LOOKING INSIDE THE BLACK BOX: ASSESSING THE MODULAR STRUCTURE OF DEEP GENERATIVE MODELS WITH COUNTERFACTUALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep generative models such as Generative Adversarial Networks (GANs) and Variational Auto-Encoders (VAEs) are important tools to capture and investigate the properties of complex empirical data. However, the complexity of their inner elements makes their functionment challenging to assess and modify. In this respect, these architectures behave as black box models. In order to better understand the function of such networks, we analyze their modularity based on the counterfactual manipulation of their internal variables. Our experiments on the generation of human faces with VAEs and GANs support that modularity between activation maps distributed over channels of generator architectures is achieved to some degree, can be used to better understand how these systems operate and edit the content of generated images.

1 INTRODUCTION

Deep generative models have proven powerful in learning to design realistic images in a variety of complex domains (handwritten digits, human faces, interior scenes). Complex neural architectures are now used to acquire a generative function mapping a latent space (such as $\mathbb{R}^n$) to a desired output space which depends on what requires generating. In particular, two very distinct approaches have recently emerged as state of the art: Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), and Variational Autoencoders (VAEs) (Kingma and Welling, 2013; Rezende et al., 2014). Generative Adversarial Networks train an image *generator* by having it fool a *discriminator* that should tell apart real from artificially generated images. Variational Autoencoders take a very different route by learning both a mapping from latent variables to the data, the *decoder*, and the converse mapping from the data to the latent variables, the *encoder*, such that correspondences between latent variables and data features can be easily investigated.

However, those complex architectures mostly act as black boxes, making it difficult to add human input to the generative process. We can act on how the network is trained, what it learns to generate, but not on *how* the learnt generative process operates. For example, if one wanted to use a network that generates faces to create a face with the eyes of one generated face and the features of another one, it would be necessary to do the merging either from the output images or through some transformation on the input. Directly influencing the generative process learnt by the network on the other hand is made difficult due to the complexity of the network function class entailed by their non-linearities and high dimensional parameter space. To grasp the properties of such a system, a possible approach would be to intervene on parts of the architecture that implements the generative function. Ideally, the effect of such interventions on the output would be interpretable. This suggests we should uncover a modular structure in those architectures, such that each part of a network can be assigned a specific function.

In this paper, we propose that modularity can be quantified and exploited in a causal framework to infer whether modules within the architecture can be further disentangled. This hypothesis relies on the general principle of Independence of Mechanisms stating that different mechanisms responsible for generating a cause can be modified and intervened on without affecting each other (Peters et al., 2017). It has been recently demonstrated that many approaches relying on this principle can be interpreted as exploiting a form of invariance of the output of a mechanism with respect

to transformations that are applied to its input, and that this principle can be applied to generative models encountered in machine learning Besserve et al. (2018).

One key aspect of causality frameworks is to study the effect of interventions on variables of a given system, and to evaluate counterfactuals that evaluate how an outcome would have changed, provided some variable would have taken different values. We use interventions and counterfactuals applied to the internal variables of trained deep generative models to assess the modular structure of these systems. We start by introducing this perspective formally with the notion of intrinsic disentanglement, and show that it extends the classical notion of disentangled representation investigated in the deep learning literature. Then, we introduce tools to analyze this disentanglement in existing systems. We show empirically how VAEs trained on the CelebA and Cifar10 dataset express a form of modularity with intermediate activation maps responsible for encoding different parts of the generated images. Finally, we assess quantitatively how different architectures and training objectives influence disentanglement.

2 Modularity as intrinsic disentanglement

In this paper we propose a framework to implement interventions inside a generative network as well as study their effect. To this end, we present here elements and formal definitions necessary to understand the suggested intervention process.

2.1 Forward-inverse optics in vision

We first introduce our framework in the context of a seminal example: work in Neuroscience and Computer vision has long since tried to address how a scene can be related to a high level internal representation. Such a question can be framed using two objects: (1) the mapping of a 3D scene to its perceived (2D) image, called *forward optics*, (2) the converse mapping, called *inverse optics*, (see e.g. Kawato et al. (1993)).

A key difference between both maps is that forward optics can be concisely described with a restricted set of equations taking into account physical parameters of the scene, while inverse optics does not have an explicit form and relies heavily on prior assumptions to be solved numerically. This has led in particular to forward-inverse approaches to visual perception, relying on the assumption that forward optics can be implemented by feedback projections from higher to lower visual areas, while feedforward projections provide an initial rough estimate of the inverse optics Kawato et al. (1993); Tajima and Watanabe (2011). The forward optics was then further refined iteratively, following a *predictive coding principle* Rao and Ballard (1999).

