Neural Computing and Applications
https://doi.org/10.1007/s00521-022-07890-2

ORIGINAL ARTICLE

Comparing the latent space of generative models

Andrea Asperti[1] · Valerio Tonelli[1]

Received: 16 July 2022 / Accepted: 22 September 2022

Abstract
Different encodings of datapoints in the latent space of latent-vector generative models may result in more or less effective and disentangled characterizations of the different explanatory factors of variation behind the data. Many works have been recently devoted to the exploration of the latent space of specific models, mostly focused on the study of how features are disentangled and of how trajectories producing desired alterations of data in the visible space can be found. In this work we address the more general problem of comparing the latent spaces of different models, looking for transformations between them. We confined the investigation to the familiar and largely investigated case of generative models for the data manifold of human faces. The surprising, preliminary result reported in this article is that (provided models have not been taught or explicitly conceived to act differently) a simple linear mapping is enough to pass from a latent space to another while preserving most of the information. This is full of consequences for representation learning, potentially paving the way to the transformation of editing trajectories from one space to another, or the adaptation of disentanglement techniques between different generative domains.

Keywords Generative models · Latent space · Representation learning · Generative adversarial networks · Variational autoencoders

1 Introduction

The task of generating new data from samples has always exerted a particular fascination in machine learning, both because of the potential for almost endless streams of new and original data, as well as for the implications on the knowledge extracted by a model about the data manifold. It is clear that the effectiveness of generative techniques crucially depends on data representation, and different encodings may result in more or less entangled combinations of the different explanatory factors of variation behind the data [1, 2]. The key idea behind unsupervised learning of disentangled representations is that real-world data depends on a relatively small number of explanatory factors of variation which can be compressed and recovered by unsupervised learning techniques [3–5]. Strictly related to representation learning, the task of *exploration* of the latent space of generative models aims to understand the "arithmetic" of the variational factors [6, 7], and the effect that particular trajectories inside the latent space could produce in the visible domain [8–10].

In spite of the huge amount of work devoted to the exploration of latent spaces, relatively little attention has been so far devoted to the problem of *comparing* the latent space of different generative techniques, i.e., to the problem of locating the internal representation z_X of X in a given space starting from its representation in the latent space of a different model (see Fig. 1).

The key questions we are interested in are the following:

1. Do different trainings of the same generative model induce the extraction of similar features from data, and hence substantially isomorphic spaces up to, say permutations or linear transformations? We refer to this type of transformations as being of Type 1.
2. Do different architectural models driven by common learning objectives (e.g., maximizing log-likelihood)

✉ Andrea Asperti
andrea.asperti@unibo.it

Valerio Tonelli
valerio.tonelli2@studio.unibo.it

[1] Department of Informatics: Science and Engineering (DISI), University of Bologna, Mura Anteo Zamboni 7, 40126 Bologna, Italy

Published online: 07 October 2022

learn similar features? How much do the extracted features depend on the neural network structure? We refer to this type of transformations, between spaces of variants of models in the same class, as being of Type 2.

3. Finally, what is the influence of the learning objective on the internal representation? Is, e.g., a Generative Adversarial Network learning the same features of a Variational AutoEncoder? We refer to these transformations as being of Type 3.

Any answer, whether positive or negative, could substantially improve our knowledge of generative techniques.

Our surprising preliminary results, reported in this article, seem to suggest that (provided models have not been taught or explicitly conceived to act differently) it seems to be possible to pass from a latent space to another by means of a *simple linear mapping* preserving most of the information.

This linear transformation may be computed directly through linear regression, but we advocate a learning-based technique based on a suitable small "support set" of data samples enucleating, in the visible space, the key variational factors of the data manifold. When we say "small", we mean that the set has a cardinality comparable with the number of variables in the latent space (so, *really small*): for instance, in the case of CelebA, we experimented with a support set of 150 images. Locating these 150 samples in the two spaces is enough to allow the definition of a relocation map for all data.

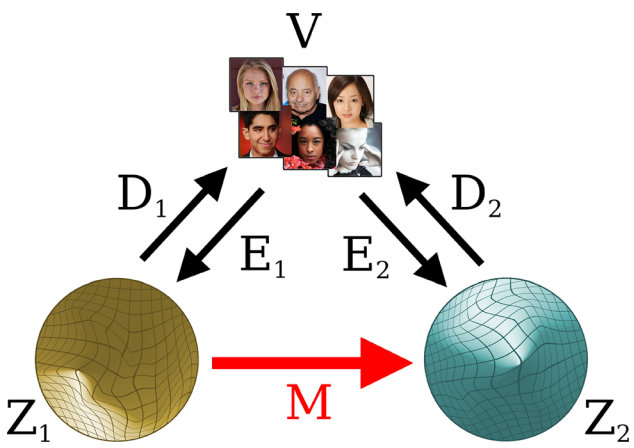

Fig. 1 Given a generative model, it is usually possible to have an encoder-decoder pair mapping the visible space to the latent one (even GANs can be inverted, see Sect. 2.2.1). From this assumption, it is always possible to map an internal representation in a space Z_1 to the corresponding internal representation in a different space Z_2 by passing through the visible domain. This provides a supervised set of input/output pairs: we can try to learn a direct map, as simple as possible. The astonishing fact is that a simple linear map gives excellent results, in many situations. This is quite surprising, given that both encoder and decoder functions are modeled by deep, nonlinear transformations

The main results of our investigation are summarized in Fig. 2. Figure. 2a describes an example of relocation between different trainings of a same network (relocation of Type 1); Fig. 2b is relative to the relocation between different models of a same class—two different VAEs, in this case (relocation of Type 2); Fig. 2c is an example of relocation from a VAE to a GAN, that is between different models with different learning objectives (relocation of Type 3). While details may slightly differ, especially for transformations between different generative models, the overall appearance (pose, colors and background) is substantially preserved. Considering the nonlinearity of these generative processes, the result is, at a first glance, quite surprising: pairs of points related by a simple linear mapping in the latent spaces of two different generative models are decoded by the respective decoders in closely related—in some cases almost identical—images!

1.1 Structure of the article

The structure of the article is the following. We start by providing, in Sect. 2, a quick introduction to generative modeling, and in particular to latent variables models, comprising the popular Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs); in this section, we also discuss the problem of inverting GANs. Section 3 covers the domain of semantic exploration of latent spaces, representation learning and disentanglement. In Sect. 4 we start introducing the datasets, the models and the methodology that we used for our experiments. Since we focus on linear transformations, they can be defined by a small set of points, that we call Support Set: locating the points in the Support Set in the two latent spaces is enough to define the transformation. Our approach to get a good Support Set is discussed in Sect. 5. In Sect. 6 we give numerical results about the mappings (visual examples, more readily interpretable, are spread over the article). Section 7 is devoted to the discussion of the latent space of StyleGAN that seems to present some pathological issues: many faces in the CelebA dataset lie outside of its generative range. Even in this case, however, provided we confine the transformation to the StyleGAN subspace, we discover interesting linear mapping to other spaces. Conclusion and future works are discussed in Sect. 8. Additional material is given in appendices: a detailed description of the models used in this work (Sect. A), a full list of all images in the CelebA Support Set (Sect. B).

