mathbf{z}|\mathbf{y}) = P(\mathbf{s}|\mathbf{y})P(\mathbf{z}|\mathbf{y})) \\
& = \int P(\mathbf{y}) \int\int P(\mathbf{s}|\mathbf{y})P(\mathbf{z}|\mathbf{y}) \left(\log(P(\mathbf{s}|\mathbf{y})) + \log(P(\mathbf{z}|\mathbf{y}))\right) dzdsdy + H(\mathbf{z}) + H(\mathbf{s}) = \\
& = \int_y P(\mathbf{y}) \left(\int_s P(\mathbf{s}|\mathbf{y}) \log(P(\mathbf{s}|\mathbf{y})) + \int_z P(\mathbf{z}|\mathbf{y}) \log(P(\mathbf{z}|\mathbf{y})) \right) dzdsdy + H(\mathbf{z}) + H(\mathbf{s}) = \\
& = \mathbb{E}_{\mathbf{y}\sim P(\mathbf{y}),\mathbf{s}\sim P(\mathbf{s}|\mathbf{y})} \log(P(\mathbf{s}|\mathbf{y}) + \mathbb{E}_{\mathbf{y}\sim P(\mathbf{y}),\mathbf{z}\sim P(\mathbf{z}|\mathbf{y})} \log(P(\mathbf{z}|\mathbf{y}) + H(\mathbf{z}) + H(\mathbf{s}) = \\
& = -H(\mathbf{s}|\mathbf{y}) - H(\mathbf{z}|\mathbf{y}) + H(\mathbf{z}) + H(\mathbf{s}) = I(\mathbf{z};\mathbf{y}) + I(\mathbf{s};\mathbf{y})
\end{aligned}
\tag{12}
$$

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

$$
\begin{aligned}
& \log P(\mathbf{y}) = \log \int\int\int P(\mathbf{y},\mathbf{z},\mathbf{s},\mathbf{y}_R) dzdsdy_R = \log \int\int\int P(\mathbf{y}|\mathbf{z},\mathbf{s},\mathbf{y}_R) P(\mathbf{z},\mathbf{s}|\mathbf{y}_R) P(\mathbf{y}_R) \frac{Q(\mathbf{z},\mathbf{s}|\mathbf{y}_R)}{Q(\mathbf{z},\mathbf{s}|\mathbf{y}_R)} dzdsdy_R \\
& = \log \mathbb{E}_{\mathbf{y}_R\sim P(\mathbf{y}_R),(\mathbf{z},\mathbf{s})\sim Q(\mathbf{z},\mathbf{s}|\mathbf{y}_R)} P(\mathbf{y}|\mathbf{z},\mathbf{s},\mathbf{y}_R) \frac{P(\mathbf{z},\mathbf{s}|\mathbf{y}_R)}{Q(\mathbf{z},\mathbf{s}|\mathbf{y}_R)} \geq \mathbb{E}_{\mathbf{y}_R\sim P(\mathbf{y}_R),(\mathbf{z},\mathbf{s})\sim Q(\mathbf{z},\mathbf{s}|\mathbf{y}_R)} \log P(\mathbf{y}|\mathbf{z},\mathbf{s},\mathbf{y}_R) \frac{P(\mathbf{z},\mathbf{s}|\mathbf{y}_R)}{Q(\mathbf{z},\mathbf{s}|\mathbf{y}_R)} = \\
& = \mathbb{E}_{\mathbf{y}_R\sim P(\mathbf{y}_R),(\mathbf{z},\mathbf{s})\sim Q(\mathbf{z},\mathbf{s}|\mathbf{y}_R)} \log P(\mathbf{y}|\mathbf{z},\mathbf{s},\mathbf{y}_R) - \mathbb{E}_{\mathbf{y}_R\sim P(\mathbf{y}_R)} KL(Q(\mathbf{z},\mathbf{s}|\mathbf{y}_R)||P(\mathbf{z},\mathbf{s}|\mathbf{y}_R))
\end{aligned}
\tag{13}
$$

