COUNTERFACTUALS UNCOVER THE MODULAR STRUCTURE OF DEEP GENERATIVE MODELS

Michel Besserve[1,2,*], **Arash Mehrjou**[1,3], **Rémy Sun**[1,3], **Bernhard Schölkopf**[1]
1. MPI for Intelligent Systems, Tübingen, Germany.
2. MPI for Biological Cybernetics, Tübingen, Germany.
3. Department of Computer Science, ETH Zürich, Switzerland.
4. ENS Rennes, France.

ABSTRACT

Deep generative models can emulate the perceptual properties of complex image datasets, providing a latent representation of the data. However, manipulating such representation to perform meaningful and controllable transformations in the data space remains challenging without some form of supervision. While previous work has focused on exploiting statistical independence to *disentangle* latent factors, we argue that such requirement can be advantageously relaxed and propose instead a non-statistical framework that relies on identifying a modular organization of the network, based on counterfactual manipulations. Our experiments support that modularity between groups of channels is achieved to a certain degree on a variety of generative models. This allowed the design of targeted interventions on complex image datasets, opening the way to applications such as computationally efficient style transfer and the automated assessment of robustness to contextual changes in pattern recognition systems.

Figure 1: Counterfactual manipulation of samples of a BigGan trained on ImageNet (see section 4.2).

1 INTRODUCTION

Deep generative models, by learning a non-linear function mapping a latent space to a space of observations, have proven successful at designing realistic images in a variety of complex domains (objects, animals, human faces, interior scenes). In particular, two kinds of approaches emerged as state-of-the-art (SOTA): Generative Adversarial Networks (GAN) (Goodfellow et al., 2014), and Variational Autoencoders (VAE) (Kingma & Welling, 2013; Rezende et al., 2014).

Efforts have been made to have such models produce *disentangled* latent representations that can control interpretable properties of images (Kulkarni et al., 2015; Higgins et al., 2017). However, the resulting models are not necessarily *mechanistic* (or *causal*) in the sense that interpretable properties of an image cannot be ascribed to a particular part, a *module*, of the network architecture. Gaining access to a modular organization of generative models would benefit the *interpretability* and allow *extrapolations*, such as generating an object in a background that was not previously associated with this object, as illustrated in a preview of our experimental results in Fig. 1.

Such extrapolations are an integral part of human representational capabilities (consider common expressions such as "like an elephant in a china shop") and consistent with the modular organization

*michel.besserve@tuebingen.mpg.de

of its visual system, comprising specialized regions encoding objects, faces and places (see e.g. Grill-Spector & Malach (2004)). Extrapolations moreover likely support adaptability to environmental changes and robust decision making (Dvornik et al., 2018). How to leverage trained deep generative architectures to perform such extrapolations is an open problem, largely due to the non-linearities and high dimensionality that prevent interpretability of computations performed in successive layers.

In this paper, we propose a causal framework to explore modularity, which relates to the causal principle of *Independent Mechanisms*, stating that the causal mechanisms contributing to the overall generating process do not influence nor inform each other (Peters et al., 2017).[1] We study the effect of direct interventions in the network from the point of view that the mechanisms involved in generating data can be modified individually without affecting each other. This principle can be applied to generative models to assess how well they capture a causal mechanism (Besserve et al., 2018a). Causality allows to assay how an outcome would have changed, had some variables taken different values, referred to as a *c*ounterfactual (Pearl, 2009; Imbens & Rubin, 2015). We use counterfactuals to assess the role of specific internal variables in the overall functioning of trained deep generative models, along with a rigorous definition of disentanglement in a causal framework. Then, we analyze this disentanglement in implemented models based on unsupervised counterfactual manipulations. We show empirically how VAEs and GANs trained on image databases exhibit modularity of their hidden units, encoding different features and allowing counterfactual editing of generated images.

Related work. Our work relates to the interpretability of convolutional neural networks, which has been intensively investigated in discriminative architectures (Zeiler & Fergus, 2014; Dosovitskiy & Brox, 2016; Fong & Vedaldi, 2017; Zhang et al., 2017b;a). Generative models require a different approach, as the downstream effect of changes in intermediate representations are high dimensional. InfoGANs. β-VAEs and other works (Chen et al., 2016; Mathieu et al., 2016; Kulkarni et al., 2015; Higgins et al., 2017) address supervised or unsupervised disentanglement of latent variables related to what we formalize as *extrinsic disentanglement* of transformations acting on data points. We introduce the novel concept of *intrinsic disentanglement* to uncover the internal organization of networks, arguing that many interesting transformations are statistically dependent and are thus unlikely to be disentangled in the latent space. This relates to Bau et al. (2018) who proposed a framework based on interventions on internal variables of a GAN which, in contrast to our fully unsupervised approach, requires semantic information. Higgins et al. (2018) suggest a definition of disentanglement based on group representation theory. Compared to this proposal, our approach (introduced independently in (Besserve et al., 2018b)) is more flexible as it applies to arbitrary continuous transformations, free from the strong requirements of representation theory (see Appendix F). Finally, an interventional approach to disentanglement has also be taken by Suter et al. (2018), who focuses on extrinsic disentanglement in a classical graphical model setting and develop measures of interventional robustness based on labeled data.

2 From disentanglement to counterfactuals and back

We introduce a general framework to formulate precisely the notion of disentanglement and bridge it to causal concepts. This theory section will be presented informally to ease high level understanding. Readers interested in the mathematical aspects can refer to Appendix A where we provide all details.

