International Journal of Computer Vision (2021) 129:1451–1466
https://doi.org/10.1007/s11263-020-01429-5

Semantic Hierarchy Emerges in Deep Generative Representations for Scene Synthesis

Ceyuan Yang[1] · Yujun Shen[1] · Bolei Zhou[1]

Received: 31 January 2020 / Accepted: 31 December 2020 / Published online: 10 February 2021

Abstract
Despite the great success of Generative Adversarial Networks (GANs) in synthesizing images, there lacks enough understanding of how photo-realistic images are generated from the layer-wise stochastic latent codes introduced in recent GANs. In this work, we show that highly-structured semantic hierarchy emerges in the deep generative representations from the state-of-the-art GANs like StyleGAN and BigGAN, trained for scene synthesis. By probing the per-layer representation with a broad set of semantics at different abstraction levels, we manage to quantify the causality between the layer-wise activations and the semantics occurring in the output image. Such a quantification identifies the human-understandable variation factors that can be further used to steer the generation process, such as changing the lighting condition and varying the viewpoint of the scene. Extensive qualitative and quantitative results suggest that the generative representations learned by the GANs with layer-wise latent codes are specialized to synthesize various concepts in a hierarchical manner: the early layers tend to determine the spatial layout, the middle layers control the categorical objects, and the later layers render the scene attributes as well as the color scheme. Identifying such a set of steerable variation factors facilitates high-fidelity scene editing based on well-learned GAN models without any retraining (code and demo video are available at https://genforce.github.io/higan).

Keywords Generative model · Scene understanding · Image manipulation · Representation learning · Feature visualization

1 Introduction

Success of deep neural networks stems from representation learning, which identifies the explanatory factors underlying the high-dimensional observed data (Bengio et al. 2013). Prior work has shown that many concept detectors spontaneously emerge in the deep representations trained for classification tasks (Zhou et al. 2015; Zeiler and Fergus 2014; Bau et al. 2017; Gonzalez-Garcia et al. 2018). For example, Gonzalez-Garcia et al. (2018) observes that networks for object recognition are able to detect semantic object parts, and Bau et al. (2017) confirms that representations from classifying images learn to detect different categorical concepts at different layers.

Analyzing the deep representations and their emergent structures gives insight into the generalization ability of deep features (Morcos et al. 2018) as well as the feature transferability across different tasks (Yosinski et al. 2014), but current efforts mainly focus on discriminative models (Zhou et al. 2015; Gonzalez-Garcia et al. 2018; Zeiler and Fergus 2014; Agrawal et al. 2014; Bau et al. 2017). Generative Adversarial Networks (GANs) (Goodfellow et al. 2014; Karras et al. 2017, 2019; Brock et al. 2018) are capable of mapping random noises to high-quality images, however, the nature of the learned generative representations and how a synthesized image is composed over different layers of the GAN generator remain much less explored.

It has been known that some internal units of deep models emerge as object detectors when trained to categorize scenes (Zhou et al. 2015). Representing and detecting objects that are most informative to a specific category provides an ideal solution for classifying scenes, like sofa and TV are rep-

Communicated by Jifeng Dai.

Ceyuan Yang and Yujun Shen are equal contribution in this work.

✉ Ceyuan Yang
yc019@ie.cuhk.edu.hk

✉ Bolei Zhou
bzhou@ie.cuhk.edu.hk

Yujun Shen
sy116@ie.cuhk.edu.hk

[1] Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China

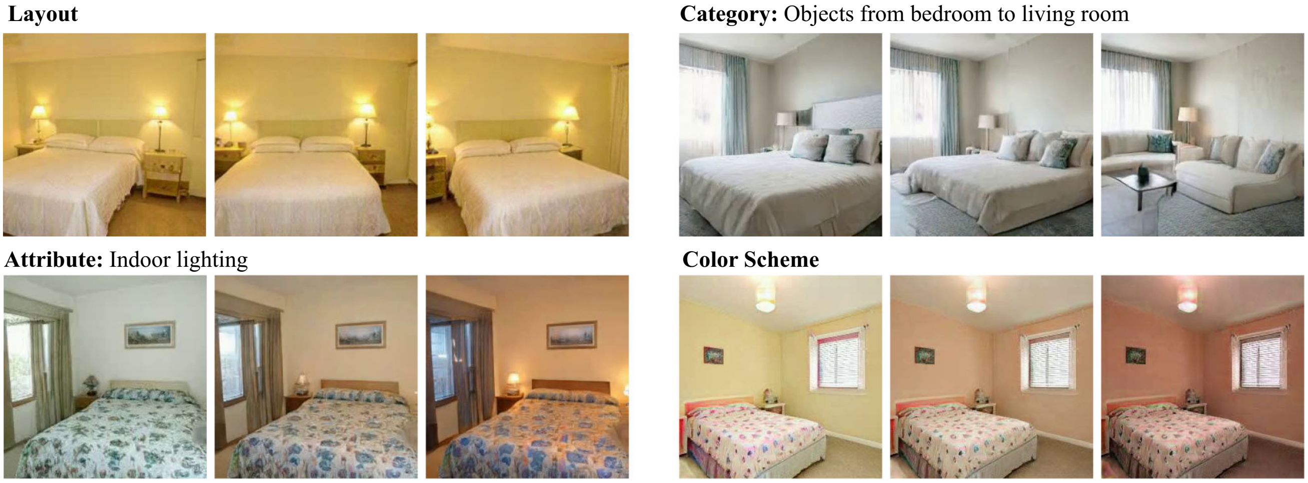

Layout
Category: Objects from bedroom to living room
Attribute: Indoor lighting
Color Scheme

Fig. 1 Scene manipulation results at four different abstraction levels, including *spatial layout*, *categorical objects*, *scene attributes*, and *color scheme*. For each tuple of images, the first is the raw synthesis, whilst the followings present the editing process (Color figure online)

resentative of the living room while bed and lamp are of the bedroom. However, synthesizing a scene requires far more complex knowledge. In particular, in order to produce realistic yet diverse scene images, a good generative representation is required to not only generate every individual object, but also decide the underlying room layout and render various scene attributes (e.g., the lighting condition). Bau et al. (2018) has found that some filters in the GAN generator correspond to the generation of some certain objects, however this analysis is only at the object level. Fully understanding how a scene image is synthesized requires examining the variation factors of scenes at multiple levels, i.e., from the layout level, the category level, to the attribute level. Recent GAN variants introduce layer-wise stochasticity to control the synthesis from coarse to fine (Karras et al. 2019; Brock et al. 2018; Shaham et al. 2019; Nguyen-Phuoc et al. 2019), however, how the variation factors originate from the generative representations layer by layer and how to quantify such semantic information still remain unknown.

In this paper, instead of designing new architectures for better synthesis, we examine the nature of the internal representations learned by the state-of-the-art GAN models. Starting with StyleGAN (Karras et al. 2019) as an example, we reveal that highly-structured semantic hierarchy emerges from the deep generative representations, which can well match the human-understandable scene variations from multiple abstraction levels, including layout, category, attribute, and color scheme. We first probe the per-layer representations of the generator with a broad set of visual concepts as candidates and then identify the most relevant variation factors for each layer. For this purpose, we propose a simply yet effective re-scoring technique to quantify the causality between the layer-wise activations and the semantics occurring in the output image. In particular, we find that the early layers determines the spatial layout, the middle layers compose the categorical objects, and the later layers render the attributes and color scheme of the entire scene. We also show that identifying such a set of steerable variation factors facilitates the versatile semantic image editing, as shown in Fig. 1. The proposed manipulation technique is applicable to other GAN variants, such as BigGAN (Brock et al. 2018) and PGGAN (Karras et al. 2017). More importantly, discovering the emerged hierarchy in the scene generation brings impacts on the research of scene understanding, which is one of the milestone tasks in computer vision and visual perception. Our work shows that the deep generative models 'draws' a scene like what humans do, i.e., drawing layout first, then representative objects, and finally fine-grained attributes and color schemes. It leads to many applications in scene understanding tasks such as scene editing, categorization, and parsing.

2 Related Work

2.1 Deep Representations from Classifying Images

Many attempts have been made to study the internal representations of deep models trained for classification tasks. Zhou et al. (2015) analyzed hidden units by simplifying the input image to see which context region gives the highest response, Simonyan et al. (2014) applied the back-propagation technique to compute the image-specific class saliency map, Bau et al. (2017) interpreted the hidden representations via the aid of the segmentation mask, Alain and Bengio (2016) trained independent linear probes to analyze the information separability among different layers. There are also some studies transferring the discriminative features to verify how learned representations fit with different datasets or tasks (Yosinski et al. 2014; Agrawal et al. 2014). In addition, reversing the feature extraction process by mapping a given representation back to the image space (Zeiler and Fergus 2014; Nguyen et al. 2016; Mahendran and Vedaldi 2015) also gives insight into how neural networks learn to distinguish different categories. However, these interpretation techniques developed for classification networks cannot be directly applied to generative models.