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 :

$$
\begin{aligned}
\underset{G,Q,H,C}{min}\ \underset{D}{max} \quad & w_{Adv} \cdot \mathcal{L}_{Adv}(G,D) + w_{Class} \cdot \mathcal{L}_{Class}(G,C) \\
& + w_{Info} \cdot \mathcal{L}_{Info}(G,Q) + w_{Im} \cdot \mathcal{L}_{Im}(G,Q) \\
& + w_{CR} \cdot \mathcal{L}_{CR}(G,H)
\end{aligned}
\tag{14}
$$

D.1 Adversarial GAN Loss

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

$$
\begin{aligned}
\mathcal{L}_{Adv}(D,G) = & w_{bg}\Big(-\mathbb{E}_{\mathbf{x}_R\sim P(\mathbf{x}_R)}\big[\log(D(\mathbf{x}_R)\big] - \mathbb{E}_{z\sim P_x(\mathbf{z})}\big[\log(1-(D(G(\mathbf{z},0))))\big]\Big) \\
& w_t\Big(-\mathbb{E}_{\mathbf{y}_R\sim P(\mathbf{y}_R)}\big[\log(D(\mathbf{y}_R)\big] - \mathbb{E}_{\mathbf{z},\mathbf{s}\sim P_y(\mathbf{z},\mathbf{s})}\big[\log(1-(D(G(\mathbf{z},\mathbf{s}))))\big]\Big)
\end{aligned}
\tag{15}
$$

D.2 Info Loss

The weight w_{Info} is actually divided in three components : w^s_{Info}, w^z_{Info} and w^{real}_{Info}

$$
\begin{aligned}
\mathcal{L}_{Info}(G,Q) = & w_{bg}\mathbb{E}_{\mathbf{z}\sim P_y(\mathbf{z})}\big[w^z_{Info}|(Q_z(G(z,0)) - z| + w^s_{Info}|Q_s(G(z,0)) - 0|\big] \\
& + w_t\mathbb{E}_{\mathbf{z},\mathbf{s}\sim P_y(\mathbf{z},\mathbf{s})}\big[w^z_{Info}|(Q_z(G(z,s)) - z| + w^s_{Info}|Q_s(G(z,s)) - s|\big] \\
& + w^{real}_{Info}\mathbb{E}_{\mathbf{x}_R\sim P(\mathbf{x}_R)}\big[|(Q_s(\mathbf{x}_R)) - 0|\big]
\end{aligned}
\tag{16}
$$

parameters	definition	value for different datasets			
		celeba	cifar-mnist	mnist-dsprite	brats
loop G	number of G loop	1	1	2	1
loop D	number of D loop	1	1	1	1
loop CR (when used)	number of CR loop	-	1	1	1
lr G	learning rate G	0.0002	0.0002	$5 \cdot 10^{-5}$	0.0001
lr D	learning rate D	0.0002	0.0002	$5 \cdot 10^{-5}$	0.0001
lr CR (when used)	learning rate CR	-	0.0002	$5 \cdot 10^{-5}$	0.0001
w_{bg}	weight for background	0.5	0.5	0.5	0.5
w_t	weight for target losses	1.0	1.0	1.0	1.0
w_{Adv}	weight for adversarial loss	0.5	0.5	0.5	0.5
w_{Class}	weight for class classification	0.5	0.5	0.5	0.5
w_{Image}	weight for image reconstruction	1.0	1.0	1.0	1.0
w_{Info}^{z}	weight for info loss z	1.0	1.0	1.0	1.0
w_{Info}^{s}	weight for info loss s	1.0	1.0	1.0	1.0
w_{Info}^{real}	weight for info loss real image	1.0	1.0	1.0	1.0
w_{CR} (when used)	weight for CR loss	-	1.0	1.0	1.0