Paralleling this framework, many computer vision algorithms have relied on *inverse graphics* approaches that model both forward and inverse optics simultaneously Kulkarni et al. (2015). In recent years, emphasis has been put on producing compact latent description of the scene in terms of high level features, sometimes coined a graphics code, reflecting a *disentangled* latent representation. However, the resulting architectures do not reflect the fundamental asymmetry in complexity between the original forward and inverse optics maps. Taking as reference VAE and GAN architectures, their forward and inverse maps are indeed typically implemented as mirrored convolutional networks, resulting in two densely connected multilayered architectures whose weights are mostly not interpretable. The simplicity of the forward optics can allow an agent to efficiently manipulate and update internal representations, in order to plan interactions with the outside world for instance. Therefore, we argue that modularity of the forward optics models implemented by generators should be enforced in order to be understood and manipulated easily. This is the general aim of the framework presented in this paper.

2.2 Causal generative models

In our aim to isolate a modular functional structure in deep networks, we now introduce a general formalism. We will rely on the notion of causal generative models to represent any latent variable model used to fit observational data. Causality entails the idea that discovered relationships between variables have some degree of robustness to perturbations of the system under consideration. As a consequence a causal model allows predicting interventions and counterfactuals, and may thus

generalize better. Causal models can be described based on Structural Equations (SEs) of the form

$$Y := f(X_1, X_2, \cdots, X_N, \epsilon)\,,$$

expressing the assignment of a value to variable Y based on values of other variables X_k, with possibly additional exogenous effects accounted for through the random variable ϵ. This expression thus stays valid if something selectively changes on the right hand side variables, and accounts for the robustness or invariance to interventions and counterfactuals expected from causal models as opposed to purely probabilistic ones (see for example Peters et al. (2017); Pearl (2000)). Such SEs can be combined to build a Structural Causal Model made of interdependent modules to represent a more complex system, for which dependencies between variables can be represented by a directed acyclic graph $\mathcal{G}$. Let us use such structural model to represent our generator:

Definition 1 (Causal Generative Model (CGM))**.** *A causal generative model $\mathbb{G}(P_{\mathbf{Z}}, \boldsymbol{S}, G)$ consists in a distribution $P_{\mathbf{Z}}$ over latent variables $\mathbf{Z} = (Z_k)$ that account for exogenous effects, a collection $\boldsymbol{S}$ of structural equations assigning endogenous random variables $\mathbf{V} = (V_k)$ and output I based on values of their endogenous or latent parents $\boldsymbol{Pa}_k$ in the directed acyclic graph G. We assume I has no latent parent, such that it is assigned by two deterministic mappings using either latent or endogenous variables*

$$I = g(\mathbf{Z}) = \tilde{g}(\mathbf{V})\,.$$

The graphical representation of a CGM is exemplified on Fig. 1a.

2.3 Two forms of disentanglement

The notion of model modularity that we propose to isolate in networks can take multiple forms. We introduce here a formal definition of the above concepts of disentangled representation. In order to relate the definition to the concrete examples that will follow, we consider without loss of generality that the generated variable I is meant to be an image.

Definition 2 (Extrinsic disentanglement)**.** *A CGM $\mathbb{G}$ is extrinsically disentangled with respect to endomorphism T and subset of latent variables $\mathcal{L}$, if there exists an endomorphism T' of the latent variables such that for any image generated by a realization $\mathbf{z} = \mathbf{Z}(\omega)$ of the latent variables ($I = g(\mathbf{z})$)*

$$T(I) = g(T'(\mathbf{z}))\,, \tag{1}$$

where $T'(\mathbf{z})$ only affects values of components of $\mathbf{z}$ in $\mathcal{L}$.

The sparsity of the disentanglement is then reflected by the minimal size of the subset $\mathcal{L}$. Extrinsic disentanglement can be seen as a form of intervention on the CGM as illustrated in Fig. 1b. In this figure, we represent the effect of applying a transformation that affects only Z_1 (we thus abusively write $T'(Z_1)$), thus modifying descendant nodes, leading to a modified output $I' = T(I)$. We can easily see that this definition is compatible with the intuitive concept of disentangled representation as used for example in Kulkarni et al. (2015) in the context of inverse graphics, where T would correspond to a change in e.g. illumination of the scene, while T' would simply shift the values of the sparse set of latent variables controlling it.[1] In order for the definition to be useful in a quantitative setting, we would require an approximate version of equation (2), however we mostly use this definition for the purpose of contrasting it with the following.