2.2 Deep Representations from Synthesizing Images

Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) advance the image synthesis significantly. Some recent models (Karras et al. 2017, 2019; Brock et al. 2018) are able to generate photo-realistic faces, objects, and scenes, making GANs applicable to real-world image editing tasks, such as image manipulation (Shen et al. 2018; Xiao et al. 2018a; Wang et al. 2018; Yao et al. 2018), image painting (Bau et al. 2018; Park et al. 2019), and image style transfer (Zhu et al. 2017; Choi et al. 2018). Despite such a great success, it remains uncertain what GANs have learned to produce diverse and realistic images. Radford et al. (2015) pointed out the vector arithmetic phenomenon in the underlying latent space of GAN, however, discovering what kinds of semantics exist inside a well-trained model and how these semantics are structured to compose high-quality images still remain unsolved. Bau et al. (2018) analyzed the individual units of the generator in GAN and found that they learn to synthesize informative visual contents such as objects and textures spontaneously. Besides, Jahanian et al. (2019) explored the steerability of GANs via distributional shift, and Goetschalckx et al. (2019) boosted the memorability of GANs by modulating the latent codes. Unlike them, our work *quantitatively* explores the emergence of *hierarchical* semantics inside the layer-wise generative representations. A closely relevant work, InterFaceGAN (Shen et al. 2020a), interpreted the latent space of GANs for diverse face editing. We *differ from InterFaceGAN* in the following three aspects. First, instead of examining the initial latent space, we study the layer-wise generative representations and reveal the semantic hierarchy learned for scene generation, which highly aligns with human perception. Second, scene images are far more complex than faces due to the large variety of scene categories as well as the objects inside, increasing the difficulty of interpreting scene synthesis models. Accordingly, unlike InterFaceGAN that clearly knows the target semantics in advance, we employ a broad set of 105 semantics to serve as candidates for further analysis. Third, we propose a re-scoring technique to *quantify* how a particular variation factor is relevant to different layers of the generator. This also enables layer-wise manipulation, resulting in a more precise control of scene editing.

2.3 Scene Manipulation

Editing scene images has been a long-standing task in the computer vision field. Laffont et al. (2014) defined 40 transient attributes and managed to transfer the appearance of a similar scene to the image for editing. Cheng et al. (2014) proposed verbal guided image parsing to recognize and manipulate the objects in indoor scenes. Karacan et al. (2016) learned a conditional GAN to synthesize outdoor scenes based on pre-defined layout and attributes. Bau et al. (2019) developed a technique to locally edit generated images based on the internal interpretation of GANs. Some other work (Liao et al. 2017; Zhu et al. 2017; Isola et al. 2017; Luan et al. 2017; Park et al. 2020) studied image-to-image translation and can be used to transfer the style of one scene to another. Besides, recent work (Abdal et al. 2019, 2020; Zhu et al. 2020) projected real images onto the latent space of a well-trained GAN generator and leveraged the GAN knowledge for image editing. Different from prior work, we achieve scene manipulation from multiple abstraction levels by reusing the knowledge from well-learned GAN models without any retraining.

2.4 Scene Understanding at Multiple Abstraction Levels

The abstraction levels of scene representations are inspired by prior literature on cognition studies of scene understanding. Oliva and Torralba (2001) proposed a computational model for a holistic representation (i.e., the shape of the scene) instead of individual objects or regions. Oliva and Torralba (2006) investigated that scene images are initially processed as a single entity and local information about objects and parts comes into play at a later stage of visual processing. Torralba and Oliva (2003) demonstrated how scene categories could provide the contextual information in the visual processing chain. Considering that scenes would have a multivariate attribute representation instead of simply a binary category membership, Patterson et al. (2014) advanced the scene understanding into more fine-grained representations, i.e., scene attributes. In this work, we discover the semantic hierarchy learned by deep generative networks and manage to align the aforementioned various concepts at different layers in a hierarchy.

3 Variation Factors in Generative Representations

3.1 Multi-Level Variation Factors for Scene Synthesis

Imagine an artist drawing a picture of the living room. The very first step is to choose a perspective and set up the room

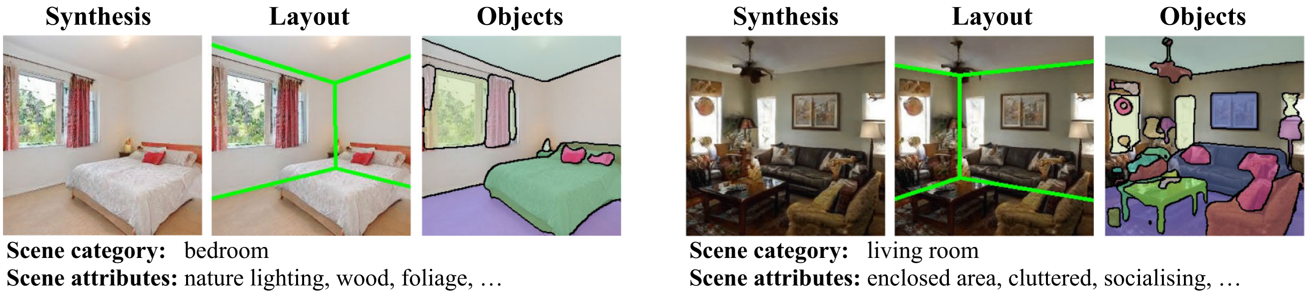

Synthesis
Layout
Objects
Scene category: bedroom
Scene attributes: nature lighting, wood, foliage, …
Synthesis
Layout
Objects
Scene category: living room
Scene attributes: enclosed area, cluttered, socialising, …

Fig. 2 Multi-level semantics extracted from two synthesized scenes (Color figure online)

layout. After the spatial structure is set, the next step is to add objects that typically occur in a living room, such as sofa and TV. Finally, the artist will refine the details of the picture with specified decoration styles, e.g., warm or cold, natural lighting or indoor lighting, etc.. The above process reflects how a human draws a scene by interpreting it from multiple abstraction levels. Meanwhile, given a scene image, we are able to extract multiple levels of semantics from it, as shown in Fig. 2. As a comparison, GANs follow a completely end-to-end training manner for synthesizing scenes without any prior knowledge about the drawing techniques or the concepts of layout and object. Even so, the trained GANs are able to produce photo-realistic scenes, which makes us wonder if the GANs have mastered any human-understandable drawing knowledge as well as the variation factors of scenes spontaneously.

3.2 Layer-Wise Generative Representations

In general, existing GANs take a randomly sampled latent code as the input and output an image synthesis. Such a mapping from the latent codes to the synthesized images is very similar to the feature extraction process in discriminative models. Accordingly, in this work, we treat the input latent code as the *generative representation* which will uniquely determine the appearance and properties of the output scene. On the other hand, the recent state-of-the-art GAN models [e.g., StyleGAN (Karras et al. 2019) and BigGAN (Brock et al. 2018)] introduce layer-wise stochasticity, as shown in Fig. 3 We therefore treat them as per-layer generative representations.

To explore how GANs are able to produce high-quality scene synthesis by learning multi-level variation factors as well as what role the generative representation of each layer plays in such generation process, this work aims at establishing the relationship between the variation factors and the generative representations. Karras et al. (2019) has already pointed out that the design of layer-wise stochasticity actually controls the synthesis from coarse to fine, however, what “coarse” and “fine” actually refer to still remains uncertain. To better align the variation factors with human perception, we separate them into four abstraction levels, including *layout*, *categorical objects*, *scene attributes*, and *color scheme*. We further propose a framework in Sect. 4 to quantify the causality between the input generative representations and the output variation factors. We surprisingly find that GANs synthesize a scene in a manner that is highly consistent with humans. Over all convolutional layers, GANs manage to organize these multi-level abstractions as a hierarchy. In particular, GAN constructs the spatial layout at the early stage, synthesizes category-specified objects at the middle stage, and renders the scene attribute and color scheme at the later stage.

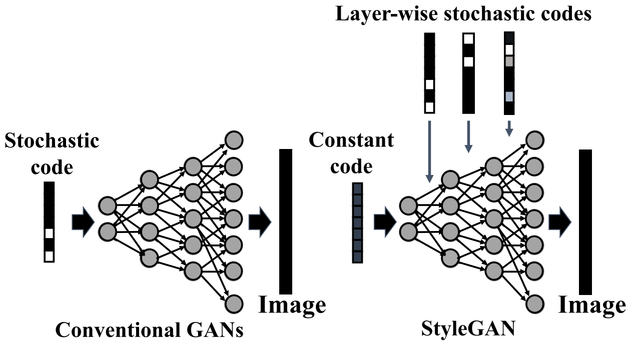


Fig. 3 Comparison between the conventional generator structure where the latent code is only fed into the very first layer and the generator in state-of-the-art GANs [e.g., StyleGAN (Karras et al. 2019) and BigGAN (Brock et al. 2018)] which introduce layer-wise stochasticity by feeding latent codes to all convolutional layers

4 Identifying the Emergent Variation Factors

As described in Sect. 3, we target at interpreting the latent semantics learned by scene synthesis models from four abstraction levels. Previous efforts on several scene understanding databases (Zhou et al. 2017; Xiao et al. 2010; Laffont et al. 2014; Patterson et al. 2014) enable a series of classifiers to predict scene attributes and categories. Besides, we also employ several classifiers focusing on layout detection (Zhang et al. 2019) and semantic segmentation (Xiao

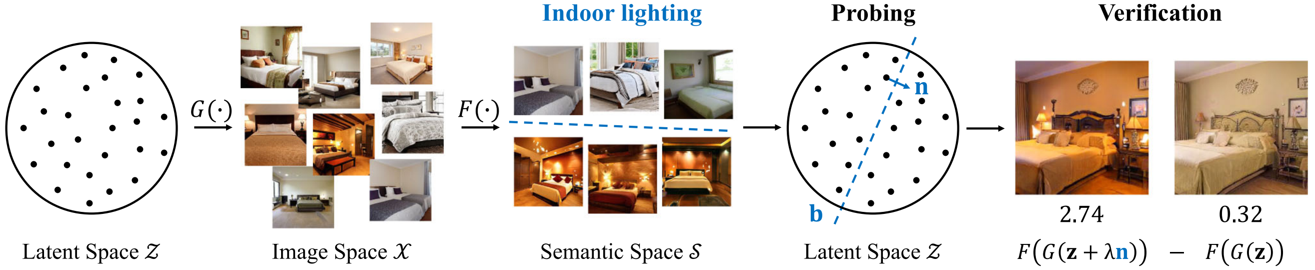

Indoor lighting
Probing
Verification
$G(\cdot)$
$F(\cdot)$
n
b
2.74
0.32
Latent Space $\mathcal{Z}$
Image Space $\mathcal{X}$
Semantic Space $\mathcal{S}$
Latent Space $\mathcal{Z}$
$F(G(\mathbf{z}+\lambda\mathbf{n})) - F(G(\mathbf{z}))$