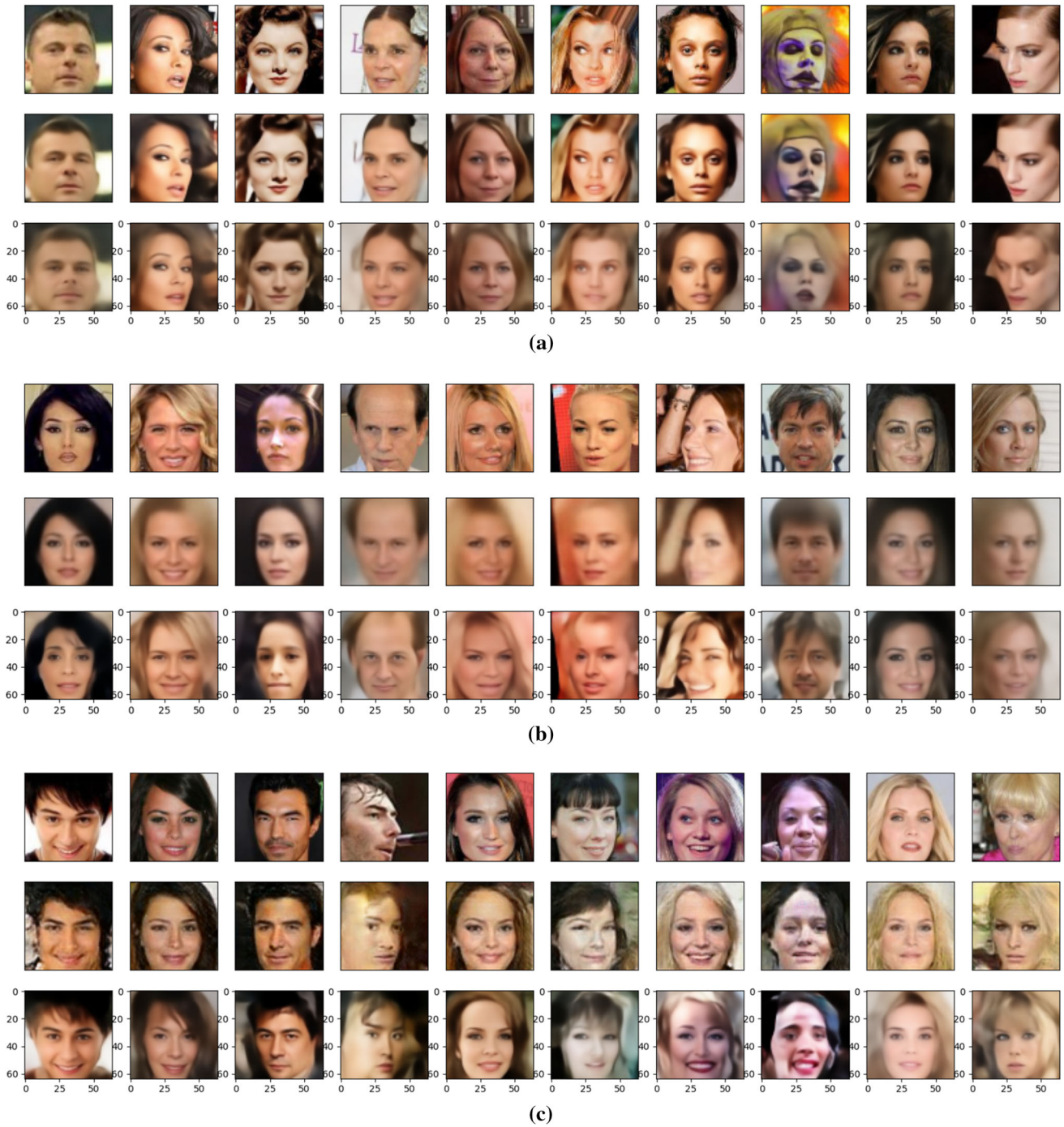

Fig. 2 Examples of relocations of different Types. In the first row we have the original, in the second row the image reconstructed by the first generative model, and in the third row the image obtained by the second model after linear relocation in its space. (a) Relocation of Type 1, between latent spaces relative to different training instances of the same generative model, in this case a particular Variational Autoencoder [11] . The two reconstructions are almost identical. (b) Relocation of Type 2, between a Vanilla VAE and a state-of-the-art Split-VAE [11] . The SVAE produces better quality images, even if not necessarily in the direction of the original: the information lost by the VAE during encoding cannot be recovered by the SVAE, which instead makes a reasonable guess. (c) Relocation of Type 3, between a vanilla GAN and a SVAE. Additional examples involving StyleGAN are given in Sect. 7. To map the original image (First row) into the latent space of the GAN we use an inversion network. Details of reconstructions may slightly differ, but colors pose and the overall appearance is surprisingly similar. In some cases (e.g., the First picture) the reconstruction re-generated by the VAE (from the GAN encoding) is closer to the original than that of the GAN itself.

2 Generative modeling

Generative modeling is the task of learning the high-dimensional probability distribution of a data manifold starting from a representative set of samples. When successfully trained, generative models can be used to create new samples from the underlying distribution, possibly providing estimations of their likelihood. The learning process provides an essential and valuable insight of the kind of features used to encode the distribution, and the way the model "interpreted" and "understood" data.

At the heart of generative techniques there is a relatively small set of techniques [12, 13]: Auto-Regressive models [14, 15], Flow models [16–19], Energy-based models [20–22] and Latent-Variable models, particularly GANs [6, 23, 24] and VAEs [25–27].

In this article, we shall mostly focus on the popular and effective Latent-Variable models, that is models where the actual distribution $p(x)$ of a data point x is expressed through marginalization over a vector z of *latent variables*:

$$p(x) = \int_z p(x|z)p(z)dz = \mathbb{E}_{p(z)}[p(x|z)]$$

where z is the latent encoding of x distributed with a known distribution $p(z)$ named *prior distribution*. The distribution $p(x|z)$ is usually learned by a deep neural network; after training it can be used to generate new samples via ancestral sampling:

1. sample $z \sim p(z)$;
2. generate $x \sim p(x|z)$.

2.1 Variational autoencoders

A Variational AutoEncoder (VAE) [28] has a structure similar to a classical auto-encoder [29, 30], being composed of an *encoder* producing a latent vector z from an input x and of a *decoder* which reconstructs the input $\hat{x}$ from a latent code; the two components are simultaneously trained using, e.g., a mean squared error loss $|x - \hat{x}|_2$. However, in order to regularize the latent space, which is a precondition to support semantically meaningful generation [12], latent variables are interpreted as parameters of a local distribution $q(z|x)$ and a Kullback–Leibler component $KL(q(z|x) \parallel \mathcal{N}(0,1))$ is added to the reconstruction loss, with the purpose of pushing the marginal distribution $q(z)$ toward a standard Gaussian $\mathcal{N}(0,1)$. Balancing of these two loss components, usually via a γ or β parameter, is crucial for better generation and learning of disentangled features [31–33].

Several issues affect the performance of VAEs, most importantly blurriness of generated images [34]. As such, many variants have been proposed over the years to improve results by addressing the mismatch between the aggregate inference distribution $q(z)$ and the prior $p(z)$. These comprise: quantization of the latent code (VQ-VAE [35]), use of normalizing flows (Hybrid VAE [36]), two-Stage architectures [37], and hierarchical models [16, 38].

2.2 Generative adversarial networks

In a Generative Adversarial Network (GAN) [6, 39, 40] a *generator*, acting as a sampler for the desired distribution, is jointly trained with a *discriminator*, evaluating the output of the generator by attempting to distinguish real from generated ("fake") data. This can be formalized in the form of a zero-sum game, where one agent's gain is another agent's loss; the generator and the discriminator must be trained alternately, freezing the respective adversarial component; at the end of the process the generator is supposed to win, producing samples that the discriminator is unable to distinguish from real.

GANs are known to have unstable training and several issues among which the well-known mode collapse phenomenon [40]. Indeed, multiple variations for the loss function have been studied over time [41], including the Wasserstein loss [42], least squares loss [43] or the introduction of a penalty term for the discriminator [44]. Furthermore, a myriad of variations on the structure itself have been proposed, among which: maximizing the mutual information between specific latent variables [45]; exploiting pairs of GANs to perform style transfer between images in distinct datasets [46]; GANs with attention layers [47].

A particularly interesting series of works come from the application of style transfer concepts to GANs (StyleGAN and its successors [48–50]). StyleGAN builds on Progressive GANs [24], whose structure is unchanged from that of a baseline GAN but is implemented progressively: the architecture is trained starting from down-sampled images at very low resolution, and at each progression step the input size is increased while additional layers are introduced to both generator and discriminator.

StyleGAN further builds on this structure by adding to the generator (*Synthesis* network) a fully connected *Mapping* network which takes the usual seed $z \in Z$ and produces a "style" vector $w \in W$. This vector is then specialized per-layer through Adaptive Instance Normalization (AdaIN), which according to the authors produce a behavior similar to style-transfer. Furthermore, a small amount of noise is added to all blocks of the Synthesis network to better fill in the output details. The full structure of StyleGAN can be seen in Fig. 3.

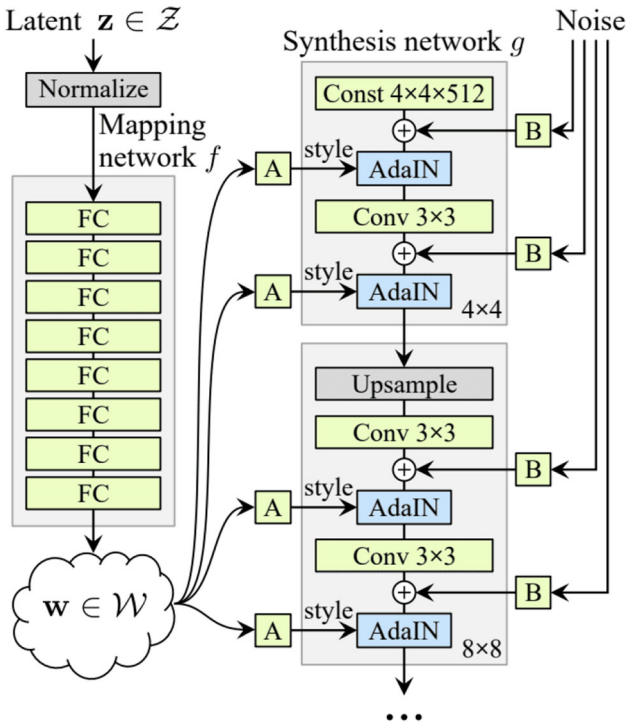

Fig. 3 Structure of the StyleGAN generative network (picture from [48]). Observe: (1) the two distinct latent spaces Z and W; (2) the *mapping network* taking a randomly sampled point $z \in Z$ as input and generating a *style vector* w; (3) the use of Adaptive Instance Normalization, or AdaIN (Blocks A), to apply style vectors after each convolution layer of the Synthesis network; (4) the exploitation of noise as an additional source of randomness passed through learned scaling layers (Blocks B)

2.2.1 GAN inversion

The generator of a GAN usually takes as input a seed $z \sim \mathcal{N}(0, 1)$, and has a role directly comparable to that of a VAE decoder. However, GANs lack a direct encoding process of the original input sample, unlike a VAE encoder. If, as is the case for our study, both generative and encoding processes are needed, a third neural network has to be added to a pre-trained GAN as a sort of plug-in encoder. This re-coder component is known as an inverse GAN, and building an accurate re-coder is a known problem in the literature [51].

Several approaches to inversion have been explored [52–55], mostly for editing applications, the simplest being SGD optimization [56] or a learning-based approach such as using a neural network trained on generated images to reconstruct the original latent vector using a mean squared error loss $\mid z - \hat{z} \mid_2$, with the advantage that over-fitting is never an issue since training is not constrained to samples of the original data. Hybrid methods combining both efforts have also been explored [57, 58].