Table 6: Hyperparameters used for every dataset

D.3 Image reconstruction loss

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

$$\mathcal{L}_{Im}(G,Q) = w_{bg}\mathbb{E}_{\mathbf{x}_R\sim P(\mathbf{x}_R),\hat{z}=Q_z(\mathbf{x}_R)}\Big[|G(\hat{z},0)-\mathbf{x}_R|\Big] + w_t\mathbb{E}_{\mathbf{y}_R\sim P(\mathbf{y}_R),\hat{z},\hat{s}=Q(\mathbf{y}_R)}\Big[|G(\hat{z},\hat{s})-\mathbf{y}_R|\Big] \tag{17}$$

D.4 CR Loss

The Contrastive Regularization loss is computed to improve the disentanglement of $\mathbf{s}$. 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 $\forall j \neq k$):

$$\mathcal{L}_{CR}(G,H) = \mathbb{E}_{k\in\mathbb{N},z\sim P_y(\mathbf{z}),s^1,s^2\sim P_y(\mathbf{s}),s^1_k=s^2_k,s^1_j\neq s^2_j,\forall j\neq k}\Big[CE\big(H(G(z,s^1),G(z,s^2)),k\big)\Big] \tag{18}$$

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

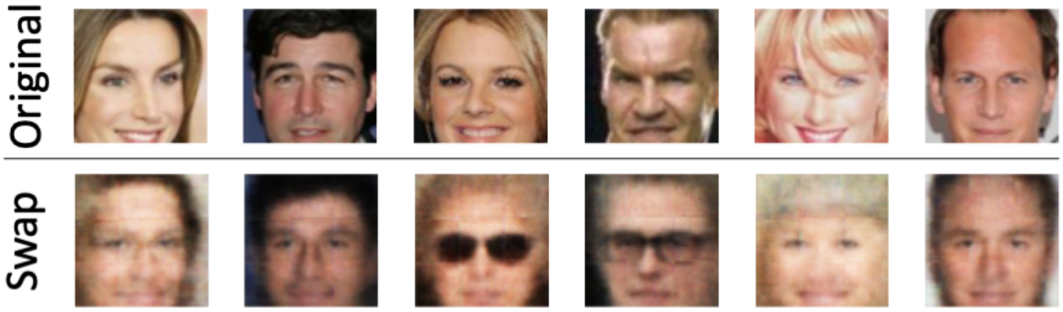

Figure 13: Swap using the cVAE method with the CelebA with accessories dataset. We can clearly see that personal traits (which should be encoded in the common space) are lost during the swap and accessories (which should be encoded in the salient space) are not always correctly added. This could explain the poor quantitative performance of cVAE.