Fig. 4 Pipeline of identifying the emergent variation factors in generative representation. By deploying a broad set of *off-the-shelf* image classifiers as scoring functions, $F(\cdot)$, we are able to assign a synthesized image with semantic scores corresponding to each candidate variation factor. For a particular concept, we learn a decision boundary in the latent space by considering it as a binary classification task. Then we move the sampled latent code towards the boundary to see how the semantic varies in the synthesis, and use a re-scoring technique to quantitatively verify the emergence of the target concept (Color figure online)

et al. 2018b). Specially, given an image, we are able to use these classifiers to get the response scores with respect to various semantics. However, only predicting the semantic labels is far from identifying the variation factors that GANs have captured from the training data. More concretely, among all the candidate concepts, not all of them are meaningful to a particular model. For instance, "indoor lighting" will never happen in outdoor scenes such as bridge and tower, which "enclosed area" is always true for indoor scenes such as bedroom and kitchen. Accordingly, we come up with a method to quantitatively identify the most relevant and manipulable variation factors that emerge inside the learned generative representation. Figure 4 illustrates the identification process which consists of two steps, i.e., probing (Sect. 4.1) and verification (Sect. 4.2). Such identification enables the diverse scene manipulation (Sect. 4.3). Note that we use the same approach as InterFaceGAN (Shen et al. 2020b) to get the latent boundary for each candidate in the probing process in Sect. 4.1.

4.1 Probing Latent Space

The generator of GAN, $G(\cdot)$, typically learns the mapping from latent space $\mathcal{Z}$ to image space $\mathcal{X}$. Latent vectors $\mathrm{z} \in \mathcal{Z}$ can be considered as the generative representations learned by GANs. To study the emergence of variation factors inside $\mathcal{Z}$, we need to first extract semantic information from z. For this purpose, we utilize the synthesized image, $\mathrm{x} = G(\mathrm{z})$, as an intermediate step and employ a broad set of image classifiers to help assign semantic scores for each sampled latent code z. Taking "indoor lighting" as an example, the scene attribute classifier is able to output the probability of how an input image looks like having indoor lighting, which we use as the semantic score. Recall that we divide scene representation into layout, object (category), and attribute levels, we introduce layout estimator, scene category recognizer, and attribute classifier to predict semantic scores from these abstraction levels respectively, forming a hierarchical semantic space $\mathcal{S}$. After establishing the mapping from the latent space $\mathcal{Z}$ to the semantic space $\mathcal{S}$, we search the decision boundary for each concept by treating it as a bi-classification problem, as shown in Fig. 4. Here, taking "indoor lighting" as an instance, the boundary separates the latent space $\mathcal{Z}$ to two sets, i.e., presence or absence of indoor lighting.

4.2 Verifying Manipulable Variation Factors

After probing the latent space with a broad set of candidate concepts, we still need to figure out which ones are most relevant to the generative model by acting as the variation factors. The key issue is how to define "relevance". We argue that if the target concept is manipulable from the latent space perspective (e.g., changing the indoor lighting status of the synthesized image via simply varying the latent code), the GAN model is considered as having captured such variation factor during training.

As mentioned above, we have already got a separation boundary for each candidate. Let $\{\mathrm{n}_i\}_{i=1}^C$ denote the normal vectors of these boundaries, where C is the total number of candidates. For a certain boundary, if we move a latent code z along its normal direction (positive), the semantic score should also increase correspondingly. Therefore, we propose to re-score the varied latent code to *quantify* how a variation factor is relevant to the target model for analysis. As shown in Fig. 4, this process can be formulated as

$$\Delta s_i = \frac{1}{K}\sum_{k=1}^{K} \max\left(F_i\left(G\left(\mathrm{z}^k + \lambda \mathrm{n}_i\right)\right) - F_i\left(G\left(\mathrm{z}^k\right)\right), 0\right), \tag{1}$$

where $\frac{1}{K}\sum_{k=1}^{K}$ stands for the average of K samples to make the metric more accurate. λ is a fixed moving step. To make this metric comparable among all candidates, all normal vectors $\{\mathrm{n}_i\}_{i=1}^C$ are normalized to the fixed norm 1 and λ is set

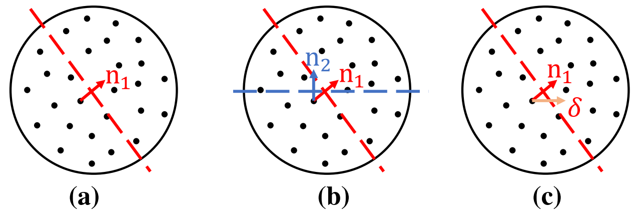


Fig. 5 Three types of manipulation: **a** *independent* manipulation; **b** *joint* manipulation; **c** *jittering* manipulation (Color figure online)

as 2. With this re-scoring technique, we can easily rank the score Δs_i among all C concepts to retrieve the most manipulable variation factors. Here, this technique is also performed layer by layer to identify the most relevant layers for each semantic.

4.3 Manipulation with Diversity

After identifying the semantics as well as the most adequate layers, we propose several manipulation approaches to control the generation process, as shown in Fig. 5. A simple and straightforward way, named *independent* manipulation, is to push the code z along the normal vector n_i of a certain semantic with a step length λ. The manipulated code $\mathrm{z}' \leftarrow \mathrm{z} + \lambda\mathrm{n}$ is then fed into the most relevant layers of the generator to produce a new image. A second way of manipulation enables scene editing with respect to more than one variation factor simultaneously. We call it *joint* manipulation. Taking two variation factors, with normal vector n_1 and n_2, as an example, the original code z is moved along the two directions simultaneously as $\mathrm{z}' \leftarrow \mathrm{z} + \lambda_1\mathrm{n}_1 + \lambda_2\mathrm{n}_2$. Here, λ_1 and λ_2 are step parameters which control the strength of the manipulation of these two semantics respectively. Besides the above two types of manipulation, we further propose to introduce randomness into the manipulation process to increase the diversity, namely *jittering* manipulation. The key idea is to slightly modulate the manipulation direction with a randomly sampled noise $\delta \sim \mathcal{N}(0, 1)$, bringing perturbation onto the main direction. It can be accordingly formulated as $\mathrm{z}' \leftarrow \mathrm{z} + \lambda\mathrm{n} + \delta$.

5 Experiments

In the generation process, the deep representation at each layer, especially for StyleGAN (Karras et al. 2019) and BigGAN (Brock et al. 2018), is actually directly derived from the projected latent code. Therefore, we consider the latent code as the *generative representation*. In addition, we conduct a detailed empirical analysis of the variation factors identified across the layers of the generators in GANs. Experimental results suggest that the hierarchy of variation factors emerges in the deep generative representations as a result of learning to synthesize scenes.

Table 1 Description of the StyleGAN models trained on different categories

Scene category	Type	Training number	FID↓
Bedroom (official)	Indoor	3 M	2.65
Living room	Indoor	1.3 M	5.16
Kitchen	Indoor	1 M	5.06
Restaurant	Indoor	626 K	4.03
Bridge	Outdoor	819 K	6.42
Church	Outdoor	126 K	4.82
Tower	Outdoor	708 K	5.99
Mixed	Indoor	500 K each	3.74

↓ Means the lower the better

The experimental section is organized as follows: Sect. 5.1 introduces our experimental details including generative models, training datasets and the *off-the-shelf* classifiers. Section 5.2 contains the layer-wise analysis on the state-of-the-art StyleGAN model (Karras et al. 2019), quantitatively and qualitatively verifying that the multi-level variation factors are encoded in the latent space. In Sect. 5.3, we explore the question on how GANs represent categorical information such as bedroom v.s. living room, revealing that GAN synthesizes the shared objects at some intermediate layers. By controlling their activations only, we can easily overwrite the category of the output image, e.g. turning bedroom into living room, while preserving its original layout and high-level attributes such as indoor lighting. Section 5.4 further shows that our approach can faithfully identify the most relevant attributes associated with a particular scene, facilitating semantic scene manipulation. Section 5.5 conducts the ablation studies on re-scoring technique and layer-wise manipulation to show the effectiveness of our approach.

5.1 Experimental Details

5.1.1 Generator Models

This work conducts experiments on state-of-the-art deep generative models for high-resolution scene synthesis, including StyleGAN (Karras et al. 2019), BigGAN (Brock et al. 2018), and PGGAN (Karras et al. 2017). Among them, PGGAN employs the conventional generator structure where the latent code is only fed into the very first layer. Differently, StyleGAN and BigGAN introduce layer-wise stochasticity by feeding latent codes to all convolutional layers as shown in Fig. 3. And our layer-wise analysis sheds light on why it is effective.