Definition 3 (Intrinsic disentanglement)**.** *A CGM $\mathbb{G}$ is intrinsically disentangled with respect to endomorphism T and subset of endogenous variables $\mathcal{E}$ if there exists an endomorphism T' such that for any image generated by a realization $\mathbf{z} = \mathbf{Z}(\omega)$ of the latent variables ($I = g(\mathbf{z}) = \tilde{g}(\mathbf{v})$),*

$$T(I) = \tilde{g}(T'(\mathbf{v})) \tag{2}$$

where $T'(\mathbf{v})$ only affects values of endogenous variables in $\mathcal{E}$.

An illustration of this second notion of disentanglement is provided on Fig. 1c, where the split node indicates that the value of V_3 is computed as in the original CGM (Fig. 1a) before applying transformation T' to the outcome. Intrinsic disentanglement directly relates to a causal interpretation

[1] In the context of inverse graphics, it could be argued that transformation T should apply to the unobserved 3D scene (for example a 3D rotation of a face). This is compatible with our definition as long as it exists a (non-necessarily unique) solution of the inverse graphics that can then be composed with T.

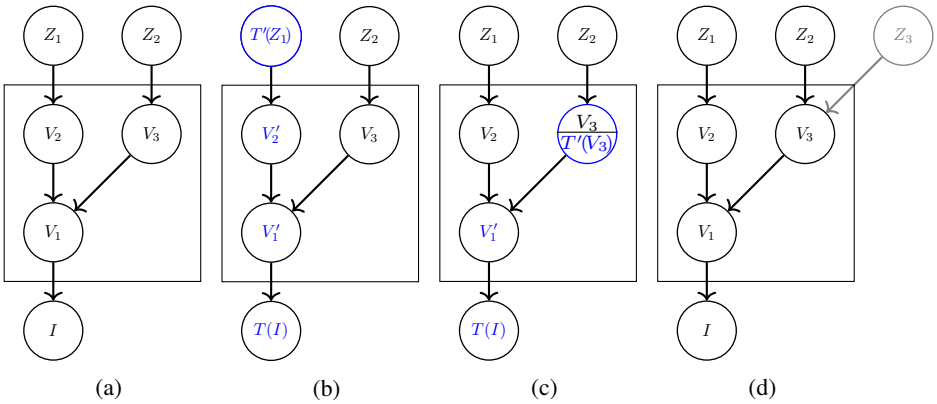

Figure 1: Graphical representation of CGMs. (a) Example CGM with 2 latent variables. (b) Illustration of extrinsic disentanglement with $\mathcal{L} = \{1\}$. (c) Illustration of intrinsic disentanglement with $\mathcal{E} = \{3\}$. (d) Illustration of unaccounted latent variable emulated by (c). Nodes modified by intervention on the graph are indicated in blue.

of the generative model and its robustness to perturbation of its subsystems. To justify it, consider the case of Fig. 1d, where the GCM has an unaccounted latent variable Z_3. This may be due to the absence of significant variations of Z_3 in the training set, or simply bad performance of the estimation algorithm. If the remaining causal structure has been estimated in a satisfactory way, and the full structure is simple enough, a change in this missing variable can be ascribed to a change in only a small subset of the endogenous nodes. Then the transformation T' from the definition can be seen as a proxy for the change in the structural equations induced by a change in Z_3. Broadly construed, appropriate transformations pairs (T, T') emulate changes of unaccounted latent variables, allowing to check whether the fitted causal structure is likely to be robust to plausible changes in the dataset.

2.4 Related work

The issue of interpretability in convolutional neural networks has been the topic of intensive investigation. Most of that research however has focused on discriminative neural networks, not generative ones. In the discriminative case, efforts have been made to find optimal activation patterns for filters (Zeiler and Fergus (2014),Dosovitskiy and Brox (2016)), to find correlation between intermediate feature space and data features (Fong and Vedaldi (2017),Zhang et al. (2017b)) or to disentangle patterns detected by various filters to compute an explanatory graph Zhang et al. (2017a). Furthermore, explicitly enforcing modularity in networks has been tried recently with Capsule networks architectures (Sabour et al. (2017)), although Capsule network explicitly separate the architecture in different modules before training. A more detailed overview can found in review Zhang and Zhu (2018). It is important to emphasize discriminative and generative processes differ significantly, and working on generative processes allows to directly observe the effect of changes in intermediate representations on the generated picture rather than having to correlate it back input images.

The recent InfoGAN network (Chen et al. (2016)) and other works (Mathieu et al. (2016); Kulkarni et al. (2015); Higgins et al. (2017)) in disentanglement of latent variables in generative models can be seen as what we define as extrinsic disentanglement. As such, we believe our intrinsic disentanglement perspective should be complementary with such approaches and are not in direct competition.