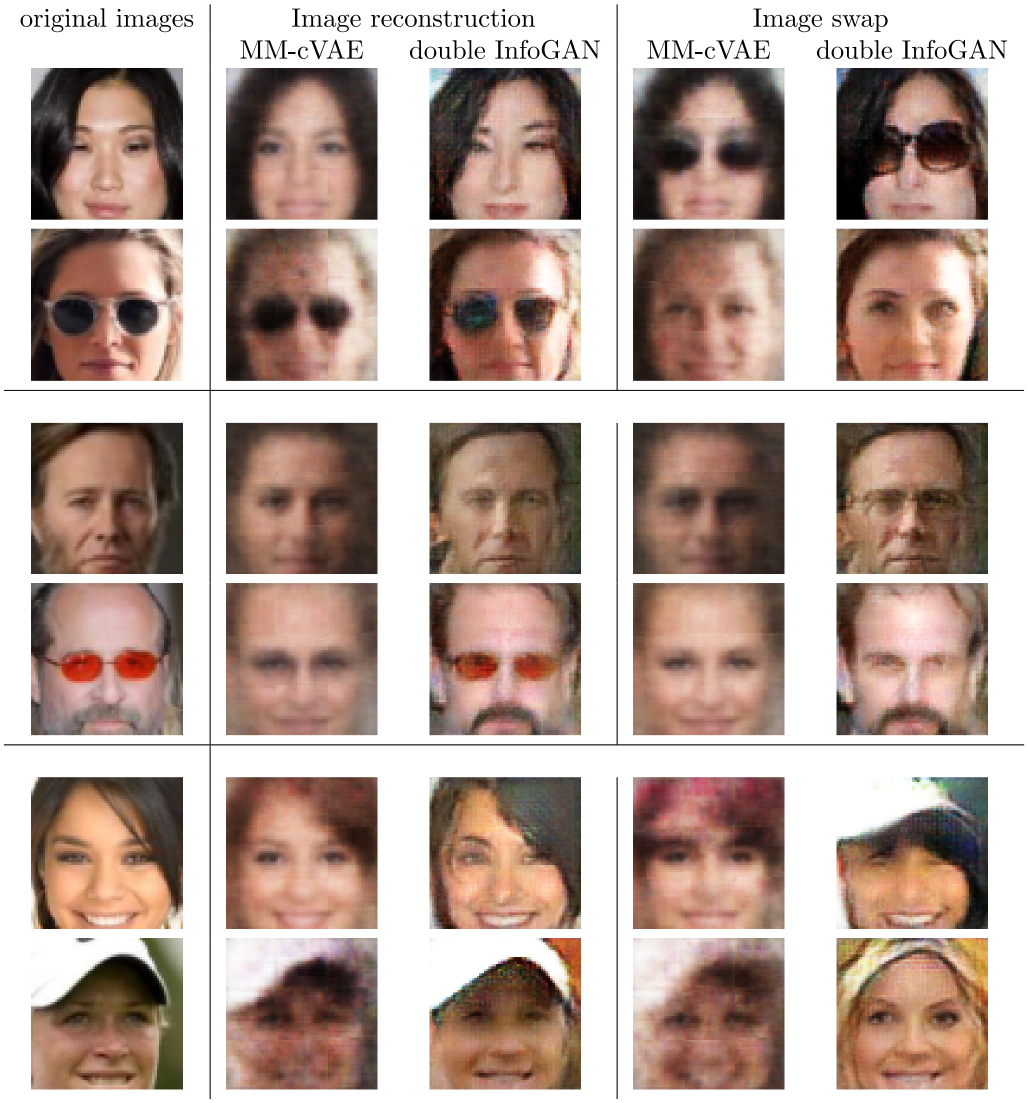

Figure 11: Image reconstruction and swap with CelebA. In every block, first row refers to X and second row to Y. It's interesting to notice that rare attributes, such as the glasses in the first two blocks, are correctly reconstructed by our method and not by MM-cVAE. However, they are changed towards more "common" glasses after swapping.