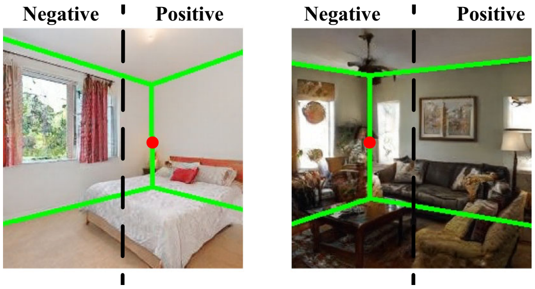


Fig. 6 The definition of layout for indoor scenes. Green lines represent the outline predicted by the layout estimator. The dashed line indicates the horizontal center, and the red point is the center point of the intersection line between two walls. The relative position between the vertical line and the center point is used to split the dataset (Color figure online)

5.1.2 Scene Categories

Among the mentioned generator models, PGGAN and StyleGAN are actually trained on LSUN dataset (Yu et al. 2015) while BigGAN is trained on Places dataset (Zhou et al. 2017). To be specific, LSUN dataset consists of 7 indoor scene categories and 3 outdoor scene categories, and Places dataset contains 10 million images across 434 categories. For PGGAN model, we use the officially released models, each of which is trained to synthesize scene within a individual category of LSUN dataset. For StyleGAN, only one model related to scene synthesis (i.e., bedroom) is released at this link. For a more thorough analysis, we use the official implementation to train multiple models on other scene categories, including both indoor scenes (living room, kitchen, restaurant) and outdoor scenes (bridge, church, tower). We also train a *mixed* model on the combination of images from bedroom, living room, and dining room with the same implementation. This model is specifically used for categorical analysis. For each StyleGAN model, Table 1 shows the category, the number of training samples, as well as the corresponding Fréchet inception distances (FID) (Heusel et al. 2017) which can reflect the synthesis quality to some extent. For BigGAN, we use the author's officially unofficial PyTorch BigGAN implementation to train a conditional generative model by taking category label as the constraint on Places dataset (Zhou et al. 2017). The resolution of the scene images synthesized by all of the above models is 256×256.

5.1.3 Semantic Classifiers

To extract semantic from synthesized images, we employ various *off-the-shelf* image classifiers to assign these images with semantic scores from multiple abstraction levels, including *layout*, *category*, *scene attribute*, and *color scheme*. Specifically, we use (1) a *layout estimator* (Zhang et al. 2019), which predicts the spatial structure of an indoor place, (2) a *scene category classifier* (Zhou et al. 2017), which classifies a scene image to 365 categories, and (3) an *attribute predictor* (Zhou et al. 2017), which predicts 102 pre-defined scene attributes in SUN attribute database (Patterson et al. 2014). We also extract color scheme of a scene image through its hue histogram in HSV space. Among them, the category classifier and attribute predictor can directly output the probability of how likely an image belongs to a certain category or how likely an image has a particular attribute. As for the layout estimator, it only detects the outline structure of an indoor place, shown in Fig. 6.

5.1.4 Semantic Probing and Verification

Given a well-trained GAN model for analysis, we first generate a collection of synthesized scene images by randomly sampling N latent codes (5,00,000 in practice). And then, the aforementioned image classifiers are used to assign semantic scores for each visual concept. It is worth noting that we use the relative position between image horizontal center and the intersection line of two walls to quantify layout, as shown in Fig. 6. After that, for each candidate, we select 2000 images with the highest response as positive samples, and another 2000 with the lowest response as negative ones. In particular, living room and bedroom are treated as positive and negative for scene category respectively for the mixed model. A linear SVM is trained by treating it as a bi-classification problem (i.e., data is the sampled latent code while the label is binary indicating whether the target semantic appears in the corresponding synthesis or not) to get a linear decision boundary. Finally, we re-generate $K = 1000$ samples for semantic verification as described in Sect. 4.2.

5.2 Emerging Semantic Hierarchy

Humans typically interpret a scene in a hierarchy of semantics, from its layout, underlying objects, to the detailed attributes and the color scheme. Here the underlying objects refer to the set of objects most relevant to a specific category. This section shows that GAN composes a scene over the layers in a similar way with human perception. To enable analysis on layout and object, we take the *mixed* StyleGAN model trained on indoor scenes as the target model. StyleGAN (Karras et al. 2019) learns a more disentangled latent space $\mathcal{W}$ on top of the conventional latent space $\mathcal{Z}$. Specifically, for ℓ-th layer, $\mathrm{w} \in \mathcal{W}$ is linearly transformed to layer-wise transformed latent code $\mathrm{y}^{(\ell)}$ with $\mathrm{y}^{(\ell)} = \mathrm{A}^{(\ell)}\mathrm{w} + \mathrm{b}^{(\ell)}$, where $\mathrm{A}^{(\ell)}$, $\mathrm{b}^{(\ell)}$ are the weight and bias for style transformation respectively. We thus perform layer-wise analysis by studying $\mathrm{y}^{(\ell)}$ instead of z in Eq. (1).

To quantify the importance of each layer with respect to each variation factor, we use the re-scoring technique to identify the causality between the layer-wise generative representation $\mathrm{y}^{(\ell)}$ and the semantic emergence. The normalized

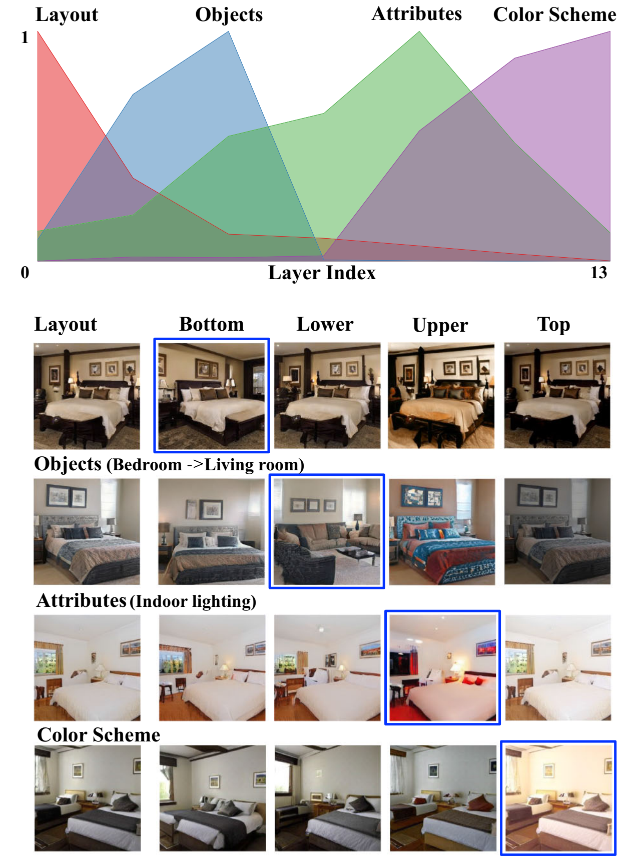
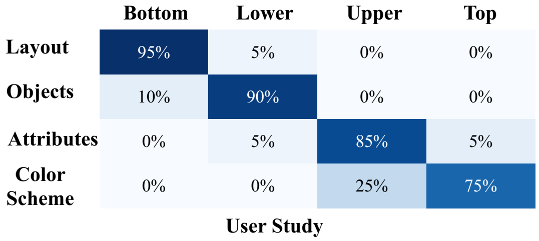


Fig. 7 Top: Four levels of visual abstractions emerge at different layers of StyleGAN. Vertical axis shows the normalized perturbation score Δs_i. Bottom: Layer-wise manipulation results. The first column is the original synthesis and the other columns are the manipulated images at layers from four different stages respectively. Blue boxes highlight the results from varying the latent code at the most proper layer for the target concept (Color figure online)



	Bottom	Lower	Upper	Top
Layout	95%	5%	0%	0%
Objects	10%	90%	0%	0%
Attributes	0%	5%	85%	5%
Color Scheme	0%	0%	25%	75%

Fig. 8 User study on how different layers correspond to variation factors from different abstraction levels (Color figure online)

score in the top Fig. 7 shows that the layers of the generator in GAN are specialized to compose semantics in a hierarchical manner: the bottom layers determine the layout, the lower layers and upper layers control category-level and attribute-level variations respectively, while color scheme is mostly rendered at the top. This is consistent with human perception. In StyleGAN model that is trained to produce 256 × 256 scene images, there are totally 14 convolutional layers. According to our experimental results, *layout*, *object (category)*, *attribute*, *color scheme* correspond to *bottom*, *lower*, *upper*, and *top* layers respectively, which are actually [0, 2), [2, 6), [6, 12) and [12, 14) layers.

To visually inspect the identified variation factors, we move the latent vector along the boundaries at different layers to show how the synthesis varies correspondingly. For example, given a boundary in regards to room layout, we vary the latent code towards the normal direction at bottom, lower, upper, and top layers respectively. The bottom of Fig. 7 shows the qualitative results for several concepts. The emerged variation factors follow a highly-structured semantic hierarchy, e.g., layout can be best controlled at the early stage while color scheme can only be changed at the final stage. Besides, varying latent code at the inappropriate layers may also change the image content, but the changing might be inconsistent with the desired output. For example, in the second row, modulating the code at bottom layers for category only leads to a random change in the scene viewpoint.

To better evaluate the manipulability across layers, we conduct a user study. We first generate 500 samples and manipulate them with respect to several concepts on different layers. For each concept, 20 users are asked to choose the most appropriate layers for manipulation. Specifically, in terms of a certain concept, we manipulate it at the bottom, lower, upper, top layers to produce a quadruplet. Users are asked to select single image with the desired change, unknowing the shuffled order of the quadruplet. The distribution of the choice for each abstraction level is recorded. Figure 8 shows the user study results, where most people think bottom layers best align with layout, lower layers control scene category, etc.. This is consistent with our observations in Fig. 7. It suggests that hierarchical variation factors emerge inside the generative representation for synthesizing scenes. and that our re-scoring method indeed helps identify the variation factors from a broad set of semantics.

Identifying the semantic hierarchy and the variation factors across layers facilitates semantic scene manipulation. We can simply push the latent code toward the boundary of the desired attribute at the appropriate layer. Figure 10a shows that we can change the decoration style (crude to glossy), the material of furniture (cloth to wood), or even the cleanliness (tidy to cluttered) respectively. Furthermore, hierarchical variation factors could be jointly manipulated. In Fig. 10b we simultaneously edit the room layout (rotating viewpoint) at early layers, scene category (converting bedroom to living room) at middle layers, and scene attribute (increasing indoor lighting) at later layers.