Finally our approach relates to modularity and invariance principles formulated in the field of causality, in particular to Besserve et al. (2018).

3 Quantifying modularity in deep generative models

Quantifying modularity of a given GCM presents several challenges. State of the art deep generative networks are made of densely connected layers, such that modules cannot be identified easily beyond the trivial distinction between successive layers. In addition, analysis of statistical dependencies

between successive nodes in the graph is not likely to help, as the entailed relations are purely deterministic. In addition, the notion of intrinsic disentanglement is specific to transformations T and T', and the relationship between these functions may be very complex. We first propose a very general approach that avoids specifying such transformation pairs.

3.1 GENERATION OF HYBRID SAMPLES

Choosing the transformation to apply at a given module may be challenging without priors on the functional organization of the network. Indeed, values of activations in a given layer may be highly constrained by the precise arrangement of upstream synaptic weights. If we thus apply a given transformation to these activations, the internal variable may leave the manifold it is implicitly embedded in as a result of the model's training.

To avoid assessing directly the properties of such manifold, we rely on their variation across samples to intervene on the system. Indeed, if values of internal variables are generated from regular sampling of the latent variables, they will naturally belong to the right manifold. We exploit this property to generate hybrid samples as follows. Consider two (possibly multidimensional) internal variables M_1 and M_2 consisting in the output values of two independent modules inside the network, represented by two distinct subsets of nodes S_1 and S_2 in the SCM, i.e. $\boldsymbol{M}_1 = (V_k)_{k \in S_1}$ and $\boldsymbol{M}_2 = (V_k)_{k \in S_2}$. We assume the sets are chosen such that the output of the generator is fully determined based on them (any path from a latent input to the output contains an element of either S_1 or S_2). The hybridization procedure is illustrated in Fig. 2a. If we take two independent samples of the latent variable z_1 and z_2, they will generate two *original* samples (that we denote *Original 1* and *Original 2*). We then record the value $\boldsymbol{m}_1(z_1)$ of M_1 when generating Original 1, and the value $\boldsymbol{m}_2(z_2)$ taken by $\boldsymbol{M}_2$ when generating Original 2. $\boldsymbol{m}_1(z_1)$ and $\boldsymbol{m}_2(z_2)$ are assumed to encode different aspects of their corresponding generated images, such that one can generate a *hybrid sample* mixing these features by feeding these values $(m_1(z_1), m_2(z_2))$ to the downstream part of the generator network. The modularity assumption then justifies that $(m_1(z_1), m_2(z_2))$ is on the manifold of possible (M_1, M_2) values and therefore that the hybrid output should represent a reasonable sample from the distribution.

3.2 COMPUTING INFLUENCE MAPS

A challenge with the above hybridization approach is to select the size of the groups of nodes to intervene on, especially with networks containing a large amount of units or channels per layer. We propose a fine to coarse approach to extract such groups. First, we study (in a way detailed below) each elementary unit/channel of the network and characterize their functional role in the generation process as an *influence map*. Then influence maps are grouped by similarity to define modules at a coarser scale.

The influence map associated to a given channel k at layer l within a network is obtained by repetitively generating pairs $(z_1,\ z_2)$ from the latent space, where both vectors are sampled i.i.d. independently from each other. We then generate hybrids by choosing S_1 from the above section to be the set of all channels in layer l except the k-th channel, while S_2 contains only the remaining channel k. The fluctuations of the hybrid output across samples are then characterized by computing their variance at each pixel and color channel. We thus obtain for each channel k in a given layer an influence map that has the dimension of the generators' output. Unless otherwise specified, color information is then removed from the influence maps by averaging across color channels.

3.3 CLUSTERING GROUPS BASED ON INFLUENCE MAPS

Representative influence maps for a VAE trained on the CelebA face dataset (see result section) are shown on Fig. 2b and suggests channels are to some extent functionally segregated, with for example some influencing finer face feature (eyes, mouth,...) and other 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 image.