	cVAE									
	epoch 100		epoch 200		epoch 300		epoch 400		epoch 500	
	$s_x \uparrow$	$z_x \downarrow$	$s_x \uparrow$	$z_x \downarrow$	$s_x \uparrow$	$z_x \downarrow$	$s_x \uparrow$	$z_x \downarrow$	$s_x \uparrow$	$z_x \downarrow$
Training 1	0.81	0.80	0.82	0.80	0.83	0.80	0.82	0.79	0.83	0.80
Training 2	0.78	0.82	0.80	0.82	0.79	0.82	0.80	0.81	0.81	0.81
Training 3	0.79	0.82	0.79	0.82	0.80	0.81	0.81	0.81	0.81	0.81
Training 4	0.83	0.79	0.84	0.78	0.83	0.78	0.84	0.78	**0.84**	**0.78**
Training 5	0.79	0.82	0.80	0.81	0.82	0.81	0.81	0.80	0.82	0.80
	MM-cVAE									
	epoch 100		epoch 200		epoch 300		epoch 400		epoch 500	
	$s_x \uparrow$	$z_x \downarrow$	$s_x \uparrow$	$z_x \downarrow$	$s_x \uparrow$	$z_x \downarrow$	$s_x \uparrow$	$z_x \downarrow$	$s_x \uparrow$	$z_x \downarrow$
Training 1	0.84	0.75	0.84	0.75	0.85	0.73	0.85	0.73	**0.85**	**0.72**
Training 2	0.83	0.77	0.84	0.76	0.85	0.75	0.85	0.75	0.85	0.74
Training 3	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*
Training 4	0.79	0.82	0.80	0.83	0.73	0.83	0.74	0.83	0.74	0.83
Training 5	0.81	0.79	0.82	0.79	0.82	0.79	0.83	0.78	0.83	0.77
	double InfoGAN									
	epoch 100		epoch 200		epoch 300		epoch 400		epoch 500	
	$s_x \uparrow$	$z_x \downarrow$	$s_x \uparrow$	$z_x \downarrow$	$s_x \uparrow$	$z_x \downarrow$	$s_x \uparrow$	$z_x \downarrow$	$s_x \uparrow$	$z_x \downarrow$
Training 1	0.93	0.74	0.93	0.73	0.95	0.72	0.95	0.73	0.94	0.70
Training 2	0.92	0.70	0.94	0.73	0.95	0.73	0.95	0.74	0.95	0.73
Training 3	0.92	0.69	0.94	0.71	0.95	0.74	0.95	0.73	**0.95**	**0.69**
Training 4	0.93	0.72	0.94	0.70	0.95	0.72	0.95	0.71	0.95	0.72
Training 5	0.93	0.72	0.94	0.70	0.95	0.72	0.95	0.76	0.95	0.73

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.