Recent works have focused mostly on the inversion of the popular StyleGAN, building on previous work with a variety of inversion structures and minimization objectives [59–63] with the aim of generalization to any dataset. However, we used a simpler and narrow approach by developing our own StyleGAN inverter for the W space using a naive recoding network. It works surprisingly well for commonly generated samples, with a final mean square error close to 0.0040. We show some examples of recoding in Fig. 4.

3 Semantic interpretation of latent spaces

The latent space of a generative model efficiently synthesizes information from data, however, the resulting compressed vectors cannot be easily mapped onto understandable features such as labels or attributes. Therefore, it is also unknown how exactly a model learns from data, in terms of how well it encodes its features, biases and human-meaningful characteristics. At the same time, this knowledge could fundamentally influence the quality of models and provide a foundation on which to improve their performances without relying solely on empirical and qualitative analyses.

Conditional architectures [45, 64] can indeed mitigate this issue by explicitly feeding features alongside samples during training, but in doing so they remodel the task as a supervised problem with respect to the classes on which conditioning is done, with all other data features remaining non-explainable. These approaches do not provide interesting information about the way the neural network understand data, and for this reason, they will not be discussed in this work.

3.1 Exploration and disentanglement

Many works attempt to understand the latent space of GANs by performing exploration on the latent space, that is, they introduce small nudges in a direction based on the empirical principle that they will correspond to a small change in the corresponding generated data. The approach can be particularly useful for image editing, as once a semantically meaningful direction is found (e.g., color, pose, and shape), it can be traveled to tweak an image, introducing a desired feature without the need for a conditional generation model. InterFaceGAN [8] supposes that for a given feature taking values in $(-\infty; \infty)$ there exists an hyperplane in the latent space whose normal vector allows for a gradual modification of the feature, which can be found, e.g., via an SVM [65]. Further work based on this idea searches for these directions as an iterative or an optimization problem [66] and also extend it to controllable walks in the latent space [10].

Fig. 4 Results of our own network for StyleGAN inversion. Images in the first row have been generated by StyleGAN; they are re-coded into the W space and regenerated (second row). The two images are hardly distinguishable. However, as we shall see in Sect. 7, inversion can be more problematic for images outside the generative range of the model; in principle, a good generative model should be able to produce any sample, provided it is not too atypical

A different, more systemic approach to the problem is by [7], which use a closed-form equation to find the editing direction n_i applied per-layer i of a generator, which is then composed to find the overall direction n. Another approach of the same "arithmetic" flavor comes from [67], where a generative application of PCA with a nonlinear kernel is used to determine the hidden features of a small-scale dataset, without any reliance on a particular generative model.

Much less work on exploration has been devoted to VAEs. An example is given by [9], which however works on a conditional architecture, in order to produce lower-dimensionality subspaces that are easier to analyze.

4 Datasets, models, methodology

4.1 Datasets

As stated in the abstract, we confined our analysis to the familiar and largely investigated data manifold of human faces. Our dataset of reference is CelebA [68], including its higher-quality version CelebAHQ [24]. Images taken from CelebA have been aligned as per their paper [68] and then cropped to size 128×128 with a y offset of 45 and an x offset of 25 in order to remove as much background information as possible. The crop is then downsampled to size 64×64 with bilinear interpolation).

CelebaHQ is a dataset of 30K images at resolution 1024×1024, obtained from a subset of CelebA with a complex methodology explained in Appendix C of [24], comprising a sophisticated preprocessing phase, super-resolution techniques, and selection of best quality samples.

4.2 Generative models

For our experiments we took into considerations four different models, two GANs and two VAEs; in each class, we investigated a basic, average quality "vanilla" version and a more sophisticated, state-of-the-art model. A summarizing Table 1 for these models is provided. More in-detail, we have investigated the following architectures:

1. Vanilla VAE [28] using γ balancing [31] with a latent dimension $Z = 64$ trained on the cropped CelebA;
2. Vanilla GAN [39] with a latent dimension $Z = 64$ trained on the cropped CelebA;
3. SVAE [11] with a latent dimension $Z = 150$ trained on the cropped CelebA;
4. StyleGAN [48] pre-trained on CelebA-HQ[1], which has a latent dimension Z of size 512 and a style-vector latent dimension W of the same size.

The structure of the StyleGAN has been already briefly discussed in Sect. 2.2. The in-depth architecture of the other models, not central to the topic of this article, is given in Appendix A.

The dimension of the latent space and the resolution of the different models is summarized in Table 1.

4.3 Methodology

For each one of the previous models, apart StyleGAN where we only had at our disposal a single set of pre-trained parameters, we trained and tested five different instances. When reporting values in the results, if not differently stated, they have to be understood as an average over the different trainings.

[1] We have used the original model weights at https://github.com/NVlabs/stylegan and the Tensorflow 2.0 conversion repository https://github.com/ialhashim/StyleGAN-Tensorflow2.

Table 1 Dimension of the Latent Space and Resolution for the different models

Model	Latent dim	Resolution
GAN	64	64×64
VAE	64	64×64
SVAE	150	64×64
StyleGAN	512	1024×1024

Mapping between different models (transformations of Type 2 and 3) can have a lot of additional issues. Firstly, the two latent spaces may have sensibly different dimensions, for instance 512 for StyleGAN versus 150 for the SVAE and for the other models, and may work at different resolutions, for instance 1024×1024 for StyleGAN versus 64×64 for the other models. Furthermore, the two generative models may have been trained on the two different datasets which, albeit similar, have different data and different crops. To this aim, when passing from CelebA-HQ to CelebA we take a simplified crop of dimension 880×880 with an height offset of 20 and a width offset of 60, which is then downsampled to size 64×64 with bilinear interpolation.

Since we are interested in linear mappings, the transformations may be defined by a small set of "corresponding" points common to both spaces: this is what we call a Support Set. Our methodology to build it is defined in Sect. 5. The support Set is defined in the visible domain; we trace their respective encodings in the different spaces, and define the map by linear regression with mean squared error as a loss. When we cannot use a Support Set, we may directly work with the whole visible domain (or the subset of the visible domain common to the two spaces), sampling minibatches in it.

5 Support set

In this Section we explain the technique used to build a small support set of examples driving the linear transformation. This is based on the following steps, each one detailed in a respective subsection:

features ordering we order latent variables according to their relevance for reconstruction, using a suitable metric discussed below;

features selection we select a small number n of particularly significant latent variables; 2^n must be lower than the cardinality of the support set;

sample selection we select points in the space belonging to extremal regions with respect to the selected features.

5.1 Features ordering

Feature importance—the task of associating a score to input features based on how useful they are for solving a specific problem—is a major subfield of Machine Learning. In the case of generative modeling, the goal is to maximize the (log)likelihood of data, and it is natural to associate a score to features according to their contribution to this objective. It is worth observing that different techniques, like, e.g., PCA, would not be beneficial to this aid, due to the shape of the prior latent distribution which is, typically, a spherical Gaussian distribution[2].

Our feature importance technique requires an encoder in addition to a decoder: it fits particularly well with VAEs, but it can be generalized to GANs by exploiting a re-coder network (see Sect. 2.2.1). Specifically, in order to evaluate the contribution of the variable to the loss function, we compute over a large number of data the average difference between the reconstruction error when the latent variable is zeroed out with respect to the case when it is normally taken into account. We call this information the *reconstruction gain* associated with the latent variable. It was introduced in [69] where it was used to compare the reconstruction error and the Kullback–Leibler divergence on a per-variable base, in order to clarify the variable collapse phenomenon [27, 70, 71].

We did the experiment on the SVAE, which in our experiments has a latent space of 150 variables. In Fig. 5 we show the information gain relative to all its latent variables, ordered by relevance.

Eleven variables have a score higher than 10, although the distribution has a relatively long tail: the first 20 variables are responsible for about 75% of the information.

5.2 Feature selection

We keep a small number of the most informative variables. For the way we shall use it, this number must be smaller than the logarithm of the cardinality of the support set. In our case, we aim to a support set of dimension 150, so we focus on the seven most relevant variables.

In Fig. 6 we show examples of the effect of some of these variables on generated images: we take a random point and progressively modify the given variable in the

[2] Even the potential mismatch between the prior and the aggregate inference distribution in the case of VAEs cannot be exploited by PCA, since this technique only takes into consideration the first two moments of the distribution.

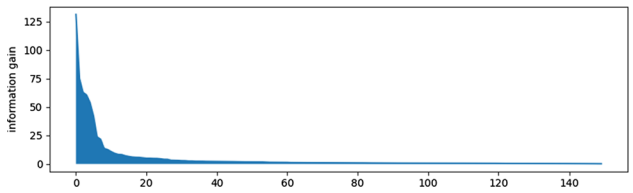

Fig. 5 Information gain for all variables, in decreasing order. Only a bunch of variables are in charge of the macroscopic factors of variations

range between -2.25 and 2.25 (remember that the latent space standard deviation is 1).

5.3 Sample selection

Finally, we divide the latent space in sectors corresponding to extreme values for the previously selected variables, and pick up samples in these sectors.

More precisely, having defined a threshold *th* and a "direction" *dir* given by a $+/-$ sign for each selected variable, a sector defined by the pair (*th*, *dir*) is the set of points with direction compatible with *dir* and at a distance from the origin larger than *th*. Since we consider all possible directions, this gives a total of 2^n sectors where n is the number of selected variables (for a fixed *th*). In each sector, we pick up a sample at random (enlarged *th* sectors become progressively less inhabited).