In order to achieve this grouping in an unsupervised way, we perform clustering of the influence maps. In order to extract the overall influence patterns from each map that can be used to evaluate similarity between them, we first pre-process individual maps. A local averaging with a small rectangular sliding window is applied to smooth the maps spatially. Then maps are thresholded at the 75% percentile of

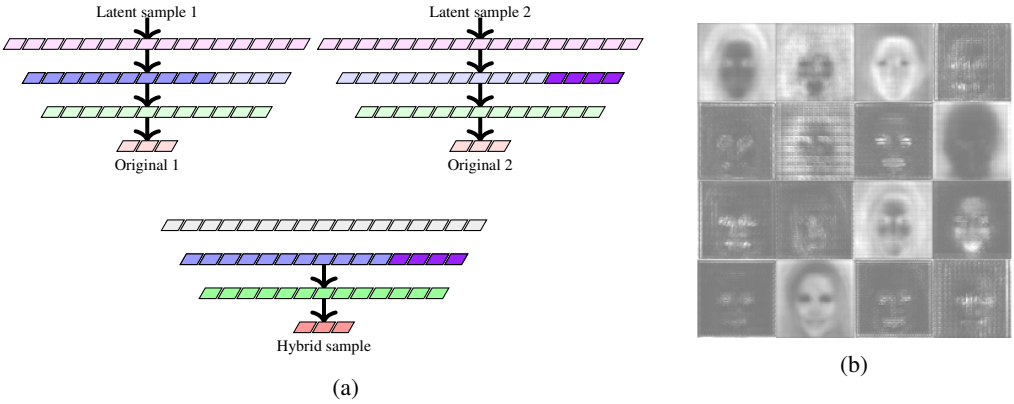

Figure 2: **Generation of influence maps.** (a) Principle of sample hybridization through counterfactuals. (b) Example of influence maps generated by a VAE on the CelebA dataset (lighter pixel indicate larger variance and thus stronger influence of the perturbations on that pixel).

the distribution of values over the image to get a binary image. After flattening image dimensions we get a (channel$\times$pixels) matrix S which is then fed to a non-negative matrix factorization algorithm with manually selected rank K, leading to the factorization $S = WH$. From the two resulting factor matrices, we get the cluster template patterns (the K rows of H after reshaping to image dimensions), and the weights representing the contribution of each of these pattern to individual maps (encoded in W). Each influence map is then ascribed a cluster based on which template pattern contributes to it with maximum weight.

4 Experiments

We investigated our approach on real data in the form of the CelebFaces Attributes Dataset (CelebA)[2]. We used the official tensorlayer DCGAN implementation[3] and a plain β-VAE[4] (Higgins et al. (2017)). The general structure of the VAE is summarized in Fig. 5 and the DCGAN architecture is very similar. We separate the different layers in 4 levels indicated in Fig.5: coarse (closest to latent variables), intermediate, fine and image level (closest to the image). Complete architecture details are provided in the supplemental material. Unless otherwise stated, original samples generated by the VAEs result from the pass of a real image trough the encoder.

4.1 Influence map clustering and hybridization in VAEs

We ran the full procedure described in previous section, comprised of influence maps calculation, clustering of channels into modules, and hybridization at the module level. Unless otherwise stated, hybridization procedures are performed by intervening at the output of the first convolutional layer. The results are summarized in Fig. 3. We observed empirically that setting the number of clusters to 3 lead 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. Exemplary influence maps shown in Fig. 3 (center panel) reflect also our general observation: some maps may spread over image locations reflecting different clusters. However, being more selective by excluding maps that are not *pure* from each clusters comes at the cost of reducing the influence of interventions on the resulting modules (result not shown). Interestingly, applying the hybridization procedure to the resulting 3 modules leads a replacement of the targeted features associated to the module we intervene on, as shown in Fig. 3 (right panel). For example, on the middle row we see the facial feature of the *Original 2* samples are inserted in the *Original 1* image (show on the left), while preserving the hair. We found that the main feature hybridization experiments a rather consistent for reasonable choices of model parameters. In particular, the VAE model used in this experiment was making a trade-off between the sharpness of reconstructed images and the quality of images generated by sampling latent variables

[2] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
[3] https://github.com/tensorlayer/dcgan
[4] https://github.com/yzwxx/vae-celebA

from the isotropic Gaussian prior. By decreasing the β parameter, we can put more emphasis on the quality of reconstructed images. Performance of our procedure on such model (β divided by 10) is shown in Fig. 6 in appendix, where we can see comparable hybridization performance with better image quality.