5.3 What Makes a Scene?

As mentioned above, GAN models for synthesizing scenes are capable of encoding hierarchical semantics inside the generative representation, i.e., from layout, object (category), to scene attribute and color scheme. One of the most noticeable properties is that the middle layers of GAN actually synthesize different objects for different scene categories. It raises the question of what makes a scene as living room rather than bedroom. Thus we further dive into the encoding of categorical information in GANs, to quantify how GAN interprets a scene category as well as how the scene category is transformed from an object perspective.

We employ the StyleGAN model trained on the mixture of bedroom, living room, and dining room, and then search the semantic boundary between every two categories. To extract the objects from the synthesized images, we apply a semantic segmentation model (Xiao et al. 2018b), which can segment 150 objects (TV, sofa, etc.) and stuff (ceiling, floor, etc.). Specifically, we first randomly synthesize 500 living room images, and then vary the corresponding latent codes towards the "living room-bedroom" boundary and "bedroom-dining room" boundary in turn. Segmentation masks of images before and after manipulation are obtained, as shown in Fig. 9. After tracking label mapping for each pixel via the image coordinate during the manipulation process, we are able to compute the statistics and observe how objects change along with transformed categories.

Figure 9 shows the objects mapping in the category transformation process. It clearly suggests that (1) when an image is manipulated among different categories, most of the stuff classes (e.g., ceiling and floor) remain the same, but some objects are mapped into other classes. For example, the sofa in living room is mapped to the pillow and bed in bedroom, and the bed in bedroom is further mapped to the table and chair in dining room. This phenomenon happens because sofa, bed, dining table and chair are distinguishable and discriminative objects for living room, bedroom, and dining room respectively. Thus, when category is transformed, the representative objects are supposed to change. (2) Some objects are shareable between different scene categories, and the GAN model is able to spot such property and learn to generate these shared objects across different classes. For example, the lamp in living room (on the left boundary of the image) still remains after the image is converted to bedroom, especially in the same position. (3) With the ability to learn object mapping as well as share objects across different classes, we are able to turn an unconditional GAN into a GAN that can control category. Typically, to make GAN produce images from different categories, class labels have to be fed into the generator to learn a categorical embedding, like BigGAN (Brock et al. 2018). Our result suggests an alternative approach (Fig. 10).

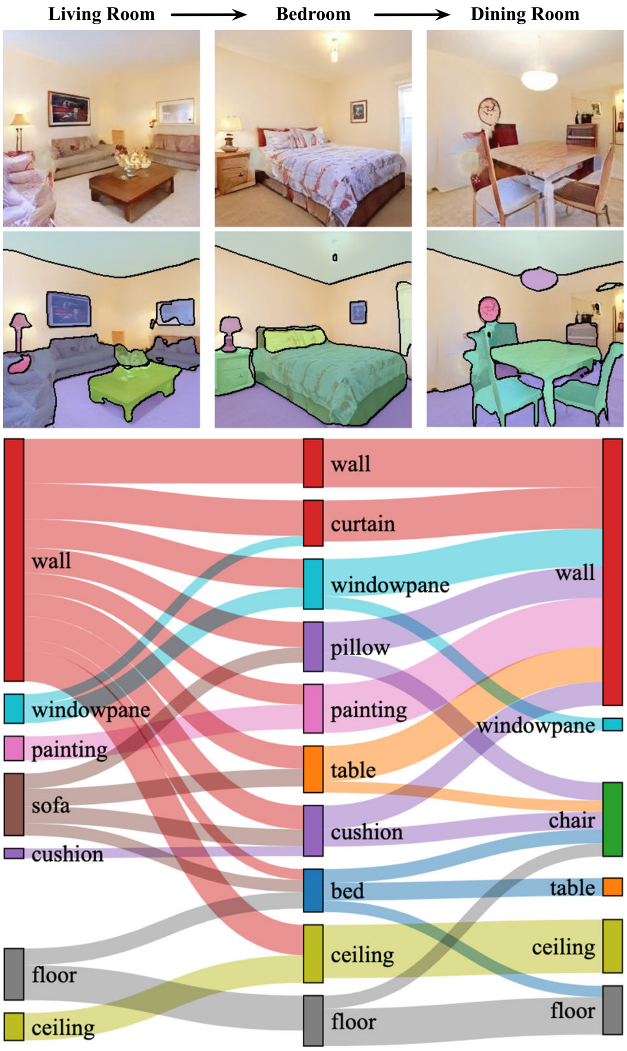


Fig. 9 Objects are transformed by GANs to represent different scene categories. The top shows that the object segmentation mask varies when manipulating a living room into a bedroom, and further into a dining room. The bottom visualizes the object mapping that appears during category transition, where pixels are counted only from object level instead of instance level. GANs can learn shared objects as well as the transformation of objects with similar appearance when trained to synthesize scene images from more than one category

5.4 Diverse Attribute Manipulation

5.4.1 Attribute Identification

The emergence of variation factors for scene synthesis depends on the training data. Here we apply our method to a collection of StyleGAN models, to capture a wide range of manipulable attributes out of the 102 scene attributes predefined in SUN attribute database (Patterson et al. 2014). Each StyleGAN in the collection is trained to synthesize scene images from a certain category, including both outdoor (bridge, church, tower) and indoor scenes (living room,

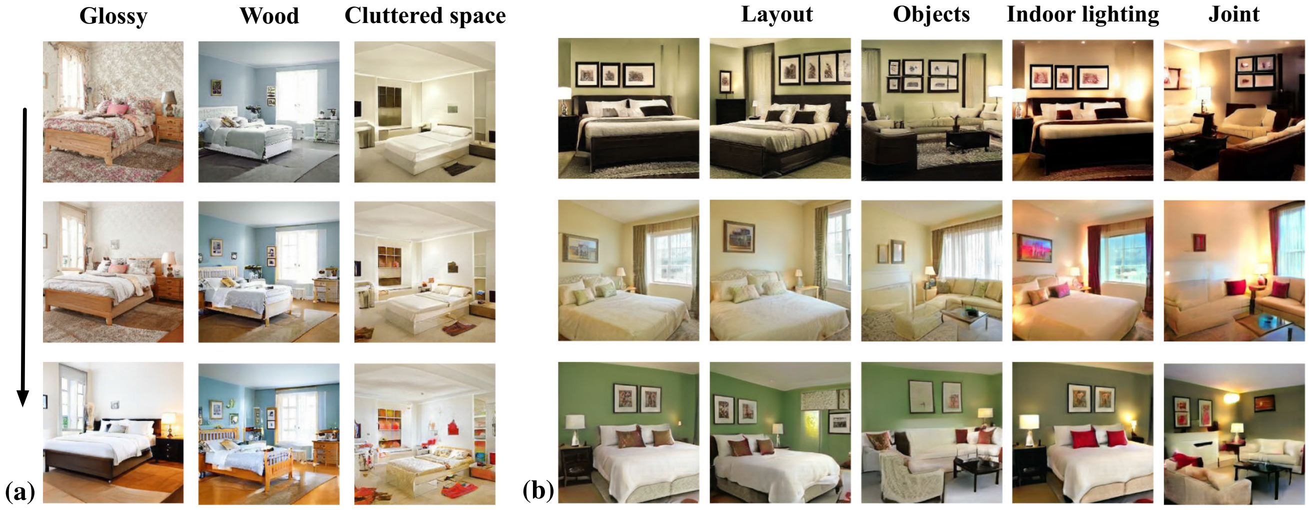


Fig. 10 **a** *Independent* attribute manipulation results on Upper layers. The middle row is the source images. We are able to both decrease (top row) and increase (bottom row) the variation factors in the images. **b** *Joint* manipulation results, where the *layout*, *objects* and *attribute* are manipulated at proper layers. The first column indicates the source images and the middle three columns are the independently manipulated images (Color figure online)

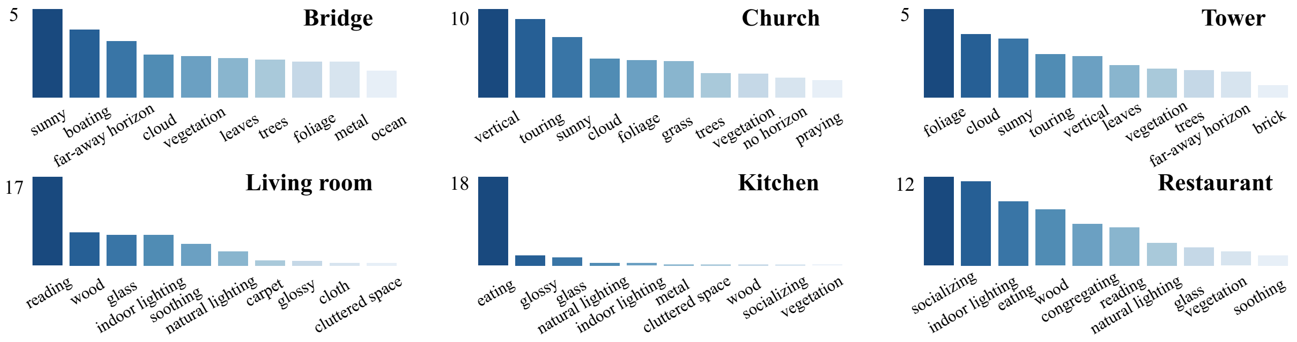


Fig. 11 Comparison of the top scene attributes identified in the generative representations learned by StyleGAN models for synthesizing different scenes. Vertical axis shows the perturbation score Δs_i (Color figure online)

kitchen). Figure 11 shows the top-10 relevant semantics to each model. It is seen that “sunny” has high scores on all outdoor categories, while “lighting” has high scores on all indoor categories. Furthermore, “boating” is identified for bridge model, “touring” for church and tower, “reading” for living room, “eating” for kitchen, and “socializing” for restaurant. These results are highly consistent with human understanding and perception, suggesting the effectiveness of the proposed quantification method.