It is interesting to observe that the number of latent points in the dataset within different sectors at a given threshold is far from uniform. This seems to be a confirmation that the actual image distribution is far from the desired Gaussian normal prior and, in a VAE, a symptom of the potential mismatch between the generative prior and the aggregate inference distribution computed by the encoder, which is a well-known and problematic aspect of VAEs [72–74]. Attempts to solve this issue have been made both by acting on the loss function [75] or by exploiting more complex priors [36, 76, 77]; the actual effects on the latent space of these techniques is an interesting research direction for future investigations.

Fig. 6 Effect of the seven most informative latent variables in the visible domain. Each image is obtained by varying a specific variable in the range [-2.25; +2.25]. Considering these are the variables with the largest information gain, it may be argued that their impact is less pronounced than expected. Most of the variables are associated with a change in luminosity of all or part of the image, possibly associated with modifications in hair color, source of illumination and tiny variations in the pose. In the case of variable 21, there seems to be progressive Female-Male transition (and vice versa for variable 114)

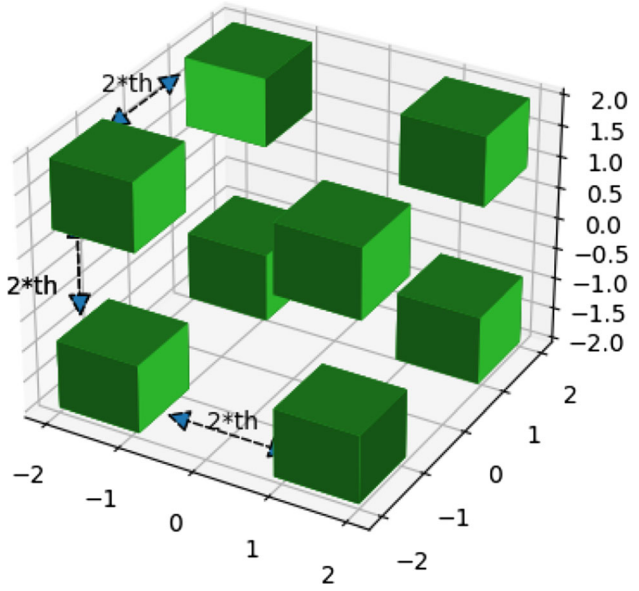

Fig. 7 Example of sectors in three dimensions (cropped to distance 2 from the origin). The distance between sectors is equal to twice a configurable threshold. We work with the seven most informative latent variables, obtaining a total of $2^7 = 128$ sectors

In Figure we show typical inhabitants for a few given sectors. As expected, they share macroscopic features like background color, pose, hairs, and illumination.

Part of the 128 images resulting from our selection process are depicted in Fig. 9. The complete list of labels for the support set is reported in the appendix. The samples in the support set occupy "extreme" positions in the latent space with respect to the most informative directions: for this reason, they as supposed to be representative of the principal factors of variations in the dataset.

As a partial confirmation of the previous hypothesis, we expect the distance between elements in the support set to be sensibly higher than the average distance between points in the full dataset. This is actually the case: the mean squared error between random CelebA images is 0.116, versus 0.183 for samples in the support set.

6 Results

This Section contains numerical results relative to the transformation between latent spaces. The discussion of StyleGAN, for its relevance and some interesting pathological issues, will be postponed to the next Section.

Here, with we shall use the names VAE, GAN and SVAE to refer to our specific implementations of these models, discussed in Sect. 4.2 and detailed in Appendix A.

We build a set of correspondent input-output pairs by encoding the Support Set (or the full set of visible data) into the two latent spaces. Then, we directly build a linear map by linear regression, minimizing the mean squared error between target and computed latent vectors.

For each transformation, we provide three values:

L-MSE Latent Mean Squared Error. This is the loss of the model, namely the mean squared error between the target vectors and those computed by the model;

R-MSE Reconstruction Error. This is the mean squared error between the original image in the visible domain and its reconstruction via the source generative model;

M-MSE Mapped Error. This is the mean squared error, in the visible domain, between original images and images reconstructed by the target generative model after linear mapping.

The three errors are graphically described in Fig. 10.

The latent error L-MSE is not easily deciphered; the comparison between R-MSE and M-MSE provides a more intelligible information about the quality of the translation.

The results are given in Table 2.

For the sake of comparison, it it worth to recall that the mean squared error between CelebA images is 0.116; in all models the M-MSE is always below 0.039.

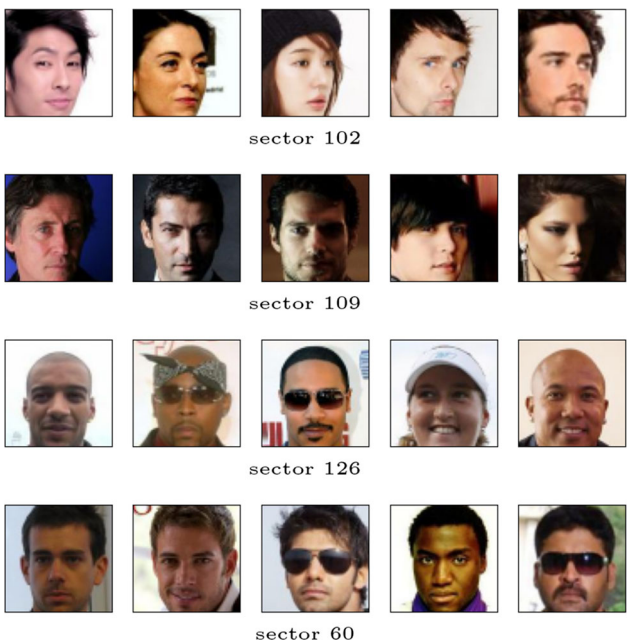

Fig. 8 Examples of data in different sectors. For each sector, images are different, but share macroscopic features: background color, pose, hairs, illumination, etc.

Fig. 9 Part of the images in the support set resulting from our selection process. The samples are supposedly representative of the principal factors of variations in the dataset. Additional examples are given in the appendix

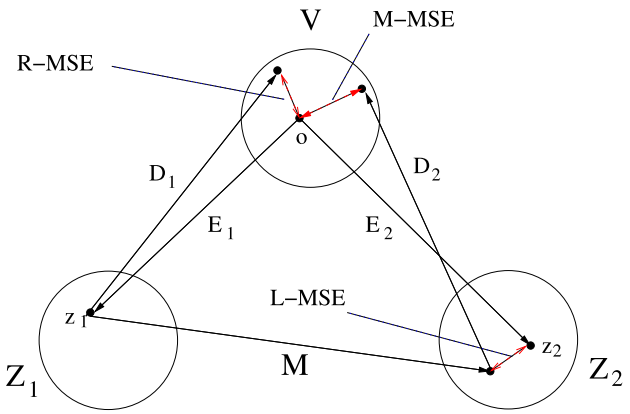

Fig. 10 Relocations Errors. An original point o in the visible domain is mapped into internal representations z_1 and z_2 in the latent spaces Z_1 and Z_2. The map M is trained to reconstruct z_2 from z_1: L-MSE is the mean squared error between z_2 and $M(z_1)$. R-MSE is the mean squared error, in the visible domain, between o and its reconstruction according to the first generative model. M-MSE is the mean squared error, in the visible domain, between o and $D_2(M(z_1))$

7 The StyleGAN space

The "extreme" nature of the images in the Support Set makes them a very natural benchmark of the expressiveness of generative models: is it possible to reconstruct these images by passing them through an encoding-decoding process?

For StyleGAN trained on CelebA-HQ, the results are disappointing (see Fig. 11, and compare them with the inversion of generated images in Fig. 4). Although the macrostructure is preserved (background, pose, and illumination), details are sensibly different. Numerically, while the average mean squared error on generated images is 0.026, the corresponding value for the Support Set is 0.251, almost ten times higher.

Our conjecture is that StyleGAN is simply unable to generate data in the support set: they do not belong to its latent space, specifically due to its training dataset. To check this claim we implemented a gradient ascent technique to generate latent representations corresponding to a desired output. Once again, the gradient ascent technique provides almost perfect results on generated images but substantially fails on images in the CelebA support set, as shown in Fig. 12.

We believe that the latent space of StyleGAN, trained on CelebA-HQ, only faithfully reflects a subspace of the latent space of our other models, trained on the full CelebA dataset. In particular, points in our extreme *sectors* seem to lie outside of the generative range of StyleGAN, or to be severely underrepresented (Fig. 13). The problem is possibly also related to the well-known fact that faces generated by StyleGAN (and other generative networks) can be easily distinguished from reals [78–80].

Table 2 Mapping results for different model pairs: (L-MSE) MSE between the target and Mapped Latent vectors; (R-MSE) MSE between the original and Reconstructed (encoded-decoded) images; (M-MSE) MSE between the original and mapped images via the learned linear mapping. When source and target coincide, we mean different trainings of the same model (Type 1 transformations)

From	To	L-MSE	R-MSE	M-MSE
VAE	VAE	0.03	0.0073	0.0103
VAE	SVAE	0.72	0.0073	0.0105
VAE	GAN	0.49	0.0073	0.0339
GAN	VAE	0.50	0.0284	0.0254
GAN	SVAE	0.86	0.0284	0.0275
GAN	GAN	0.43	0.0284	0.0335
SVAE	VAE	0.195	0.0035	0.0125
SVAE	GAN	0.63	0.0035	0.0388
SVAE	SVAE	0.20	0.0035	0.0067

7.1 Comparison with different spaces

Since exploiting the Support Set is not a viable solution, we need to define a direct mapping by regression on all data. As it is customary for exploration studies, we work with the W StyleGAN space; as matter of fact, the Z space is passed through a long series of fully connected layers (the *Mapping* network) which we presume, by construction, not being linearly invertible.