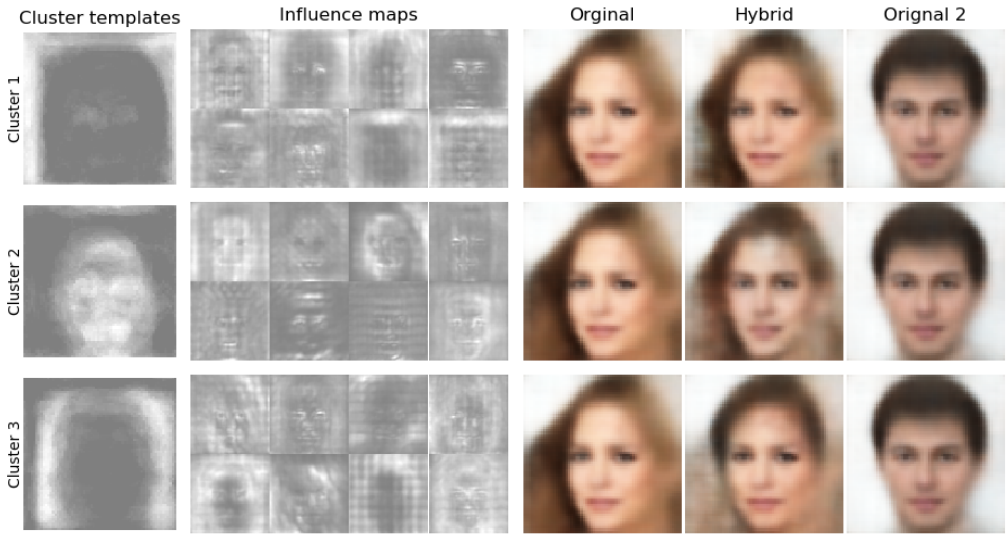

Figure 3: Clustering of influence maps and generation of hybrid samples for a VAE trained on the CelebA dataset (see text).

4.2 Influence map clustering and hybridization in GANs

We replicated the above approach for GANs on the CelebA dataset. The result shown in Fig. summarize the main differences. First, the use of three cluster seemed again optimal according to the stability of the obtained cluster templates. However, we observed that the eyes and mouth location were associated with the top of the head in one cluster. While the rest of the face and the sides of the image (including hair and background) respectively form the two remaining clusters. In this sense, the GAN cluster are less on par with high level concept reflecting the causal structure of these images. However, such clustering still allows a good visual quality of hybrid samples.

4.3 Application to the CIFAR dataset

Additional preliminary experiments were conducted on the CIFAR10 dataset made up of 50000 pictures of animals and vehicles from 10 different categories. Fig. 7. Overall, the clustering procedure is more challenging to adjust, although several influence maps are clearly associated to objects in the foreground, and others to the background.

5 Discussion

The purpose of this paper is to introduce a methodology to assess modularity in deep networks. Modularity may involve different aspects, and strongly depends on the nature of the modeled data. In this paper, we focused on features of the image that preferentially occur in specific parts of the generated images. This is a reasonable assumption for the CelebA dataset, given that the faces are spatially aligned. To some extent this is also true for the CIFAR10 dataset, where objects preferentially appear at the center of the image and the soil and sky in the background will be found at the bottom and top respectively. This approach may however have some limitations when looking at different datasets deprived from such spatial organization. In this case, capturing the structure

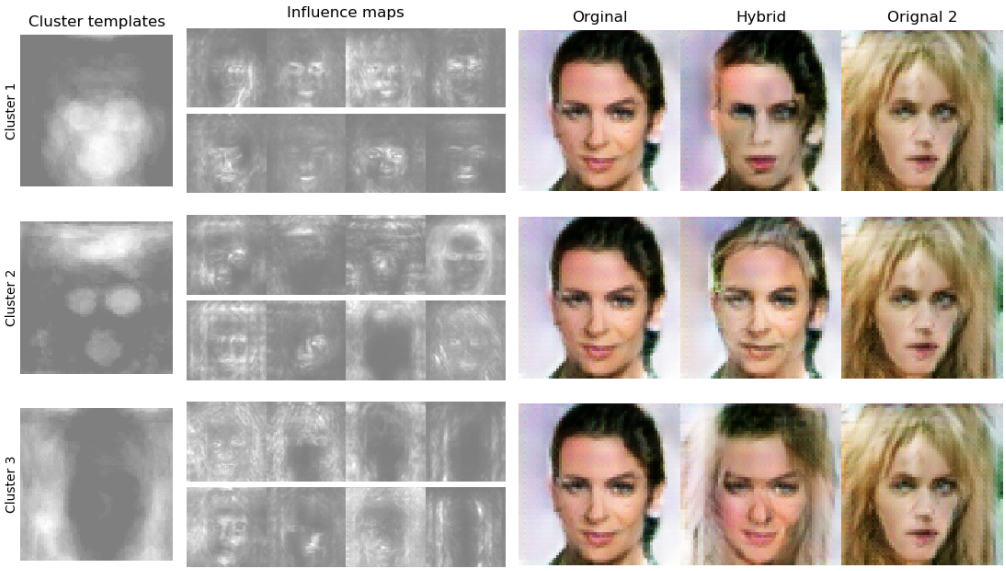

Figure 4: Clustering of influence maps and generation of hybrid samples for a GAN trained on the CelebA dataset (see text).

of output variations induced by hybridization may require a more general approach. In principle, multidimensional technique such as Principal Component Analysis and its non-linear generalization may be able to characterize each channels in order to further generate relevant module following the steps described in the present work.