5.4.2 Attribute Manipulation

Recall the three types of manipulation in Sect. 4.3: *independent* manipulation, *joint* manipulation, and *jittering* manipulation. We first conduct independent manipulation on 3 indoor and 3 outdoor scenes with the most relevant scene attributes identified with our approach. Figure 12 shows the results where the original synthesis (left image in each pair) is manipulated along the positive (right) direction. We can tell that the edited images are still with high quality and the target attributes indeed change as desired. We then jointly manipulate two attributes with bridge synthesis model as shown in Fig. 13. The central image of the 3 × 3 image grid is the original synthesis, the second row and the second column show the independent manipulation results with respect to “vegetation” and “cloud” attributes respectively, while other images on the four corners are the joint manipulation results. It turns out that we achieve good control of these two semantics and they seem to barely affect each other. However, not all variation factors show such strong disentanglement. From this point of view, our approach also provides a new metric to help measure the entanglement between two variation factors, which will be discussed in Sect. 6. Finally, we evaluate the proposed *jittering* manipulation by introducing noise into the “cloud” manipulation . From Fig. 14, we observe that the newly introduced noise indeed increases the manipula-

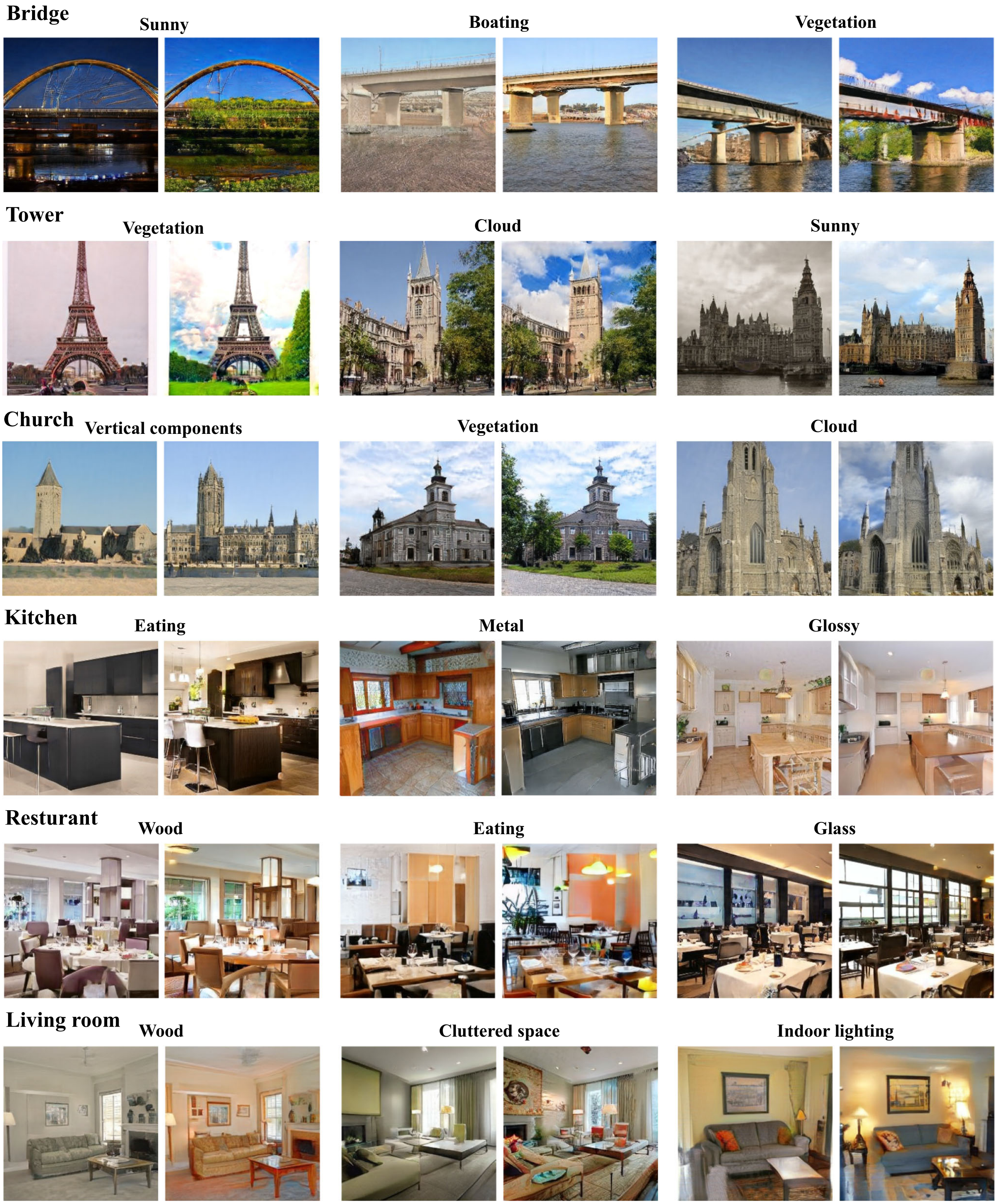


Fig. 12 *Independent* manipulation results on StyleGAN models trained for synthesizing indoor and outdoor scenes. In each pair of images, the first is the original synthesized sample and the second is the one after manipulating a certain semantic (Color figure online)

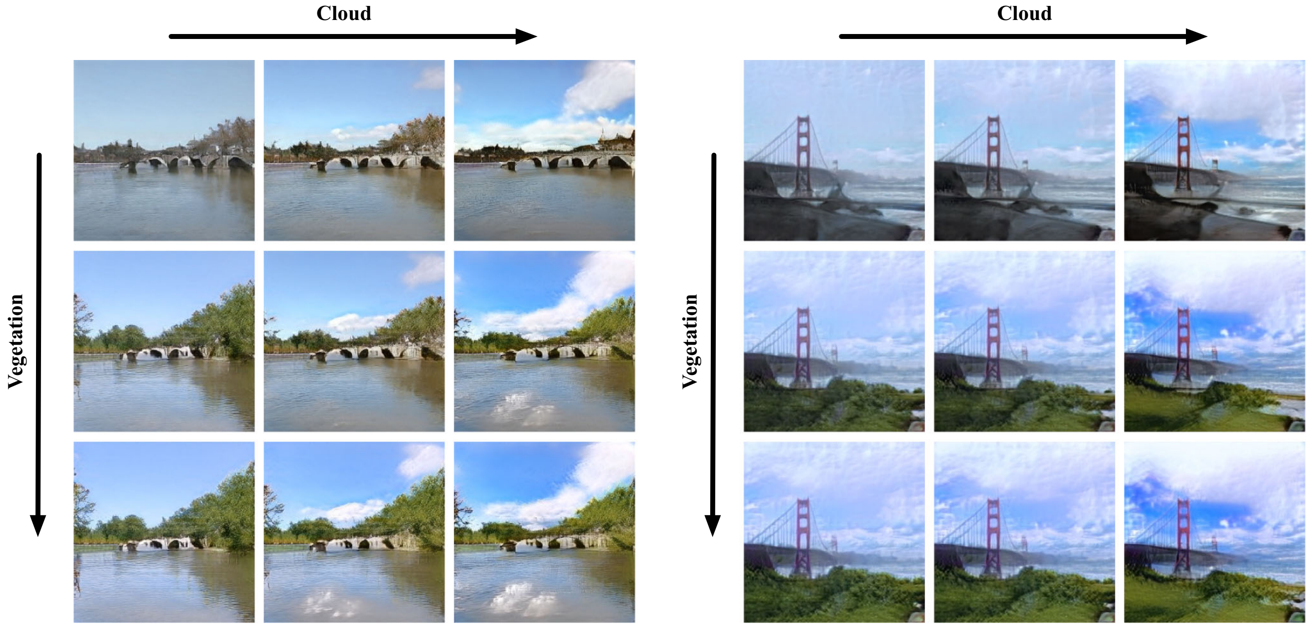


Fig. 13 *Joint* manipulation results along both *cloud* and *vegetation* boundaries with bridge synthesis model. Along the vertical and horizontal axis, the original synthesis (the central image) is manipulated with respect to *vegetation* and *cloud* attributes respectively (Color figure online)

tion diversity. It is interesting that the introduced randomness may not only affect the shape of added cloud, but also change the appearance of the synthesized tower. But both cases keep the primary goal, which is to edit the cloudiness.

5.5 Ablation Studies

5.5.1 Re-scoring Technique

Before performing the proposed re-scoring technique, we have two more steps, which are (1) assigning semantic scores for synthesized samples, and (2) training SVM classifiers to search semantic boundary. We would like to verify the necessity of the re-scoring technique in identifying manipulable semantics. Ablation study is conducted on the StyleGAN model trained for synthesizing bedrooms. As shown in Fig. 15, the left figure sorts the scene attributes by how many samples are labelled as positive ones, the middle figure sorts by the accuracy of the trained SVM classifiers, while the right figure sorts by our proposed quantification metric.

In left figure, “no horizon”, “man-made”, and “enclosed area” are attributes with highest percentage. However, all these three attributes are default properties of the bedroom and thus not manipulable. On the contrary, with the re-scoring technique for verification, our method successfully filters out these invariable candidates and reveals more meaningful semantics, like “wood” and “indoor lighting”. In addition, our method also manages to identify some less frequent but actually manipulable scene attributes, such as “cluttered space”.