	MM-cVAE with latent space size 128 (64 × 2)											
	epoch 100				epoch 300				epoch 500			
	Mnist (salient)		Cifar (bg)		Mnist (salient)		Cifar (bg)		Mnist (salient)		Cifar (bg)	
	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$
Training 1	0.68	0.57	0.16	0.36	0.68	0.51	0.15	0.36	0.70	0.49	0.16	0.36
Training 2	0.80	0.51	0.15	0.36	0.80	0.41	0.16	0.36	**0.81**	**0.43**	**0.14**	**0.36**
Training 3	0.81	0.56	0.27	0.33	0.81	0.52	0.27	0.33	0.81	0.52	0.27	0.33
Training 4	0.81	0.53	0.31	0.25	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*
Training 5	0.71	0.57	0.16	0.35	0.73	0.51	0.15	0.36	0.74	0.47	0.16	0.36
	MM-cVAE with latent space size 200 (100 × 2)											
	epoch 100				epoch 300				epoch 500			
	Mnist (salient)		Cifar (bg)		Mnist (salient)		Cifar (bg)		Mnist (salient)		Cifar (bg)	
	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$
Training 1	0.19	0.82	0.12	0.35	0.14	0.82	0.12	0.36	0.13	0.82	0.12	0.35
Training 2	0.80	0.55	0.17	0.36	0.80	0.44	0.16	0.37	0.80	0.43	0.16	0.37
Training 3	0.81	0.64	0.28	0.34	0.82	0.57	0.27	0.34	0.82	0.55	0.27	0.34
Training 4	0.66	0.67	0.18	0.36	0.64	0.63	0.16	0.36	0.61	0.60	0.15	0.36
Training 5	0.76	0.60	0.19	0.36	0.78	0.51	0.17	0.37	0.79	0.48	0.17	0.37
	double InfoGAN with latent space size 200 (100 × 2)											
	epoch 100				epoch 300				epoch 500			
	Mnist (salient)		Cifar (bg)		Mnist (salient)		Cifar (bg)		Mnist (salient)		Cifar (bg)	
	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$
Training 1	0.89	0.28	0.19	0.42	0.89	0.30	0.20	0.44	0.88	0.32	0.20	0.44
Training 2	0.89	0.29	0.21	0.41	0.88	0.31	0.22	0.43	0.87	0.32	0.22	0.44
Training 3	0.90	0.28	0.19	0.42	0.88	0.32	0.21	0.44	0.88	0.32	0.21	0.44
Training 4	0.90	0.29	0.20	0.41	0.88	0.30	0.22	0.43	0.86	0.32	0.23	0.43
Training 5	0.89	0.28	0.19	0.41	0.88	0.30	0.21	0.44	0.86	0.32	0.22	0.43
	double InfoGAN with latent space size 128 (64 × 2)											
	epoch 100				epoch 300				epoch 500			
	Mnist (salient)		Cifar (bg)		Mnist (salient)		Cifar (bg)		Mnist (salient)		Cifar (bg)	
	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$
Training 1	0.88	0.21	0.16	0.40	0.88	0.27	0.18	0.42	**0.87**	**0.26**	**0.18**	**0.43**
Training 2	0.87	0.24	0.16	0.39	0.88	0.27	0.16	0.43	0.87	0.26	0.17	0.43
Training 3	0.88	0.22	0.16	0.39	0.87	0.26	0.18	0.42	0.86	0.26	0.19	0.42
Training 4	0.87	0.25	0.17	0.40	0.88	0.25	0.19	0.42	0.87	0.28	0.19	0.41
Training 5	0.88	0.24	0.15	0.39	0.87	0.24	0.17	0.42	0.86	0.25	0.19	0.43
	double InfoGAN with latent space size 128 (64 × 2) with CR module											
	epoch 100				epoch 300				epoch 500			
	Mnist (salient)		Cifar (bg)		Mnist (salient)		Cifar (bg)		Mnist (salient)		Cifar (bg)	
	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$	$s_y \uparrow$	$z_y \downarrow$	$s_y \downarrow$	$z_y \uparrow$
Training 1	0.88	0.25	0.18	0.42	0.87	0.28	0.18	0.43	0.87	0.26	0.18	0.43
Training 2	0.87	0.23	0.16	0.40	0.88	0.24	0.18	0.43	0.87	0.26	0.18	0.42
Training 3	0.88	0.21	0.18	0.40	0.88	0.26	0.19	0.43	0.87	0.28	0.20	0.43
Training 4	0.88	0.26	0.17	0.39	0.88	0.27	0.18	0.42	0.88	0.26	0.19	0.42
Training 5	0.89	0.26	0.17	0.40	0.88	0.25	0.18	0.42	**0.87**	**0.26**	**0.18**	**0.43**

Table 8: Target Dataset separation on Cifar-10-MNIST for MM-cVAE and our method, with and without CR module, and for different latent space sizes. We report the results of the 5 trainings per method at different epochs. For clarity, we don't report the standard deviations, whose values are between 0.00 and 0.01. Even here, the trainings of MMc-VAE are less stable than the ones of our method, regardless of the latent dimension.