2.1 A Causal Generative Model (CGM) framework

We consider a generative model M that implements a function g_M, which maps a latent space $\mathcal{Z}$ to a manifold $\mathcal{Y}_M$ where the learned data points live, embedded in ambient Euclidean space $\mathcal{Y}$ (Fig. 2a). A sample from the model is generated by drawing a realization z from a prior latent variable distribution with mutually independent components, fully supported in $\mathcal{Z}$. We will use the term *representation* to designate a mapping r from $\mathcal{Y}_M$ to some representation space $\mathcal{R}$ (we also call $r(y)$ the representation of a point $y \in \mathcal{Y}_M$). In particular, we will assume (see Def. 7 and Prop. 4, in Appendix A) that g_M is (left)-invertible, such that g_M^{-1} is a representation of the data, called the *latent representation*.

Assuming the generative model is implemented by a non-recurrent neural network, we can use a causal graphical model representation of the entailed computational graph implementing the mapping g_M through a succession of operations (called *functional assignments* in causal language), as illustrated

[1]Note that this is not a *statistical* independence; the quantities transformed by the mechanisms of course do influence each other and can be statistically dependent.

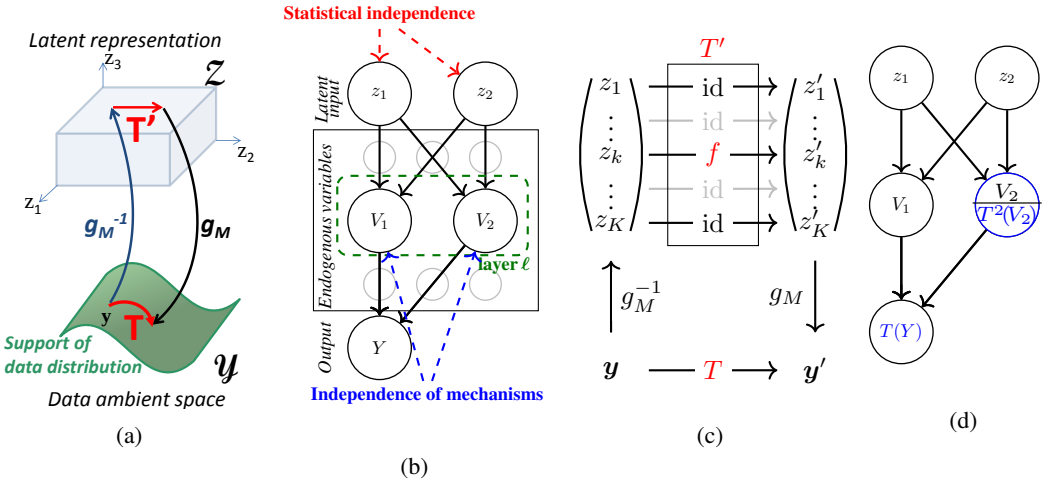


Figure 2: (a) Illustration of the generative mapping and a disentangled transformation. (b) Causal graph of an example CGM showing different types of independence between nodes. (c) Commutative diagram showing sparse transformation T' in latent space associated to a disentangled transformation T. (d) Illustration of intrinsic disentanglement with $\mathcal{E} = \{2\}$.

in Fig. 2b, that we will call Causal Generative Model (CGM). In addition to the latent representation, we can then choose a collection of possibly multi-dimensional endogenous (internal) variables represented by nodes in the causal graph, such that the mapping g_M is computed by composing the *endogenous variable assignment* v_M with the *endogenous mapping* $\tilde{g}_M$ according to the diagram

$$\mathcal{Z} \overset{v_M}{\to} \boldsymbol{\mathcal{V}}_M \overset{\tilde{g}_M}{\to} \mathcal{Y}_M \,.$$

A paradigmatic choice for these variables is the collection of output activation maps of each channel in one hidden layer of a convolutional neural network, as illustrated in Fig. 2b. As for the latent case, we use mild conditions that guarantee $\tilde{g}_M$ to be left-invertible, defining the *internal representation* of the network (see Def. 7 and Prop. 4 in Appendix A). Given the typical choice of dimensions for latent and endogenous variables, the V_k's are also constrained to take values in subsets $\mathcal{V}_M^k$ of smaller dimension than their Euclidean ambient space $\mathcal{V}^k$. As detailed in Appendix A, we introduce the *endogenous image* sets of the form $\boldsymbol{\mathcal{V}}_M^{\mathcal{E}} = \left\{\boldsymbol{v} \in \prod_{k\in\mathcal{E}} \mathcal{V}^k \colon \boldsymbol{v} = (v_k(\boldsymbol{z}))_{k\in\mathcal{E}},\, \boldsymbol{z} \in \mathcal{Z}\right\}$ for a subset of variables indexed by $\mathcal{E}$ (amounting to $\boldsymbol{\mathcal{V}}_M$ when $\mathcal{E}$ includes all endogenous variables).

This CGM framework allows defining counterfactuals in the network following Pearl (2014).