In the middle figure, almost all attributes get similar scores, making them indistinguishable. Actually, even the worst SVM classifier (i.e., “railroad”) achieves 72.3% accuracy. That is because even some variation factors are not encoded in the latent representation (or say, not manipulable), the corresponding attribute classifier still assigns synthesized images with different scores. Training SVM on these inaccurate data can also result in a separation boundary, even it is not expected as the target concept. Therefore, only relying on the SVM classifier is not enough to detect relevant variation factors. By contrast, our method pays more attention to the score modulation after varying the latent code, which is not biased by the initial response of attribute classifier or the performance of SVM. As a result, we are able to thoroughly yet precisely detect the variation factors in the latent space from a broad candidate set.

5.5.2 Layer-Wise Manipulation

To further validate the emergence of semantic hierarchy, we make ablation study on layer-wise manipulation with StyleGAN model. First, we select “indoor lighting” as the target semantic, and vary the latent code only on upper (attribute-relevant) layers *v.s.* on all layers. We can easily tell from Fig. 16 that when manipulation “indoor lighting” at all layers, the objects inside the room are also changed. By contrast, manipulating latent codes only at attribute-relevant layers can

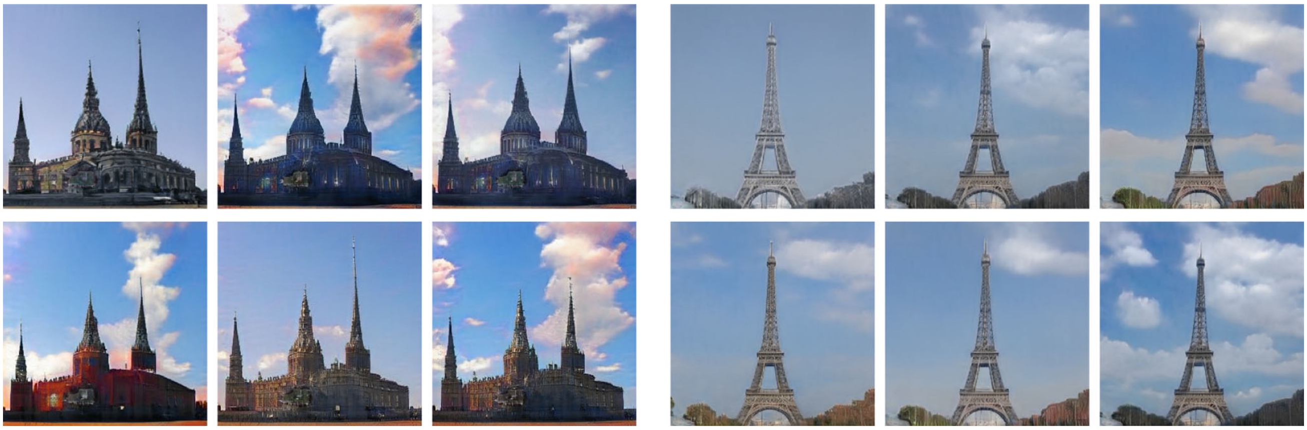


Fig. 14 *Jittering* manipulation results with tower synthesis model for *cloud* attribute. Specifically, the movement in the latent space of synthesized image is disturbed. Thus, when the cloud appears, both the shape of added cloud and appearance of the generated tower change. The top left image of two samples is the original output while the rest are the results under jittering manipulation separately (Color figure online)

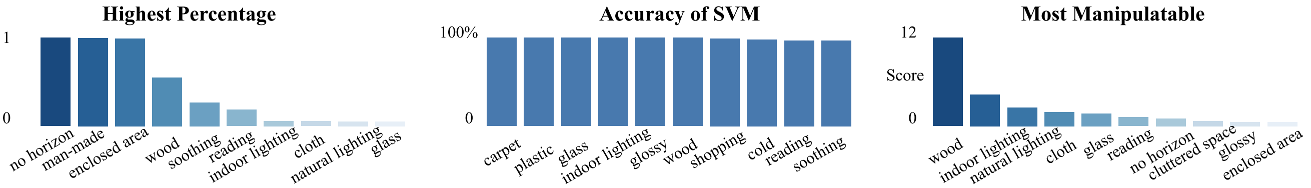


Fig. 15 Ablation study on the proposed re-scoring technique with StyleGAN model for bedroom synthesis. The left shows the percentage of scene attributes with the positive scores, the middle figure sorts by the accuracy of SVM classifiers, while the right figure sorts by our methods (Color figure online)

satisfyingly increase the indoor lighting without affecting other factors. Second, we select bottom layers as the target layers, and select boundaries from all four abstraction levels for manipulation. As shown in Fig. 17, no matter what level of semantics we choose, as long as the latent code is modified at bottom (layout-relevant) layers, only layout instead of all other semantics varies. These two experiments further verify our discovery about the emergence of the semantic hierarchy that the early layers tend to determine the spatial layout and configuration instead of other abstraction level semantics.

6 Discussions

6.1 Disentanglement of Semantics

Some variation factors we detect in the generative representation are more disentangled with each other than other semantics. Compared to the perceptual path length and linear separability described in Karras et al. (2019) and the cosine similarity proposed in Shen et al. (2020a), our work offers a new metric for disentanglement analysis. In particular, we move the latent code along one semantic direction and then check how the semantic scores of other factors change accordingly. As shown in Fig. 18a, when the spatial layout is modified, all attributes are barely affected, suggesting that GAN learns to disentangle layout-level semantic from attribute-level. However, there are also some scene attributes (from same abstraction level) entangling with each other. Taking Fig. 18c as an example, when modulating “indoor lighting”, “natural lighting” also varies. This is also aligned with human perception, further demonstrating the effectiveness of our proposed quantification metric. Qualitative results are also included in Fig. 18d–f.

6.2 Application to Other GANs

We further apply our method for two other GAN structures, i.e., PGGAN (Karras et al. 2017) and BigGAN (Brock et al. 2018). These two models are trained on LSUN dataset (Yu et al. 2015) and Places dataset (Zhou et al. 2017) respectively. Compared to StyleGAN, PGGAN feeds the latent vector only to the very first convolutional layer and hence does not support layer-wise analysis. But the proposed re-scoring method can still be applied to help identify manipulatable semantics, as shown in Fig. 19a. BigGAN is the state-of-the-art conditional GAN model that concatenates the latent vector with a class-guided embedding code before feeding it to the generator, and it also allows layer-wise analysis like StyleGAN. Figure 19b gives analysis results on BigGAN from

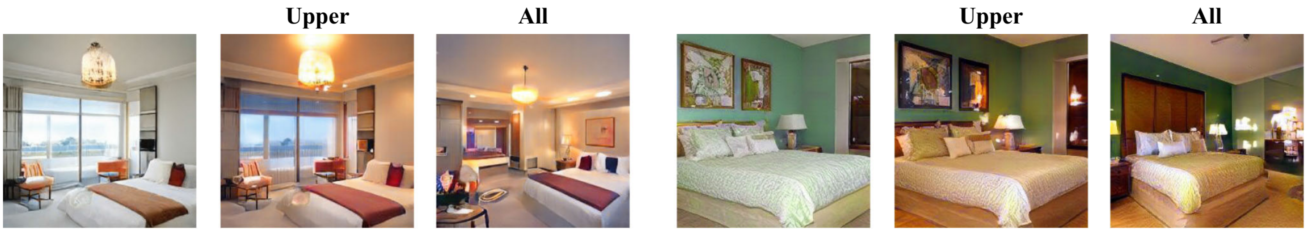


Fig. 16 Comparison results between manipulating latent codes at only upper (attribute-relevant) layers and manipulating latent codes at all layers with respect to *indoor lighting* on StyleGAN (Color figure online)

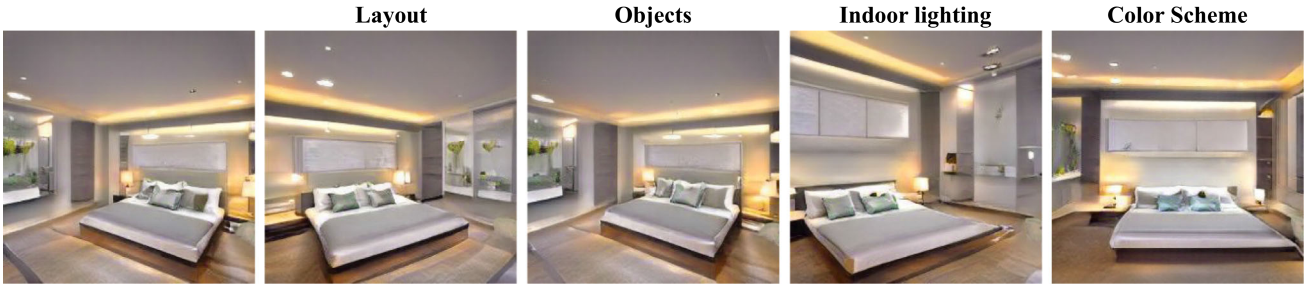


Fig. 17 Manipulation at the *bottom* layers in 4 different directions, along the directions of *layout*, *objects (category)*, *indoor lighting*, and *color scheme* on StyleGAN (Color figure online)