Another aspect that is left to further work is how to optimize modularity in deep generative networks. As stated in our background section, we believe that classical (extrinsic) disentanglement approaches are not the solution, as they focus on the input of the network without control on its internal structure. While current generative models seem to exhibit some amount of modularity, improving modularity may require specific learning objectives as well as an appropriate choice of architectures.

6 Conclusion

We propose approaching modularity in generative networks by intervening of their internal variables. To assess this notion of *intrinsic disentanglement*, we analyzed generative networks as Causal Generative Models and define procedure to characterize the role played by different groups of channels in these architectures. We found evidence of modularity in VAEs and GANs trained to generated images of human faces. The hybridization still allows to mix some properties of foreground and background element (e.g. for cluters 2 and 3), but they remain entangled. The performance of our approach on this dataset may be affected by the poorer overall performance of our generative models on this dataset.

References

M. Besserve, N. Shajarisales, B. Schölkopf, and D. Janzing. Group invariance principles for causal generative models. In *AISTATS*, 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*, pages 2172–2180, 2016.

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

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

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*, pages 2672–2680, 2014.

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.

Mitsuo Kawato, Hideki Hayakawa, and Toshio Inui. A forward-inverse optics model of reciprocal connections between visual cortical areas. *Network: Computation in Neural Systems*, 4(4):415–422, 1993.

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*, pages 2539–2547, 2015.

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*, pages 5041–5049, 2016.

J. Pearl. *Causality: models, reasoning and inference*, volume 29. Cambridge Univ Press, 2000.

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.

Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79, 1999.

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

S. Sabour, N. Frosst, and G. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3859–3869, 2017.

Satohiro Tajima and Masataka Watanabe. Acquisition of nonlinear forward optics in generative models: two-stage "downside-up" learning for occluded vision. *Neural Netw*, 24(2):148–158, 2011.

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

Q. Zhang and S. Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

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

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

APPENDIX

6.1 NETWORK ARCHITECTURE

The general VAE architecture is presented in Fig. 5 for the particular case of the CelebA dataset. All other architectures used in our experiments follow the same general architecture with hyperparameters specified in Table 1

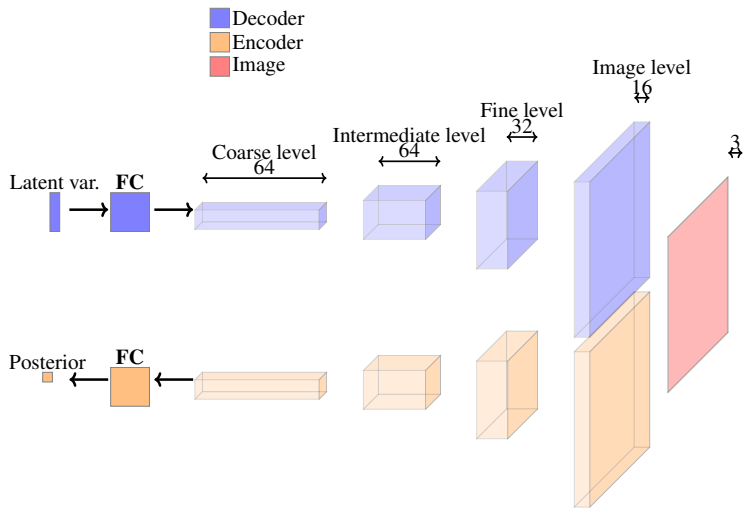

Figure 5: FC indicates a fully connected layer, z is a 100-dimensional isotropic Gaussian vector, horizontal dimensions indicate the number of channels of each layer. The output image size is 64×64 (or 32×32 for cifar10) pixels and these dimensions drop by a factor 2 from layer to layer. (reproduced from (Radford et al., 2015).

NETWORK HYPERPARAMETERS

Default network hyperparameters are summarized in Table 1 (they apply unless otherwise stated in main text).