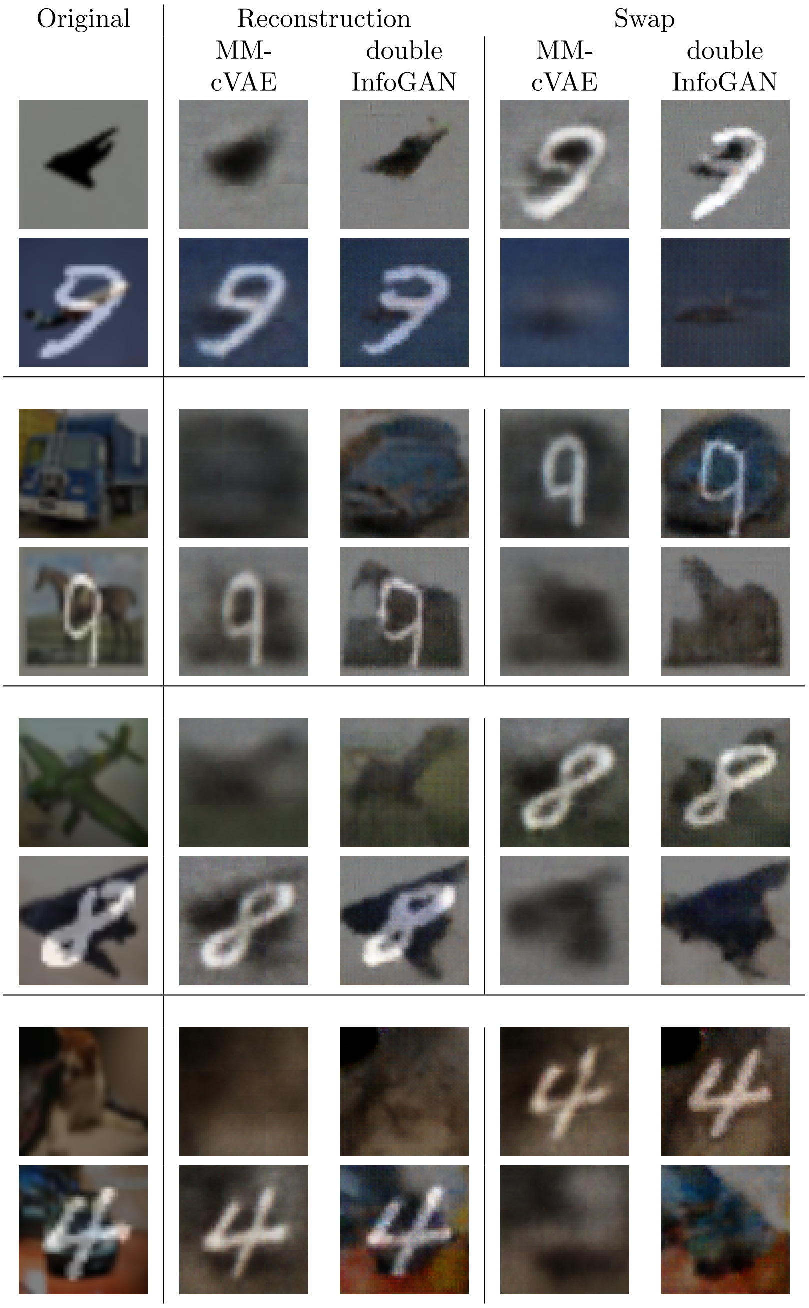

Figure 12: More examples of image reconstruction and swap with Cifar-10-MNIST dataset.

	With CR Loss							
	epoch 50	epoch 100	epoch 200	epoch 300	epoch 400	epoch 500	epoch 600	epoch 700
Training 1	0.334	0.366	0.366	0.376	0.394	0.396	0.398	0.4
Training 2	0.414	0.37	0.338	0.328	0.314	0.314	0.282	0.294
Training 3	0.258	0.294	0.25	0.258	0.222	0.222	0.218	0.216
Training 4	0.35	0.346	0.346	0.402	0.43	0.454	0.49	**0.494**
Training 5	0.202	0.202	0.206	0.216	0.22	0.222	0.226	0.24
	No CR Loss							
Training 1	0.348	0.358	0.364	0.376	0.39	0.392	0.406	**0.424**
Training 2	0.348	0.354	0.372	0.37	0.378	0.396	0.398	0.398
Training 3	0.326	0.35	0.354	0.342	0.334	0.324	0.306	0.306
Training 4	0.292	0.292	0.288	0.306	0.316	0.358	0.348	0.386
Training 5	0.306	0.32	0.318	0.292	0.304	0.324	0.334	0.34

Table 9: Ablation study of CR Loss on Mnist-dsprite dataset. Score indicated is fvae score