Definition 1 (Unit level counterfactual, informal)**.** *Given CGM M, the* interventional model $M_{\boldsymbol{h}}$ *is obtained by replacing assignments of the subset variables $\mathcal{E}$ and by the vector of assignments $\boldsymbol{h}$. Then for any latent input $\boldsymbol{z}$, called* unit*, the* unit-level counterfactual *is the output of $M_{\boldsymbol{h}}$:*

$$Y_{\boldsymbol{h}}^{\mathcal{E}} = g_{M_{\boldsymbol{h}}}(\boldsymbol{z})\,.$$

Def. 1 is also in line with the concept of *potential outcome* (Imbens & Rubin, 2015). Importantly, counterfactuals induce a transformation of the output of the generative model.

Definition 2 (Counterfactual mapping)**.** *Given an embedded CGM, we call the transformation*

$$\overset{\curvearrowright}{Y}_{\boldsymbol{h}}^{\mathcal{E}} : y \mapsto g_{M_{\boldsymbol{h}}}\left(g_M^{-1}(y)\right)$$

the $\boldsymbol{h}$-counterfactual mapping. We say it is faithful *to M whenever $\overset{\curvearrowright}{Y}_{\boldsymbol{h}}^{\mathcal{E}}[\mathcal{Y}_M] \subset \mathcal{Y}_M$.*

We introduce faithfulness of a counterfactual mapping to account for the fact that not all interventions on internal variables will result in an output that could have been generated by the original model. In the context of generative model, non-faithful counterfactuals generate examples that leave the support of the distribution learned from data, possibly resulting in an artifactual output (assigning a large value to a neuron may saturate downstream neurons), or allowing extrapolation to unseen data.

2.2 UNSUPERVISED DISENTANGLEMENT: FROM STATISTICAL TO CAUSAL PRINCIPLES

The classical notion of disentangled representation (e.g. Bengio et al. (2013); Kulkarni et al. (2015)), posits individual latent variables "*sparsely encode real-world transformations*". Although the concept of *real-world transformations* remains elusive, this insight, agnostic to statistical concepts, has driven supervised approaches to disentangling representations, where relevant transformations are well-identified and manipulated explicitly using appropriate datasets and training procedures.

In contrast, unsupervised learning approaches to disentanglement need to learn such *real-world transformations* from unlabeled data. In order to address this challenge, SOTA approaches seek to encode such transformations by changes in individual latent factors, and resort to a statistical notion of disentanglement, enforcing conditional independence between latent factors (Higgins et al., 2017). This statistical approach leads to several issues:

- **Independence is not necessary.** The i.i.d. constraints on the latent variables' prior impose statistical independence between disentangled factors for the data distribution. This is unlikely for many relevant properties, counfounded by factors of the true data generating mechanisms (e.g. skin and hair color).
- **Independence is not sufficient** to specify a disentangled representation, such that the problem remains ill-posed (Locatello et al., 2018). As a consequence, finding an appropriate inductive bias to learn a representation that benefits downstream tasks remains an open question.
- To date, SOTA unsupervised approaches are mostly demonstrated on synthetic datasets, and beyond MNIST **disentangling complex real world data has been limited** to the well-calibrated CelebA dataset. On complex real-world datasets, **disentangled generative models exhibit visual sample quality far below non-disentangled SOTA** (e.g. BigGAN exploited in our work (Brock et al., 2018)).

We propose an non-statistical definition of disentanglement by first phrasing mathematically the transformation-based insights (Bengio et al., 2013; Kulkarni et al., 2015). Consider a transformation T acting on the data manifold $\mathcal{Y}_M$. As illustrated by the commutative diagram of Fig. 2c, disentanglement of such property then amounts to having T correspond to a transformation T' of the latent space that would act only on a single variable z_k, using transformation f, leaving the other latent variables available to encode other properties. More explicitly we have

$$T'(\boldsymbol{z}) : (z_1, .., z_k, .., z_K) \mapsto (z_1, .., f(z_k), .., z_K)\,, \quad \text{and} \quad T(g_M(\boldsymbol{z})) = g_M(T'(\boldsymbol{z}))\,.$$

It is then natural to qualify two transformations T_1 and T_2 as disentangled (from each other), whenever they modify different components of the latent representation (see Def. 10 in Appendix F). This amounts to saying that the transformations follow the causal principle of independent mechanisms (Peters et al., 2017; Parascandolo et al., 2018).

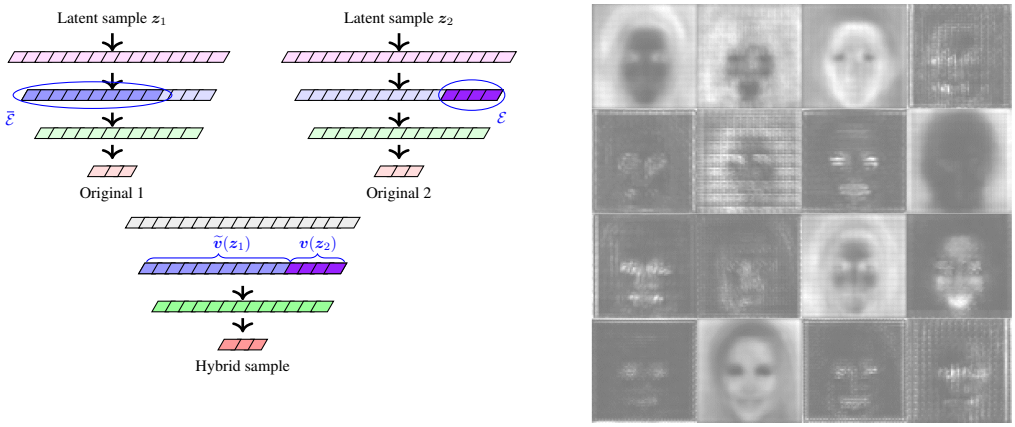
Due to the fact that it relies on transformations of the latent representation, that are exogenous to the CGM, we call this notion *extrinsic disentanglement*. This "functional" definition has the benefit of being agnostic the the subjective choice of the property to disentangle, and to the statistical notion of independence. However, we can readily notice that, if applied to the latent space (where components are i.i.d. distributed), this functional notion of disentangled transformation still entails statistical independence between disentangled factors. We thus need to exploit a different representation to uncover possibly statistically related properties, but disentangled in the sense of our definition.