Architecture	VAE CelebA	GAN CelebA	VAE cifar10	GAN cifar10
Nb. of deconv. layers/channels of generator	4/(64,64,32,16,3)	4/(128,64,32,16,3)	3/(64,32,16,3)	3/(64,32,16,3)
Size of activation maps of generator	(8,16,32,64)	(4,8,16,32)	(4,8,16)	(4,8,16)
Latent space	128	150	128	150
Optimization algorithm	Adam ($\beta = 0.5$)	Adam ($\beta = 0.5$)	Adam ($\beta = 0.5$)	Adam ($\beta = 0.5$)
Minimized objective	VAE loss (Gaussian posteriors)	GAN loss	VAE loss (Gaussian posteriors)	GAN loss
batch size	64	64	64	64
Beta parameter	0.0005	NA	0.0005	NA

Table 1

ONLINE SUPPLEMENTARY MATERIAL

Supplementary files are available at `https://www.dropbox.com/sh/4qnjictmh4a2soq/AAAa5brzPDlt69QOc9n2K4uOa?dl=0`

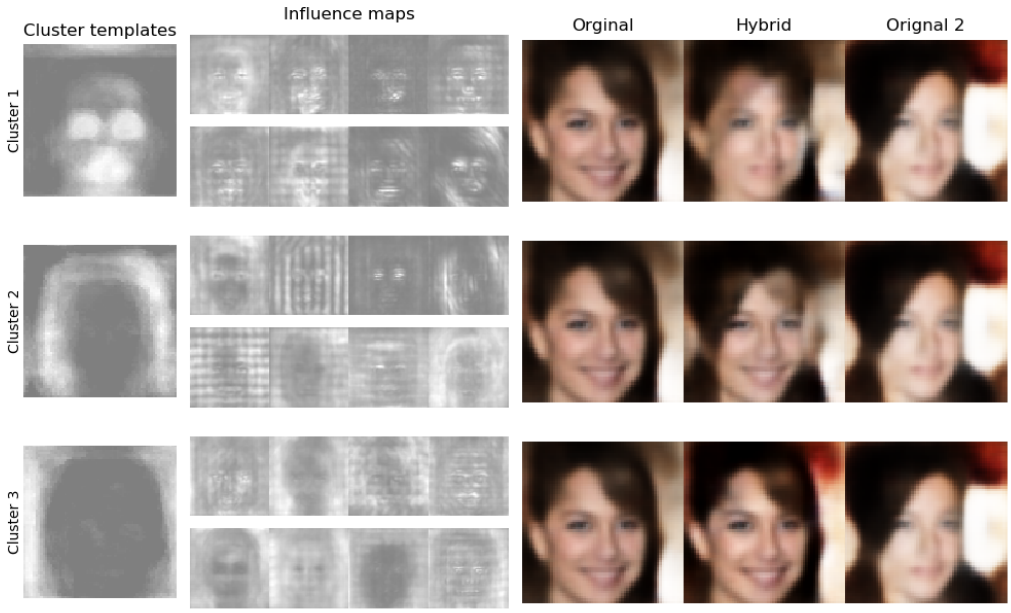

Figure 6: Clustering of influence maps and generation of hybrid samples for a VAE trained on the CelebA dataset (see text).

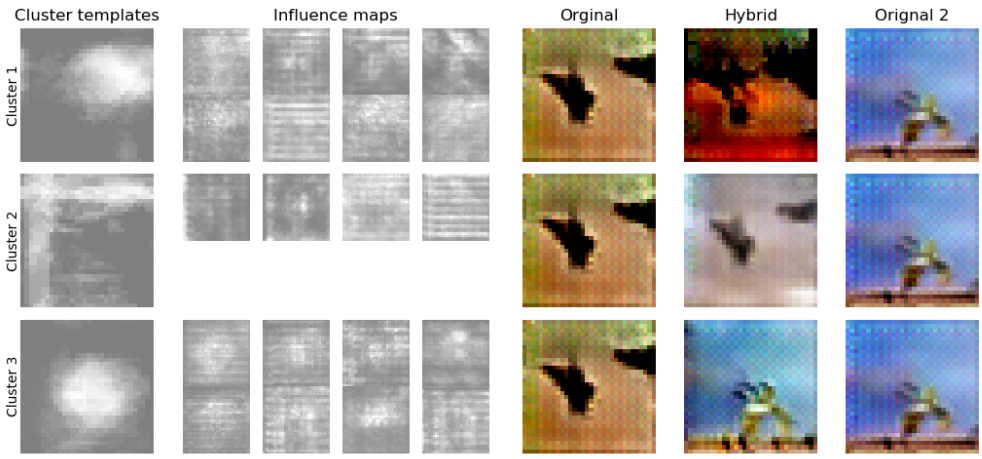

Figure 7: Clustering of influence maps and generation of hybrid samples for a GAN trained on the CIFAR10 dataset (see text).