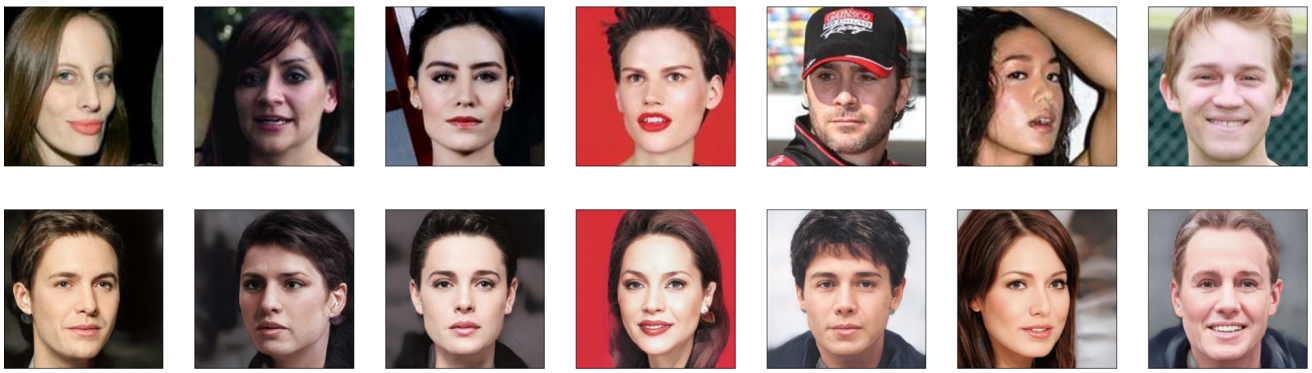

Fig. 11 StyleGAN inversion on images in the Support Set. The macro structure (background, pose, illumination, etc.) is preserved, but all other features are lost: images in the Support Set seem to lie outside of the generative range of StyleGAN. Note also the more "conventional" nature of the images obtained by the inversion

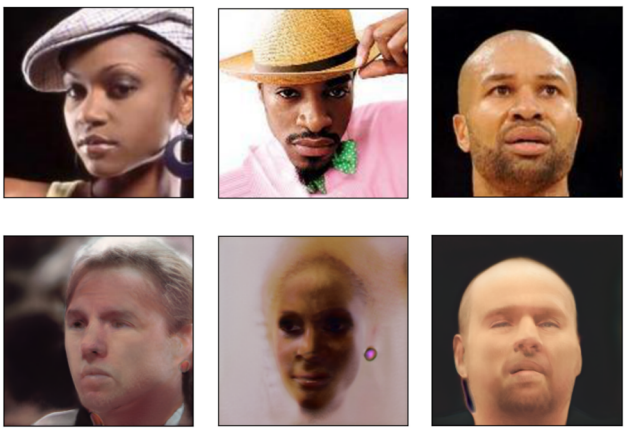

Fig. 12 Gradient ascent technique for StyleGAN on data in the Support Set. The original is in the first row, and the image generated through gradient ascent, in the second. The technique confirms that these images cannot be generated by StyleGAN

Here we try to map the W space of StyleGAN, trained over CelebA-HQ, to the latent space of SVAE trained over CelebA. The input to the transformation map is the vector w, obtained by ancestral sampling from the Z space. The expected output z is obtained by synthesizing with StyleGAN the image corresponding to w, cropping and resizing it to dimension 64×64 and encoding it in the SVAE latent space. The result of the linear map will be called $\hat{z}$; let $SVAE(z)$, and $SVAE(\hat{z})$ the corresponding decodings to the visible domain. As usual, input vectors w may be generated *ad libitum*, with no risk of overfitting.

After training, the mean squared error between z and $\hat{z}$ is around 0.45 with a standard deviation of 0.05. The mean squared error between $SVAE(z)$, and $SVAE(\hat{z})$ is 0.014 with standard deviation of 0.002. All results have been repeated over five different parameters configurations of SVAE, relative to five different trainings (obviously, each experiment results in a different linear transformation).

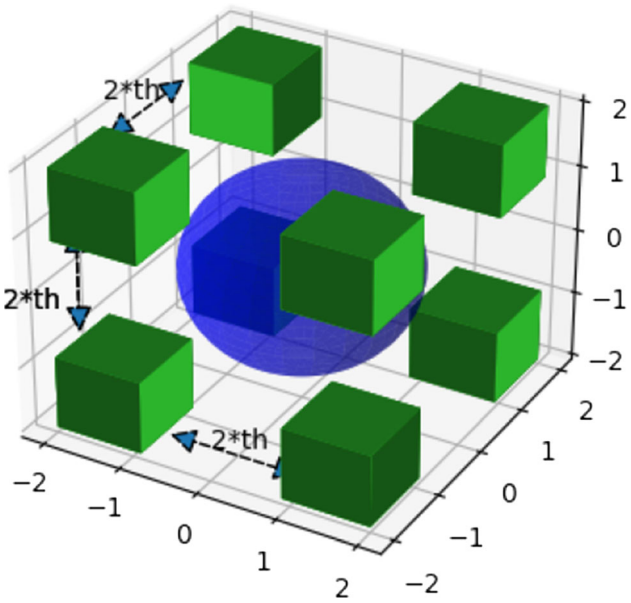

Fig. 13 CelebA Sectors seem to be external to the latent space of StyleGAN

The result is shown in Fig. 14. They are not perfect, but definitely interesting.

We also tested a few variants weighting the distance between latent variables according to their "information relevance", but we did not observe significant improvements.

Let us come to the mapping from the latent space of VAE to that of the StyleGAN. To train the transformation model (as usual, a single dense layer with no bias), we simply invert input and output of the previous network. After training, the mean squared error between w and $\hat{w}$ is around 0.029 with a standard deviation of 0.004. The mean squared error between $StyleGAN(w)$, and $StyleGAN(\hat{w})$ is 0.076 with standard deviation of 0.014. The results are visually really good, as can be visually checked in Fig. 15.

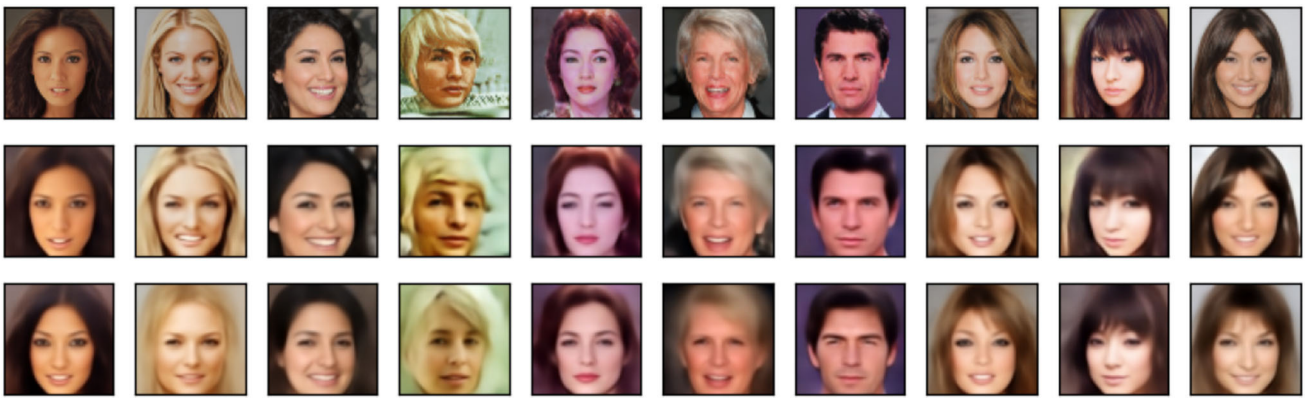

Fig. 14 Mapping from the W space of StyleGAN to the latent space of SVAE. In the first row we have sources, sampled by StyleGAN from $w \in W$. In the second row we have the SVAE reconstruction, starting from a suitably cropped and rescaled images (SVAE work at resolution 64): these images are the best possible approximation of the source images obtainable by SVAE. In the third row we show the output produced by the SVAE decoder after mapping each w in its latent space: results are very similar to those of the second row

Fig. 15 Mapping from the latent space of SVAE to the W space of StyleGAN. In the first row we have images generated by StyleGAN: $StyleGAN(w)$, for $w \in W$. In the second row we have their SVAE reconstructions, starting from suitably cropped and rescaled versions. Images in the third row are obtained by first encoding $StyleGAN(w)$ in the latent space of the SVAE, obtaining a latent representation z. This z is then linearly transformed to a vector $\hat{w} \in W$; the final image is $StyleGAN(\hat{w})$

8 Conclusions

In this article we addressed the problem of comparing the latent space of different generative models, defining transformations between them. Specifically, we proved that we can pass from a latent space to another by means of a simple *linear map* preserving most of the information. Hence, the organization of the latent space seems to be largely independent from

- The training process
- The network architecture
- The learning objective: GANs and VAEs share the same space

The result is original, surprising and largely unexpected; apparently, the latent space, if not artificially constrained with different objectives, seems to naturally organize itself in a way that is merely dependent from the data manifold. Of course, we expect that this "natural" structure can be altered in many different ways, e.g., through conditioning, which strongly impacts the latent structure, or via transformations like normalizing flows, explicitly aiming toward a strong regularization of the space. We also do not expect the two spaces Z and W of StyleGAN to be linearly related, since otherwise the long chain of eight dense layers between them would have no purpose.