2.3 DISENTANGLING BY MANIPULATING INTERNAL REPRESENTATIONS

As illustrated in the CGM of Fig. 2b, in contrast to latent variables, properties encoded by endogenous variables of the graphical model are not necessarily statistically independent due to common latent causes (z_1 and z_2 in the figure), but may still reflect interesting properties of the data that can be intervened on independently, following the principle of independence of mechanisms. We thus extend our definition of disentanglement to allow transformations of the internal variables of the network as follows.

Definition 3 (Intrinsic disentanglement, informal)**.** *In a CGM M, a transformation $T : \mathcal{Y}_M \to \mathcal{Y}_M$ is* intrinsically disentangled *with respect to a subset $\mathcal{E}$ of endogenous variables, if there is a transformation T' acting on the internal representation space such that for any endogenous value $\boldsymbol{v}$*

$$T(\widetilde{g}_M(\boldsymbol{v})) = \widetilde{g}_M(T'(\boldsymbol{v})) \tag{1}$$



(a) (b)

Figure 3: **Generation of influence maps.** (a) Principle of sample hybridization through counterfactuals. (b) Examples of EIM generated by a VAE on the CelebA dataset (lighter pixel indicate larger variance and thus stronger influence of the perturbations on that pixel).

where $T'(\boldsymbol{v})$ only affects the variables indexed by $\mathcal{E}$.

Fig. 2d illustrates this second notion of disentanglement, where the split node indicates that the value of V_2 is computed as in the original CGM (Fig. 2b) before applying transformation T^2 to the outcome. While the above definition applies to a single transformation, straightforward extensions of this concept to families of transformations are provided in Appendix F. Faithful counterfactuals represent examples of disentangled transformations:

Proposition 1 (Counterfactuals and disentanglement, informal)**.** *Consider an intervention on subset $\mathcal{E}$, its associated counterfactual mapping is faithful if and only if it is disentangled. For interventions on variables that remain within the support of the original marginal distribution, it is sufficient that $\mathcal{E}$ and its complement $\bar{\mathcal{E}}$ do not have common latent ancestors.*

This indicates that finding faithful counterfactuals can be used to learn disentangled transformations.

3 Finding modularity in deep generative models

3.1 Defining modularity

Building on Sec. 2, we define modularity as a structural property of the internal representation, allowing (with the immediately following Prop. 2) to implement arbitrary disentangled transformations.

Definition 4 (Modularity)**.** *A subset of endogenous variables $\mathcal{E}$ is called modular whenever $\mathcal{V}_M$ is the Cartesian product of $\boldsymbol{\mathcal{V}}_M^{\mathcal{E}}$ by $\boldsymbol{\mathcal{V}}_M^{\bar{\mathcal{E}}}$.*

Proposition 2 (Modularity implies disentanglement)**.** *If $\mathcal{E}$ is modular, then any transformation applied to it staying within its input domain is disentangled.*

The proof is a natural extension of the proof of Proposition 1. Both the Definition and the Proposition have trivial extensions to multiple modules (along the line described in Appendix F). While we have founded this framework on a functional definition of disentanglement that applies to transformations, the link made here with an structural property of the trained network allows us to define a *disentangled representation* as follows.

Definition 5 (Disentangled representation)**.** *A disentangled representation is a partition of the endogenous variables in several modules, such that $\mathcal{V}_M$ is the Cartesian product of the corresponding endogenous image sets.*

Then any transformation applied to a given module leads to a valid transformation in the data space (it is relatively disentangled following Def. 10 in Appendix F).

Interestingly, a disentangled representation is associated to a partition of the considered set of endogenous variables into modules. This extra requirement was not considered in classical approaches to disentanglement as it was assumed that each single scalar variables could be considered as an independent module. Our framework provides an insight relevant to artificial and biological systems: as the activity of multiple neurons can be strongly tied together, the concept of representation may not be meaningful at the "atomic" level of single neurons, but require to group them into modules forming a "mesoscopic" level, at which each group can be intervened on independently.

3.2 Hybridization as a disentangled transformation

As stated in Sec. 2, a functional definition of disentanglement, leaves unanswered how to find relevant transformations. Prop. 1 and 2 provide the following hints: (1) Once a modular structure is found in the network, a broad class of disentangled transformations are available, (2) Transformations that stay within their input domain are good candidates of disentanglement, (3) Counterfactual interventions implicitly define transformations. We follow these guidelines by assigning a constant value $\boldsymbol{v}_0$ to a subset of endogenous variables $\mathcal{E}$ to define counterfactuals (i.e. $\boldsymbol{h}$ is a constant function), aiming for faithful ones by constraining $\boldsymbol{v}_0$ to belong to $\boldsymbol{\mathcal{V}}_M^{\mathcal{E}}$. To avoid characterizing $\boldsymbol{\mathcal{V}}_M^{\mathcal{E}}$, we rely on sampling from the (joint) marginal distribution of the variables in $\mathcal{E}$.

To illustrate the procedure, we consider a standard feed-forward multilayer neural network and choose endogenous variables to be the collection of all output activations of channels of a given layer ℓ. Let $\mathcal{E}$ be a subset of these channels, the hybridization procedure, illustrated in Fig. 3a goes as follows. We take two independent examples of the latent variable $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, that will generate two *original* examples of the output $(\boldsymbol{y}_1, \boldsymbol{y}_2) = (g_M(\boldsymbol{z}_1), g_M(\boldsymbol{z}_2))$ (that we call *Original 1* and *Original 2*). We also memorize the tuple $\boldsymbol{v}(\boldsymbol{z}_2)$ gathering values of variables indexed by $\mathcal{E}$ when generating Original 2, and $\widetilde{\boldsymbol{v}}(\boldsymbol{z}_1)$ the tuple of values taken by all other endogenous variables on this layer, but when generating Original 1. Assuming the choice of $\mathcal{E}$ identifies a modular structure, $\widetilde{\boldsymbol{v}}(\boldsymbol{z}_1)$ and $\boldsymbol{v}(\boldsymbol{z}_2)$ would encode different aspects of their corresponding generated images, such that one can generate a *hybrid example* mixing these features by assigning the collection of output values of layer ℓ with the concatenated tuple $(\widetilde{\boldsymbol{v}}(\boldsymbol{z}_1),\ \boldsymbol{v}(\boldsymbol{z}_2))$ and feeding it to the downstream part of the generator network.

3.3 Measuring causal effects

The above counterfactual hybridization framework allows assessing how a given module $\mathcal{E}$ affects the output of the generator. For this purpose we quantify its causal effect by repetitively generating pairs $(\boldsymbol{z}_1,\ \boldsymbol{z}_2)$ from the latent space, where both vectors are sampled independently of each other. We then generate and collect hybrid outputs following the above described procedure for a batch of samples and use them to estimate an *influence map* as the mean absolute effect:

$$IM(\mathcal{E}) = \mathbb{E}_{\boldsymbol{z}_2 \sim P(\mathbf{Z})} \mathbb{E}_{\boldsymbol{z}_1 \sim P(\mathbf{Z})} \left| Y^{\mathcal{E}}_{v(\boldsymbol{z}_2)}(\boldsymbol{z}_1) - Y(\boldsymbol{z}_1) \right| \tag{2}$$

where $Y(\boldsymbol{z}_1) = g_M(\boldsymbol{z}_1)$ is the non-intervened output of the generator for latent input $\boldsymbol{z}_1$. In eq. 2, the difference inside the absolute value can be interpreted as a *unit-level causal effect* in the potential outcome framework (Imbens & Rubin, 2015), and taking the expectation is analogous to computing the *average treatment effect*. Our approach has however two specificities: (1) we take the absolute value of the unit-level causal effects, as their sign may not be consistent across units, (2) the result is averaged over all interventions corresponding to possible values of $\boldsymbol{z}_2$.

While IM has the same dimension as the output image, we then average it across color channels to get a single grayscale heat-map pixel map. We also define a scalar quantity to quantify the magnitude of the causal effect, the *individual influence* of module $\mathcal{E}$, by averaging IM across output pixels.

3.4 Unsupervised detection of modules and counterfactual images

A challenge with the hybridization approach is to select the subsets $\mathcal{E}$ to intervene on, especially with networks containing a large amount of units or channels per layer. We use a fine to coarse approach to extract such groups, that we will describe in the context of convolutional layers. First, we estimate *elementary influence maps* (EIM) associated to each individual output channel c of each convolutional layer of the network (i.e. we set $\mathcal{E} = \{c\}$ in eq. (2)). Then influence maps are grouped by similarity to define modules at a coarser scale, as we will describe in detail below.

Representative EIMs for channels of convolutional layers of a VAE trained on the CelebA face dataset (see result section) are shown in Supplementary Fig. 3b and suggest channels are functionally segregated, with for example some influencing finer face feature (eyes, mouth,...) and others affecting the background of the image or the hair. This supports the idea that individual channels can be grouped into modules that are mostly dedicated to one particular aspect of the output.

In order to achieve this grouping in an unsupervised way, we perform clustering of channels using their EIMs as feature vectors as follows. We first pre-process each influence map by: (1) performing a local averaging with a small rectangular sliding window to smooth the maps spatially, (2) thresholding the resulting maps at the 75% percentile of the distribution of values over the image to get a binary image. After flattening image dimensions, we get a (channel$\times$pixels) matrix $\mathbf{S}$ which is then fed to a Non-negative Matrix Factorization (NMF) algorithm with manually selected rank K, leading to the factorization $\mathbf{S} = \mathbf{W}\mathbf{H}$. From the two resulting factor matrices, we get the K cluster template patterns (by reshaping each rows of $\mathbf{H}$ to image dimensions), and the weights representing the contribution of each of these pattern to individual maps (encoded in $\mathbf{W}$). Each influence map is then ascribed a cluster based on which template pattern contributes to it with maximum weight. The choice of NMF is justified by its success in isolating meaningful parts of images in different components (Lee & Seung, 1999). However, we also compared our approach to the classical k-means clustering algorithm (see Supplementary Fig. 9). In order to further justify our NMF based approach, we also introduce a toy generative model.