Our result is full of implications from the point of view of representation learning and disentanglement. The fact that the latent space has a sort of implicit and native structure raises promising expectations about the possibility of learning features in a completely unsupervised way. Moreover, the recent observation [8, 66] that variations over a single semantical feature is a quasi-linear manifold in the latent space of generative models fits well with our empirical observations, opening interesting perspectives about the possibility of "porting" disentanglement between different spaces, and more generally, to better understand the issue in a more general framework.

The fact that the transformation between spaces is linear obviously permits its definition in terms of a small set of independent points of the same cardinality of the dimension of the latent space; this is what we call a *Support Set*. Locating these points in the two latent spaces is enough to define the map. In principle, any set of independent points could serve as a Support Set, but for robustness reasons, it seems preferable to choose points as apart as possible between each other. We described a possible approach for defining such a set, based on "sectors" in the space. This set is of interest in its own, as it is representative of the principal factors of variations in the dataset. Due to this fact, it also provides a natural benchmark to test the expressiveness of generative models.

This leads to an additional side contribution of our work: in contrast with the usual belief, StyleGAN trained on CelebA-HQ seems to have serious generative deficiencies: many images, in particular most of the images in our Support Set from CelebA, seem to lie outside the generative range of StyleGAN. In particular, as it is also evident in inversion results, the StyleGAN generative process is privileging standardization, strongly penalizing defects, oddities and eccentricities: the StyleGAN space is not a space for minorities.

This could be a cause for concern about CelebA-HQ. Not only it is computationally demanding, but one could also wander if it has statistical relevance: an assortment of 30K images in a space of dimension 3×2^{20} looks more like a collection of scattered points than a data manifold.

Our results also raise serious worries about the increasing use of generative techniques for data augmentation purposes. All generative techniques seem to have serious biases, privileging likelihood over diversity: using them for data augmentation may have no statistical significance. It is a bad practice that should be discouraged and deprecated.

As for future developments, most of the work just lies ahead. Here is a short, not-exhaustive list of possible topics:

- Test and hopefully confirm our mapping results on different datasets;
- Deepen the relationship between the field of disentanglement through suitable linear manipulations of the latent space;
- Define and test a Support Set for StyleGAN and Celeba-HQ;
- Investigate the possibility to improve the transformation with residual nonlinearities, and in that case study them;
- Better investigate and possibly find a remedy to the generative deficiencies of StyleGAN.

Appendix

A Models

In this section we briefly discuss the architecture of the generative models used for our experiments. In addition, we also largely experimented with StyleGAN, whose structure is discussed in Sect. 2.2. Two of the models are vanilla implementations of GAN and VAE with very similar structures; this was an intentional choice since we wished to evaluate the impact of the objective function independently from the network architecture.

A.1 Vanilla GAN structure

The structure of the discriminator is as follows:

1. A Convolutional layer going from an input of size (64, 64, 3) with stride $s = 2$, same padding, ReLU activation, kernel size $k = 4$ and 128 channels, followed by a Leaky ReLU layer with $\alpha = 0.2$ for regularization;
2. A Convolutional layer as in 1. but with 256 channels, followed by another leaky ReLU;
3. A Convolutional layer as in 1. but with 512 channels, followed by another leaky ReLU;
4. A Dropout layer with $\alpha = 0.2$ for GAN regularization;
5. A Dense layer outputting a single value, which is the confidence the discriminator has that its input image is real.

The structure of the generator is instead the following:

1. A Dense layer going from L to size (8, 8, 16);
2. A Transposed Convolutional layer with stride $s = 2$, same padding, ReLU activation, kernel size $k = 4$ and 128 channels, followed by a Leaky ReLU layer with $\alpha = 0.2$ for regularization;
3. A Transposed Convolutional layer as in 1. but with 256 channels, followed by another leaky ReLU;
4. A Transposed Convolutional layer as in 1. but with 512 channels, followed by another leaky ReLU;
5. A Convolutional layer with 3 channels, kernel size $k = 5$, sigmoid activation and same padding, thus producing a (64, 64, 3) output.

We also implemented a re-coder, for GAN inversion, with an essentially symmetric structure.

A.2 Vanilla VAE structure

Our VAEs use a balancing γ factor for its two loss components, which are the KL divergence and the

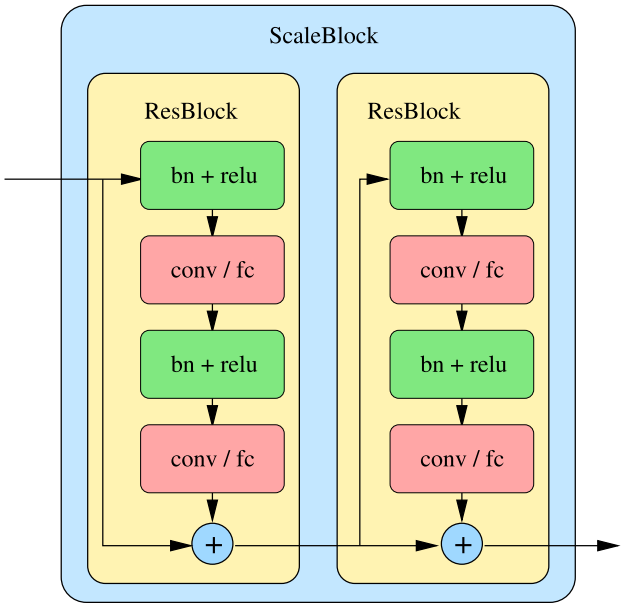

Fig. 16 Scale Block: a Scale Block is a sequence of Residual Blocks intertwined with residual connections. A Residual Block alternates BatchNormalization layers, nonlinear units and convolutions

reconstruction error, as suggested in [31] in order to improve variability and reduce blurriness.

The structure of the encoder is as follows:

1. A Convolutional layer going from an input of size (64, 64, 3) with stride $s = 2$, same padding, ReLU activation, kernel size $k = 4$ and 128 channels, followed by a Leaky ReLU layer with $\alpha = 0.2$ for regularization;
2. A Convolutional layer as in 1. but with 256 channels, followed by another leaky ReLU;
3. A Convolutional layer as in 1. but with 512 channels, followed by another leaky ReLU;
4. A Dropout layer with $\alpha = 0.2$ for GAN regularization;
5. Two separate Dense layers corresponding to the mean and variance vectors of the inference distribution $q(z|x)$ with size L, plus a third non-trainable layer which performs the sampling.

The structure of the decoder is the same as the structure of the GAN generator.

A.3 SVAE structure

Split-VAE (SVAE) [11] is a simple architectural variation of a traditional VAE where the output $\hat{x}$ is computed as a weighted sum

$$\hat{x} = \sigma \odot \hat{x_1} + (1 - \sigma) \odot \hat{x_2}$$

of two generated images $\hat{x_1}, \hat{x_2}$, and a *learned* compositional map σ. The splitting structure facilitate the synthesis

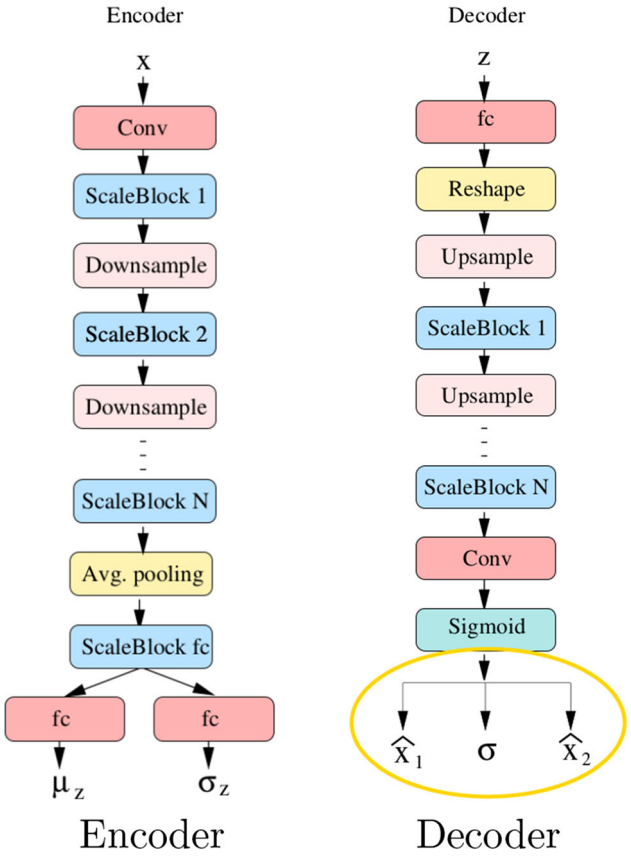

Fig. 17 Encoder: the input is progressively downsampled via convolutions, preceded by Scale Blocks. At the final scale, a global average pooling layer extract features that are further processed via dense layers to compute mean and variance for latent variables. Decoder: the decoder is essentially symmetric. A SVAE only differs in the final layer (circled in the picture): instead of directly producing $\hat{x}$, it produces two images $\hat{x_1}$ and $\hat{x_2}$ and a compositional map σ, defining $\hat{x} = \sigma \odot \hat{x_1} + (1 - \sigma) \odot \hat{x_2}$

of uncorrelated latent features, usually permitting to work with latent spaces of higher dimension.

For the implementation of the encoder and the decoder we adopted a ResNet-like architecture derived from [37] that we already used in previous works [31, 34]. The basic component, used both for encoding and decoding is a Scale-block, described in Fig. 16.