Model 1. *Consider $\boldsymbol{Z}$ a vector of K i.i.d. uniformly distributed RVs. Assume a neural network with one hidden layers composed of m vector variables $\boldsymbol{V}_k$ such that*

$$\boldsymbol{V}_k = S(\boldsymbol{H}_k Z_k)\,,$$

with $\boldsymbol{H}_k \in \mathbb{R}^n$, $n > 1$ and S a strictly increasing activation function applied entry-wise to the components of each vector (e.g. a leaky ReLU). These endogenous variables are mapped to the output

$$\boldsymbol{Y} = \sum_{k=1}^{K} \boldsymbol{W}^k \boldsymbol{V}_k\,,$$

with matrices $\boldsymbol{W}^k \in \mathbb{R}^{m\times n}$, $m > nK$. Assume additionally the following random choice for the model parameters: (1) all coefficients of $\boldsymbol{H}_k$'s are sampled i.i.d. from an arbitrary distribution that has a density with respect to the Lebesgue measure, (2) there exists K sets of indices I_k over $[1, m]$ each containing at least one element $l_k \in I_k$ such that for all $j \neq k$, $l_k \notin I_j$, (3) For a given column of $\boldsymbol{W}^k$, coefficient in I_k are sampled i.i.d. from an arbitrary distribution that has a density with respect to the Lebesgue measure, while the remaining coefficients are set to zero.

The specific condition on the I_k's enforced in (2) encodes the assumption that there is an area in the image that is only influenced by one of the modules. For example, assuming a simple background/object module pair, it encodes that the borders of the image never belong to the object while the center of the image never belong to background. For this model, we get the following identifiability result.

Proposition 3. *For Model 1, with probability 1 we have:*
(1) The partition of the hidden layer entailed by the K vectors $\{\boldsymbol{V}_k\}$ corresponds to a disentangled representation (i.e. each vector is modular relatively to the others).
(2) Assume influence maps $IM(i,k)$ of each component i in each vector $\boldsymbol{V}_k$ are known and build the $(m \times nK)$ binary matrix $\boldsymbol{B}$ by concatenating binary column vectors $\boldsymbol{B}_{i,k} = IM(i,k) > 0$. Non-negative matrix factorization of $\boldsymbol{B}$ is unique (up to trivial transformations) and identifies the subsets of endogenous variables associated to each $\boldsymbol{V}_k$.

This justifies the use of NMF on a thresholded version of the matrix gathering all EIMs (to generate a binary matrix summarizing their significant influences on each output pixel). Moreover, the application of the sliding window is justified in order to enforce the similarity between the influence maps belonging to the same module, reflected by the condition on identical support I_k for all columns of $\boldsymbol{W}^k$ in Model 1, and favoring low-rank matrix factorization.

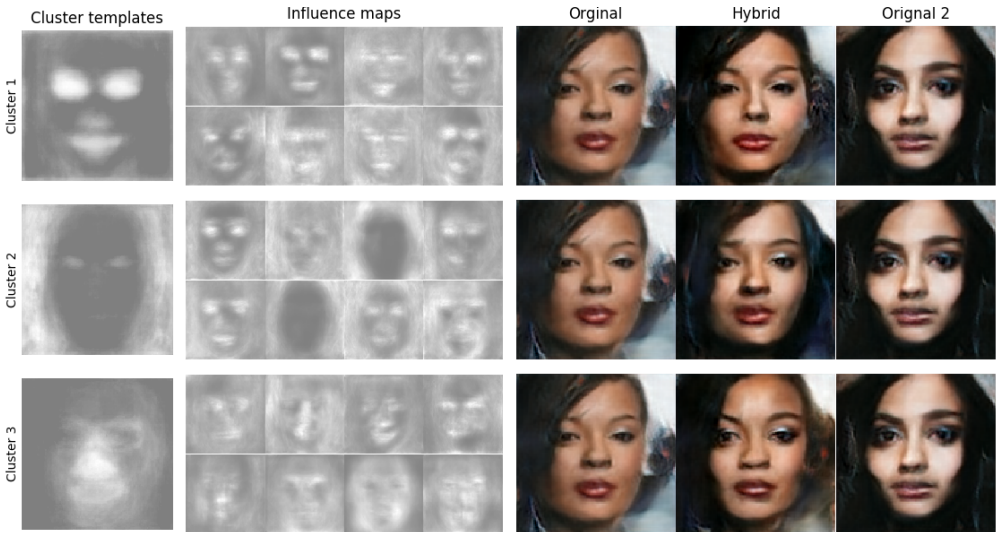


Figure 4: Left: Clustering of influence maps for a BEGAN trained on the CelebA dataset (see text). Center: example EIM of each cluster. Right: samples of the hybridization procedure using as module all channels of the intermediate layer belonging to the cluster of corresponding row. See Fig. 12 for additional samples.

4 EXPERIMENTS

4.1 DCGAN, β-VAE AND BEGAN ON THE CELEBA DATASET

We first investigated modularity of genrative models trained on the CelebFaces Attributes Dataset (CelebA)(Liu et al., 2015) and used a basic architecture: a plain β-VAE [2] (Higgins et al., 2017). We ran the full procedure described in Sec. 3, comprised of EIM calculations, clustering of channels into modules, and hybridization of generator samples using these modules. Hybridization procedures were performed by intervening on the output of the intermediate convolutional layer (indicated in Supplemental Fig. 7). The results are summarized in Supplemental Fig. 8. We observed empirically that setting the number of clusters to 3 leads consistently to highly interpretable cluster templates as illustrated in the figure, with one cluster associated to the background, one to the face and one to the hair. This observation was confirmed by running the following cluster stability analysis: we partition at random the influence maps in 3 subsets, and we use this partition to run the clustering twice on two thirds of the data, both runs overlapping only on one third. The obtained clusters were then matched in order to maximize the label consistency (the proportion of influence maps assigned the same label by both runs) on the overlapping subset, and this maximum consistency was used to assess robustness of the clustering across the number of clusters. The consistency results are provided in Supplemental Fig. 9 and show 3 clusters is a reasonable choice as consistency is large ($> 90\%$) and drops considerably for 4 clusters. Moreover, these results also show that the NMF-based clustering outperforms clustering with the more standard k-means algorithm. In addition, we also assessed the robustness of the clustering by looking at the cosine distance between the templates associated to matching clusters, averaged across clusters. The results, also provided in Supplemental Fig. 9, are consistent with the above analysis with an average cosine similarity of .9 (scalar product between the normalized feature vectors) achieved with 3 clusters (maximum similarity is 1 for perfectly identical templates). Exemplary influence maps shown in Supplemental Fig. 8 (center panel) reflect also our general observation: some maps may spread over image locations reflecting different clusters.

Interestingly, applying the hybridization procedure to the resulting 3 modules obtained by clustering leads to a replacement of the features associated to the module we intervene on, as shown in Supplemental Fig. 8 (center panel), while respecting the overall structure of the image (no discontinuity introduced). For example, on the middle row we see the facial features of the *Original 2* samples are inserted in the *Original 1* image (shown on the left), while preserving the hair.

[2] https://github.com/yzwxx/vae-celebA

While the β-VAE is designed for extrinsic disentanglement, further work has shown that it can prove suboptimal with respect to other approaches (Chen et al., 2018; Locatello et al., 2018) suggesting further work could investigate whether better extrinsic disentanglement could also favor intrinsic disentanglement. It is however important to investigate intrinsic disentanglement in models for which (extrinsic) disentanglement is not enforced explicitly. This is in particular the case of most GAN-like architectures, who typically outperform VAE-like approaches in terms of sample quality in complex image datasets. Interestingly, the above results could also be reproduced in the official tensorlayer DCGAN implementation[3], equipped with a similar architecture (see Appendix E). This suggests that our approach can be applied to models that have not been optimized for disentanglement. After these experiments with basic models, we used an implementation[4] of the Boundary Equilibrium GAN (BEGAN) (Berthelot et al., 2017), which used to set a milestone in visual quality for higher resolution face images. Most likely due to the increase in the number of layers, we observed that obtaining counterfactuals with noticeable effects required interventions on channels from the same cluster in two successive layers (see Appendix D). The results shown in Fig. 4, obtained by intervening on layers 5 and 6, reveal a clear selective transfer of features from *Original 2* to *Original 1*. As the model was trained on face images cropped with a tighter frame than for the above models, leaving little room for the hair and background, we observe only one module associated to these features (Fig. 4, middle row) showing a clear hair transfer. The remaining two modules are now encoding different aspects of face features: eye contour/mouth/nose for the top row and eyelids/face shape for the bottom row module. We further evaluated the relative quality of the counterfactual images with respect to the original generated images using the Frechet Inception Distance (FID) (Heusel et al., 2017) (Table 2 in the appendix), supporting that the hybridization procedure only mildly affects the image quality, in comparison to the original samples.

4.2 BIGGAN ON THE IMAGENET DATASET

In order to check whether our approach could scale to high resolution generative models, and generalize to complex image datasets containing a variety of objects, we used the BigGAN-deep architecture (Brock et al., 2018), pretrained[5] on the ImageNet dataset[6]. This is a conditional GAN architecture comprising 12 so-called Gblocks, each containing a cascade of 4 convolutional layers (see Appendix C for details). Each Gblock also receives direct input from the latent variables and the class label, and is bypassed by a skip connection. We then checked that we were able to generate hybrids by mixing the features of different classes. As for the case of BEGAN, intervening on two successive layers within a Gblock was more effective to generate counterfactuals (examples are provided for the 7th Gblock). Examples provided in Fig. 5 (cock-ostrich) show that it is possible to generate high quality counterfactuals with modified background while keeping a very similar object in the foreground. In a more challenging situation, with objects of different nature (Koala-teddy bear on the same figure), meaningful combinations of each original samples are still generated: e.g. a teddy bear in a tree (bottom row), or a "teddy-koala" merging teddy texture with the color of a koala on a uniform indoor background with a wooden texture (top row).

In order to investigate how the generated counterfactual images can be used to probe and improve the robustness of classifiers to contextual changes, we compared the ability of several SOTA pretrained classifier available on Tensorflow-hub (`https://tfhub.dev/`, see Appendix C for details) to recognize one of the original classes. Fig. 6 shows the average recognition rate of the most recognized original class (teddy-bear or koala), as a function of layer depth tends overall to increase. We first observe that high recognition rates are in line with the small pixel distance between hybrids and original when intervening at layers closest to the output (right panel). Interestingly, at intermediate blocks 5-6, there is a clear contrast between classifiers, with the Inception resnet performing better than the others. Interestingly, examples of non-consensual classification results in Supplementary Table 3, together with the associated hybrids (Supplementary Fig. 17) suggest different SOTA classifiers rely on different aspects of the image content to take their decision (e.g. background versus object).