Encoder and decoder are essentially alternations of ScaleBlocks and downsampling/upsampling layers, as described in Fig. 17.

B Labels for the support set

In this section we give the list of labels for elements in the support set that we used for our experiments. The precise set is not very relevant; other choices driven by the methodology described in Sect. 5.3 give similar results.

Fig. 18 Examples of images in the support set, in addition to those in Fig. 9

Due to memory limitations, we have been forced to restrict the investigation to the first 70000 images in the CelebA dataset. This is the full list (150 elements):

[30, 58, 298, 702, 842, 873, 1779, 1809, 1844, 2590, 2719, 3888, 4114, 4223, 4550, 5659, 5718, 6058, 6108, 6128, 6175, 6244, 6705, 6815, 7499, 7679, 8225, 9457, 11254, 11282, 12367, 13077, 13371, 13993, 14193, 15390, 15711, 15817, 16505, 17186, 17458, 18250, 18283, 18582, 19080, 19175, 19612, 22505, 22633, 23173, 23199, 23308, 23511, 24231, 26431, 27169, 28270, 28401, 28433, 29453, 30248, 30269, 30619, 31741, 31795, 31836, 31978, 32272, 32770, 32828, 33332, 33613, 33669, 34024, 35804, 35823, 35882, 35944, 36483, 36926, 37374, 37534, 37538, 37572, 37682, 38194, 38483, 38677, 39232, 39267, 39424, 39901, 40405, 41464, 42969, 43035, 43199, 44054, 44252, 44589, 44798, 45930, 46259, 46693, 48128, 48786, 48839, 49498, 50345, 52454, 52516, 52673, 52753, 52834, 53071, 53308, 54937, 56128, 56492, 56693, 57844, 57927, 57942, 58020, 58089, 58162, 58389, 58947, 60359, 61004, 61180, 61374, 61495, 61530, 61794, 61878, 63535, 63891, 64328, 64342, 64663, 65041, 66277, 66321, 66663, 68027, 68753, 69274, 69750, 69936]

As it is clear from Fig. 18 some images in the support set are a bit pathological: extreme poses, frequent use of accessories like hats and eyeglasses, strange illumination, etc. So the support set also provides a good test-bench to check (through inversion) the robustness and diversification of the generative model.

In the github repository we also provide a version of the Support Set with people without hats. If required, a more "comformist" Support Set can be easily derived by reducing the threshold constraint as in Sect. 5.3.

Acknowledgements We would like to thank Fabio Merizzi for many interesting discussions on the subject of this article.

Funding Open access funding provided by Alma Mater Studiorum - Università di Bologna within the CRUI-CARE Agreement.

Data Availability The training datasets can be found at CelebA-dataset and CelebAHQ-dataset The code relative to this work is available on Github in the following repository: https://github.com/asperti/We_love_latent_space. We also provide pre-trained weights that can be downloaded using suitable facilities.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Bengio Y, Courville AC, Vincent P (2013) Representation learning: A review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828. https://doi.org/10.1109/TPAMI.2013.50
2. Kim H, Mnih A (2018) Disentangling by factorising. In: Proceedings of the 35th International conference on machine learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp 2654–2663. http://proceedings.mlr.press/v80/kim18b.html
3. Locatello F, Bauer S, Lucic M, Rätsch G, Gelly S, Schölkopf B, Bachem O (2019) Challenging common assumptions in the unsupervised learning of disentangled representations. In: Proceedings of the 36th international conference on machine learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. proceedings of machine learning research, vol. 97, pp 4114–4124. http://proceedings.mlr.press/v97/locatello19a.html
4. van Steenkiste S, Locatello F, Schmidhuber J, Bachem O (2019) Are disentangled representations helpful for abstract visual reasoning? In: Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp 14222–14235. https://proceedings.neurips.cc/paper/2019/hash/bc3c4a6331a8a9950945a1aa8c95ab8a-Abstract.html
5. Locatello F, Bauer S, Lucic M, Rätsch G, Gelly S, Schölkopf B, Bachem O (2020) A commentary on the unsupervised learning of disentangled representations. In: The Thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, The thirty-Second

innovative applications of artificial intelligence conference, IAAI 2020, The Tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pp 13681–13684. AAAI Press. https://ojs.aaai.org/index.php/AAAI/article/view/7120
6. Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. In: 4th International conference on learning representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, conference track proceedings. http://arxiv.org/abs/1511.06434
7. Shen Y, Zhou B (2021) Closed-form factorization of latent semantics in gans. In: IEEE conference on computer vision and pattern recognition, CVPR 2021, Virtual, June 19-25, 2021, pp 1532–1540. https://openaccess.thecvf.com/content/CVPR2021/html/Shen_Closed-Form_Factorization_of_Latent_Semantics_in_GANs_CVPR_2021_paper.html
8. Shen Y, Yang C, Tang X, Zhou B (2022) Interfacegan: Interpreting the disentangled face representation learned by gans. IEEE Trans Pattern Anal Mach Intell 44(4):2004–2018. https://doi.org/10.1109/TPAMI.2020.3034267
9. Klys J, Snell J, Zemel R (2018) Learning latent subspaces in variational autoencoders. In: Advances in neural information processing systems 31: annual conference on neural information processing systems 2018, 2-8 December 2018, Montreal, Canada, pp 6445–6455
10. Li G, Liu Y, Wei X, Zhang Y, Wu S, Xu Y, Wong H-S (2021) Discovering density-preserving latent space walks in gans for semantic image transformations. In: Proceedings of the 29th ACM International conference on multimedia, pp 1562–1570
11. Asperti A, Bugo L, Filippini D (2022) Enhancing variational generation through self-decomposition. IEEE Access 10:67510–67520. https://doi.org/10.1109/ACCESS.2022.3185654
12. Ruthotto L, Haber E (2021) An introduction to deep generative modeling. GAMM-Mitteilungen 44(2):202100008
13. Oussidi A, Elhassouny A (2018) Deep generative models: Survey. In: 2018 International conference on intelligent systems and computer vision (ISCV), pp 1–8. https://doi.org/10.1109/ISACV.2018.8354080
14. van den Oord A, Kalchbrenner N, Kavukcuoglu K (2016) Pixel recurrent neural networks. In: Proceedings of the 33nd international conference on machine learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. JMLR workshop and conference proceedings, vol. 48, pp 1747–1756. JMLR.org. http://proceedings.mlr.press/v48/oord16.html
15. Chen X, Mishra N, Rohaninejad M, Abbeel P (2018) Pixelsnail: An improved autoregressive generative model. In: International conference on machine learning, pp 864–872. PMLR
16. Kingma DP, Dhariwal P (2018) Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems **31**
17. Dinh L, Sohl-Dickstein J, Bengio S (2016) Density estimation using real NVP. arXiv preprint arXiv:1605.08803
18. Voleti V, Finlay C, Oberman A, Pal C (2021) Multi-resolution continuous normalizing flows. arXiv preprint arXiv:2106.08462
19. Kobyzev I, Prince SJD, Brubaker MA (2021) Normalizing flows: An introduction and review of current methods. IEEE Transactions on Pattern Analysis and Machine Intelligence 43(11):3964–3979. https://doi.org/10.1109/tpami.2020.2992934. Institute of Electrical and Electronics Engineers (IEEE)
20. Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems 33:6840–6851
21. Du Y, Mordatch I (2019) Implicit generation and generalization in energy-based models. arXiv preprint arXiv:1903.08689
22. Song Y, Kingma DP (2021) How to train your energy-based models. CoRR **abs/2101.03288**. https://arxiv.org/abs/2101.03288
23. Brock A, Donahue J, Simonyan K (2018) Large Scale GAN Training for High Fidelity Natural Image Synthesis. https://doi.org/10.48550/ARXIV.1809.11096. https://arxiv.org/abs/1809.11096
24. Karras T, Aila T, Laine S, Lehtinen J (2018) Progressive growing of gans for improved quality, stability, and variation. In: 6th International conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net. https://openreview.net/forum?id=Hk99zCeAb
25. Kingma DP, Welling M (2014) Auto-encoding variational bayes. In: 2nd International conference on learning representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings
26. Razavi A, Van den Oord A, Vinyals O (2019) Generating diverse high-fidelity images with VQ-VAE-2. Advances in neural information processing systems **32**
27. Burda Y, Grosse RB, Salakhutdinov R (2016) Importance weighted autoencoders. In: 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. http://arxiv.org/abs/1509.00519
28. Kingma DP, Welling M (2019) An introduction to variational autoencoders. Found Trends Mach Learn 12(4):307–392. https://doi.org/10.1561/2200000056
29. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press. http://www.deeplearningbook.org
30. Zhai J, Zhang S, Chen J, He Q (2018) Autoencoder and its various variants. In: 2018 IEEE international conference on systems, Man, and Cybernetics (SMC), pp 415–419. https://doi.org/10.1109/SMC.2018.00080
31. Asperti A, Trentin M (2020) Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. IEEE Access 8:199440–199448. https://doi.org/10.1109/ACCESS.2020.3034828
32. Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S, Lerchner A (2017) beta-vae: Learning basic visual concepts with a constrained variational framework. In: 5th International conference on learning representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track proceedings. OpenReview.net. https://openreview.net/forum?id=Sy2fzU9gl
33. Burgess CP, Higgins I, Pal A, Matthey L, Watters N, Desjardins G, Lerchner A (2018) Understanding disentangling in β-vae. https://arxiv.org/abs/1804.03599
34. Asperti A, Evangelista D, Piccolomini EL (2021) A survey on variational autoencoders from a green AI perspective. SN Comput Sci 2(4):301. https://doi.org/10.1007/s42979-021-00702-9
35. van den Oord A, Vinyals O, Kavukcuoglu K (2017) Neural discrete representation learning. In: Advances in neural information Processing Systems 30: annual conference on neural information processing systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 6306–6315. https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html
36. Kingma DP, Salimans T, Józefowicz R, Chen X, Sutskever I, Welling M (2016) Improving variational autoencoders with inverse autoregressive flow. In: Advances in neural information processing systems 29: annual conference on neural information processing systems 2016, December 5-10, 2016, Barcelona, Spain, pp 4736–4744
37. Dai B, Wipf DP (2019) Diagnosing and enhancing vae models. In: Seventh international conference on learning representations (ICLR 2019), May 6-9, New Orleans
38. Gregor K, Danihelka I, Graves A, Rezende DJ, Wierstra D (2015) DRAW: A recurrent neural network for image generation. In: Proceedings of the 32nd International Conference on Machine

Learning, ICML 2015, Lille, France, 6-11 July 2015. JMLR workshop and conference proceedings, vol. 37, pp 1462–1471. JMLR.org. http://jmlr.org/proceedings/papers/v37/gregor15.html
39. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville AC, Bengio Y (2014) Generative adversarial nets. In: Advances in Neural Information Processing Systems 27: Annual conference on neural information processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pp 2672–2680. https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html
40. Goodfellow IJ (2017) NIPS 2016 tutorial: Generative adversarial networks. CoRR **abs/1701.00160**. https://arxiv.org/abs/1701.00160
41. Lucic M, Kurach K, Michalski M, Gelly S, Bousquet O (2018) Are gans created equal? A large-scale study. In: Advances in Neural Information Processing Systems 31: Annual conference on Neural information processing systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pp 698–707. https://proceedings.neurips.cc/paper/2018/hash/e46de7e1bcaaced9a54f1e9d0d2f800d-Abstract.html
42. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: Proceedings of the 34th international conference on machine learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp 214–223. PMLR. http://proceedings.mlr.press/v70/arjovsky17a.html
43. Mao X, Li Q, Xie H, Lau RYK, Wang Z, Smolley SP (2017) Least squares generative adversarial networks. In: IEEE international conference on computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pp. 2813–2821. IEEE computer society. https://doi.org/10.1109/ICCV.2017.304
44. Kodali N, Abernethy J, Hays J, Kira Z (2017) On convergence and stability of gans. arXiv preprint arXiv:1705.07215
45. Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P (2016) Infogan: Interpretable representation learning by information maximizing generative adversarial nets. Advances in neural information processing systems **29**
46. Zhu J-Y, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International conference on computer Vision, pp. 2223–2232
47. Zhang H, Goodfellow IJ, Metaxas DN, Odena A (2019) Self-attention generative adversarial networks. In: Proceedings of the 36th international conference on machine learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. proceedings of machine learning research, vol. 97, pp 7354–7363. PMLR. http://proceedings.mlr.press/v97/zhang19d.html
48. Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4401–4410
49. Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T (2020) Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 8110–8119
50. Karras T, Aittala M, Laine S, Härkönen E, Hellsten J, Lehtinen J, Aila T (2021) Alias-free generative adversarial networks. Adv Neural Inform Process Syst 34:852–863
51. Xia W, Zhang Y, Yang Y, Xue J-H, Zhou B, Yang M-H (2022) Gan inversion: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence
52. Perarnau G, Van De Weijer J, Raducanu B, Álvarez JM (2016) Invertible conditional gans for image editing. arXiv preprint arXiv:1611.06355
53. Bau D, Strobelt H, Peebles WS, Wulff J, Zhou B, Zhu J, Torralba A (2019) Semantic photo manipulation with a generative image prior. ACM Trans Graph 38(4):59–15911
54. Daras G, Odena A, Zhang H, Dimakis AG (2020) Your local gan: Designing two dimensional local attention mechanisms for generative models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14531–14539
55. Anirudh R, Thiagarajan JJ, Kailkhura B, Bremer P-T (2020) Mimicgan: Robust projection onto image manifolds with corruption mimicking. Inter J Comput Vision 128(10), 2459–2477. Springer
56. Creswell A, Bharath AA (2019) Inverting the generator of a generative adversarial network. IEEE Transct Neural Netw Learn Syst 30(7):1967–1974
57. Zhu J, Krähenbühl P, Shechtman E, Efros AA (2016) Generative visual manipulation on the natural image manifold. In: Computer vision - ECCV 2016 - 14th european conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V. Lecture Notes in computer Science, vol. 9909, pp 597–613. Springer. https://doi.org/10.1007/978-3-319-46454-1_36
58. Zhu J, Shen Y, Zhao D, Zhou B (2020) In-domain GAN inversion for real image editing. In: Computer vision - ECCV 2020 - 16th European conference, Glasgow, UK, August 23-28, 2020, proceedings, Part XVII. Lecture notes in computer Science, vol. 12362, pp 592–608. Springer. https://doi.org/10.1007/978-3-030-58520-4_35
59. Abdal R, Qin Y, Wonka P (2019) Image2stylegan: How to embed images into the stylegan latent space? In: 2019 IEEE/CVF International conference on computer vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pp 4431–4440. IEEE. https://doi.org/10.1109/ICCV.2019.00453
60. Collins E, Bala R, Price B, Susstrunk S (2020) Editing in style: Uncovering the local semantics of gans. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 5771–5780
61. Abdal R, Qin Y, Wonka P (2020) Image2stylegan++: How to edit the embedded images? In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8296–8305
62. Poirier-Ginter Y, Lessard A, Smith R, Lalonde J-F (2022) Overparameterization improves stylegan inversion. arXiv preprint arXiv:2205.06304
63. Alaluf Y, Tov O, Mokady R, Gal R, Bermano A (2022) Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In: Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition, pp 18511–18521
64. Sohn K, Lee H, Yan X (2015) Learning structured output representation using deep conditional generative models. In: advances in neural information processing systems 28: annual conference on neural information processing systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pp 3483–3491. https://proceedings.neurips.cc/paper/2015/hash/8d55a249e6baa5c06772297520da2051-Abstract.html
65. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on computational learning Theory, pp. 144–152
66. Li Z, Tao R, Wang J, Li F, Niu H, Yue M, Li B (2021) Interpreting the latent space of gans via measuring decoupling. IEEE transactions on artificial intelligence 2(1):58–70. IEEE
67. Winant D, Schreurs J, Suykens JAK (2021) Latent space exploration using generative kernel PCA. CoRR **abs/2105.13949**. https://arxiv.org/abs/2105.13949
68. Liu Z, Luo P, Wang X, Tang X (2015) Deep learning face attributes in the wild. In: Proceedings of international conference on computer vision (ICCV), pp 3730–3738

69. Asperti A (2019) Sparsity in variational autoencoders. In: Proceedings of the first international conference on advances in signal processing and artificial intelligence, ASPAI, Barcelona, Spain, 20-22 March 2019. http://arxiv.org/abs/1812.07238
70. Dai B, Wang Y, Aston J, Hua G, Wipf DP (2018) Connections with robust PCA and the role of emergent sparsity in variational autoencoder models. J Machine Learn Res 19:41–14142
71. Razavi A, van den Oord A, Poole B, Vinyals O (2019) Preventing posterior collapse with delta-vaes. In: 7th International conference on learning representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. https://openreview.net/forum?id=BJe0Gn0cY7
72. Alemi AA, Poole B, Fischer I, Dillon JV, Saurous RA, Murphy K (2018) Fixing a broken ELBO. In: Proceedings of the 35th international conference on machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, pp 159–168
73. Rosca M, Lakshminarayanan B, Mohamed S (2018) Distribution matching in variational inference. CoRR **abs/1802.06847**. https://arxiv.org/abs/1802.06847
74. Asperti A (2019) About generative aspects of variational autoencoders. In: Machine Learning, Optimization, and Data Science - 5th international conference, LOD 2019, Siena, Italy, September 10-13, 2019, Proceedings, pp 71–82
75. Tolstikhin IO, Bousquet O, Gelly S, Schölkopf B (2018) Wasserstein auto-encoders. In: 6th international conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, conference track proceedings. OpenReview.net. https://openreview.net/forum?id=HkL7n1-0b
76. Tomczak JM, Welling M (2018) VAE with a vampprior. In: International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain, pp 1214–1223
77. Bauer M, Mnih A (2019) Resampled priors for variational autoencoders. In: The 22nd International conference on artificial intelligence and statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan. proceedings of machine learning research, vol. 89, pp. 66–75. PMLR. http://proceedings.mlr.press/v89/bauer19a.html
78. Chen B, Tan W, Wang Y, Zhao G. Distinguishing between natural and gan-generated face images by combining global and local features. Chinese Journal of Electronics **31**(1):59–67
79. Frank J, Eisenhofer T, Schönherr L, Fischer A, Kolossa D, Holz T (2020) Leveraging frequency analysis for deep fake image recognition. In: Proceedings of the 37th International conference on machine learning, ICML 2020, 13-18 July 2020, virtual event. proceedings of machine learning research, vol. 119, pp. 3247–3258. PMLR. http://proceedings.mlr.press/v119/frank20a.html
80. Yu N, Davis L, Fritz M (2019) Attributing fake images to gans: Learning and analyzing GAN fingerprints. In: 2019 IEEE/CVF international conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pp. 7555–7565. IEEE. https://doi.org/10.1109/ICCV.2019.00765