[3] `https://github.com/tensorlayer/dcgan`
[4] `https://github.com/Heumi/BEGAN-tensorflow`
[5] `https://tfhub.dev/deepmind/biggan-deep-256/1`
[6] `http://www.image-net.org/`

Figure 5: Examples of BigGAN hybridizations across classes. Left: ostrich-cock, right: koala-teddy. See Figs. 13-15 for additional samples and Fig. 14-16 for entropy analysis.

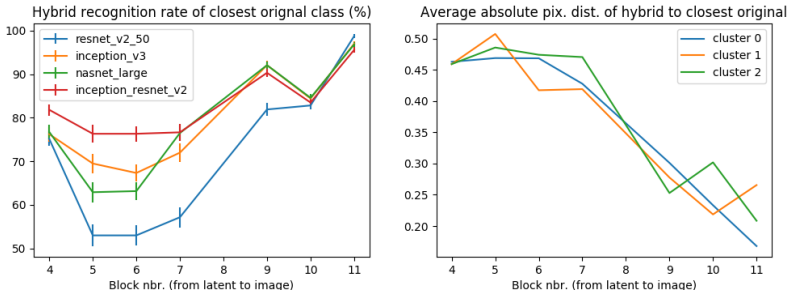


Figure 6: Analysis of classifier robustness to counterfactual changes (bars indicate standard error).

Conclusion

We introduced a mathematical definition of *disentanglement*, related it to the causal notion of counterfactual and used it for the unsupervised characterization of the representation encoded by different groups of channels in deep generative architectures. We found evidence for interpretable modules of internal variables in four different generative models trained on two complex real world datasets. Our framework opens a way to a better understanding of complex generative architectures and applications such as the style transfer (Gatys et al., 2015) of controllable properties of generated images at low computational cost (no further optimization is required), and the automated assessment of robustness of object recognition systems to contextual changes. From a broader perspective, this research direction contributes to a better exploitation of deep neural networks obtained by costly and highly energy-consuming training procedures, by (1) enhancing their interpretability and (2) allowing them to be used for tasks their where not trained for. This offers a perspective on how more sustainable research in Artificial Intelligence could be fostered in the future.

4.3 Appendices and Implementations

Appendices and supplementary Figures and Tables are available online with the arXiv version of this paper (`https://arxiv.org/abs/1812.03253`). Implementations are available on the companion website `https://gitlab.tuebingen.mpg.de/besserve/deepcounterfactuals`.

References

M. A. Armstrong. *Basic topology*. Springer Science & Business Media, 2013.

D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018.

Y Bengio, A Courville, and P Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

D. Berthelot, T. Schumm, and L. Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

M. Besserve, N. Shajarisales, B. Schölkopf, and D. Janzing. Group invariance principles for causal generative models. In *AISTATS*, 2018a.

M. Besserve, R. Sun, and B. Schölkopf. Intrinsic disentanglement: an invariance view for deep generative models. In *Workshop on Theoretical Foundations and Applications of Deep Generative Models at ICML*, 2018b.

A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 2610–2620, 2018.

X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2172–2180, 2016.

M. Diop, A. Larue, S. Miron, and D. Brie. A post-nonlinear mixture model approach to binary matrix factorization. In *25th European Signal Processing Conference (EUSIPCO)*, pp. 321–325. IEEE, 2017.

A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4829–4837, 2016.

N. Dvornik, J. Mairal, and C. Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 364–380, 2018.

R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *arXiv preprint arXiv:1704.03296*, 2017.

L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

K. Grill-Spector and R. Malach. The human visual cortex. *Annu. Rev. Neurosci.*, 27:649–677, 2004.

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

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

I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

G. W. Imbens and D. B. Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, pp. 2539–2547, 2015.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

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

F. Locatello, S. Bauer, M. Lucic, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

M. F. Mathieu, J. J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pp. 5041–5049, 2016.

G. Parascandolo, N. Kilbertus, M. Rojas-Carulla, and B. Schölkopf. Learning independent causal mechanisms. In *ICML*, pp. 4036–4044, 2018.

J. Pearl. *Causality*. Cambridge university press, second edition, 2009.

J. Pearl. The causal foundations of structural equation modeling. In Rick H. Hoyle (ed.), *Handbook of Structural Equation Modeling*, chapter 5. Guilford Press, 2014.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference – Foundations and Learning Algorithms*. MIT Press, 2017.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

M. S. M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly. Assessing generative models via precision and recall. *arXiv preprint arXiv:1806.00035*, 2018.

R. Suter, D. Miladinović, B. Schölkopf, and S. Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. *arXiv preprint arXiv:1811.00007*, 2018.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Q. Zhang, R. Cao, Y. Wu, and S. Zhu. Growing interpretable part graphs on convnets via multi-shot learning. In *AAAI*, pp. 2898–2906, 2017a.

Q. Zhang, W. Wang, and S. Zhu. Examining CNN representations with respect to dataset bias. *arXiv preprint arXiv:1710.10577*, 2017b.