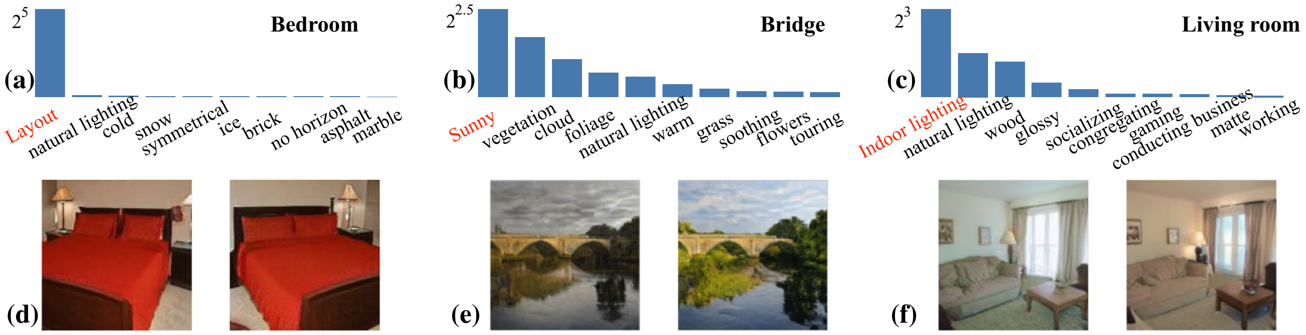


Fig. 18 **a–c** Quantitative effects on scene attributes (already sorted). Vertical axis shows the perturbation score Δs_i in log scale. **d–f** Qualitative results also show the effect when varying the most relevant factor (Color figure online)

attribute level, where we can tell that scene attribute can be best modified at upper layers compared to lower layers or all layers. As for BigGAN model with 256 × 256 resolution, there are total 12 convolutional layers. As the category information is already encoded in the “class” code, we only separate the layers to two groups, which are *lower* (bottom 6 layers) and *upper* (top 6 layers). Meanwhile, the quantitative curve shows the consistent result with the discovery on StyleGAN as in Fig. 7a. These results demonstrate the generalization ability of our approach as well as the emergence of manipulatable factors in other GANs.

6.3 Limitation

There are several limitations for future improvement. (1) More thorough and precise off-the-shelf classifiers: although we collect as many visual concepts as possible and summarize them into various levels by prior work, such as layout in Oliva and Torralba (2001), category in Torralba and Oliva (2003), and attribute in Patterson et al. (2014), such classifiers remain to be improved together with the development of scene understanding. In case the defined broad set of semantics is not enough, we could further enlarge the dictionary following the stardard annotation pipeline in Zhou et al. (2017) and Patterson et al. (2014). In addition, such classifiers trained on the large-scale benchmark of scene understanding could be replaced by more powerful discriminative models to improve the accuracy. (2) Boundary search: for simplicity we only use the linear SVM for semantic boundary search. This limits our framework from interpreting the latent semantic subspace with more complex and nonlinear structure. (3) Generalization beyond scene understanding: the main purpose of this work is to interpret scene-related GANs, which is a challenging task considering the large diversity of scene

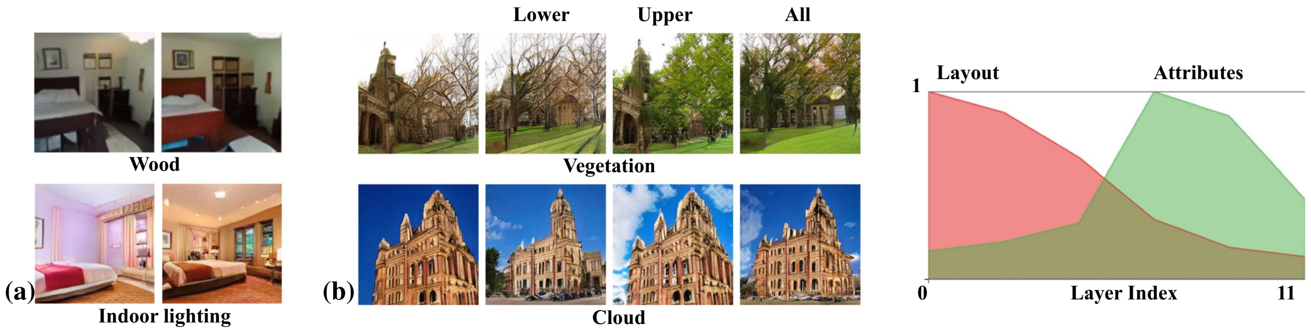


Fig. 19 **a** Some variation factors identified from PGGAN (bedroom). **b** Layer-wise analysis on BigGAN from the attribute level (Color figure online)

images as well as the difficulty of scene understanding. However, these abstraction levels can be hard to generalize to other datasets beyond scenes. Even so, we believe that this work is still able to provide some insights on analyzing GAN models trained on other datasets. For example, for scene synthesis, we found that early layers control scene layout, which can be viewed as structural information, such as rotation. Accordingly, we can fairly generalize that the early layers of face synthesis models control the face pose and the early layers of car synthesis models control the car orientation.

7 Conclusion

In this paper, we show the emergence of highly-structured variation factors inside the deep generative representations learned by GANs with layer-wise stochasticity. In particular, the GAN model spontaneously learns to set up layout at early layers, generate categorical objects at middle layers, and render scene attribute and color scheme at later layers when trained to synthesize scenes. A re-scoring method is proposed to quantitatively identify the manipulatable semantic concepts within a well-trained model, enabling photo-realistic scene manipulation. We will explore to extend this manipulation capability of GANs for real image editing in future work.

Acknowledgements This work is supported by Early Career Scheme (ECS) through the Research Grants Council (RGC) of Hong Kong under Grant No.24206219 and CUHK FoE RSFS Grant (No. 3133233).

References

Abdal, R., Qin, Y., & Wonka, P. (2019). Image2stylegan: How to embed images into the stylegan latent space? In: *International conference on computer vision* (pp. 4432–4441).

Abdal, R., Qin, Y., & Wonka, P. (2020). Image2stylegan++: How to edit the embedded images? In: *IEEE conference on computer vision and pattern recognition* (pp. 8296–8305).

Agrawal, P., Girshick, R., & Malik, J. (2014). Analyzing the performance of multilayer neural networks for object recognition. In: *European conference on computer vision* (pp. 329–344). Springer.

Alain, G., & Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. In: *International conference on learning representations workshop*.

Bau, D., Strobelt, H., Peebles, W., Wulff, J., Zhou, B., Zhu, J.-Y., & Torralba, A. (2019). Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics, 38(4),* 59.

Bau, D., Zhou, B., Khosla, A., Oliva, A., & Torralba, A. (2017). Network dissection: Quantifying interpretability of deep visual representations. In: *IEEE conference on computer vision and pattern recognition* (pp. 6541–6549).

Bau, D., Zhu, J. Y., Strobelt, H., Zhou, B., Tenenbaum, J. B., Freeman, W. T., & Torralba, A. (2018). Gan dissection: Visualizing and understanding generative adversarial networks. In: *International conference on learning representations*.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. In: *International conference on learning representations*.

Cheng, M. M., Zheng, S., Lin, W. Y., Vineet, V., Sturgess, P., Crook, N., et al. (2014). Imagespirit: Verbal guided image parsing. *ACM Transactions on Graphics*, *34*(1), 1–11.

Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S., & Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: *IEEE conference on computer vision and pattern recognition* (pp. 8789–8797).

Goetschalckx, L., Andonian, A., Oliva, A., & Isola, P. (2019). Ganalyze: Toward visual definitions of cognitive image properties. In: *Proceedings of the IEEE international conference on computer vision* (pp. 5744–5753).

Gonzalez-Garcia, A., Modolo, D., & Ferrari, V. (2018). Do semantic parts emerge in convolutional neural networks? *International Journal of Computer Vision*, *126*(5), 476–494.

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

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: *Advances in neural information processing systems* (pp. 6626–6637).

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In: *IEEE conference on computer vision and pattern recognition* (pp. 1125–1134).

Jahanian, A., Chai, L., & Isola, P. (2019). On the"steerability" of generative adversarial networks. In: *International conference on learning representations*.

Karacan, L., Akata, Z., Erdem, A., & Erdem, E. (2016) Learning to generate images of outdoor scenes from attributes and semantic layouts. arXiv preprint arXiv:1612.00215.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. In: *International conference on learning representations*.

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In: *IEEE conference on computer vision and pattern recognition* (pp. 4401–4410).

Laffont, P. Y., Ren, Z., Tao, X., Qian, C., & Hays, J. (2014). Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics*, *33*(4), 1–11.

Liao, J., Yao, Y., Yuan, L., Hua, G., & Kang, S. B. (2017). Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics*, *36*(4), 120.

Luan, F., Paris, S., Shechtman, E., Bala, K. (2017) Deep photo style transfer. In: *IEEE conference on computer vision and pattern recognition* (pp. 4990–4998).

Mahendran, A., & Vedaldi, A. (2015). Understanding deep image representations by inverting them. In: *IEEE conference on computer vision and pattern recognition* (pp. 5188–5196).

Morcos, A. S., Barrett, D. G., Rabinowitz, N. C., & Botvinick, M. (2018). On the importance of single directions for generalization. In: *International conference on learning representations*.

Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., & Clune, J. (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In: *Advances in neural information processing systems* (pp. 3387–3395).

Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., & Yang, Y. L. (2019) Hologan: Unsupervised learning of 3D representations from natural images. In: *International conference on computer vision* (pp. 7588–7597).

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, *42*(3), 145–175.

Oliva, A., & Torralba, A. (2006). Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, *155*, 23–36.

Park, T., Liu, M. Y., Wang, T. C., Zhu, J. Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In: *IEEE conference on computer vision and pattern recognition* (pp. 2337–2346).

Park, T., Zhu, J.-Y., Wang, O., Lu, J., Shechtman, E., Efros, A. A., & Zhang, R. (2020). Swapping autoencoder for deep image manipulation. In: *Advances in Neural Information Processing Systems.*

Patterson, G., Xu, C., Su, H., & Hays, J. (2014). The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, *108*(1–2), 59–81.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. In: *International conference on learning representations*.

Shaham, T. R., Dekel, T., & Michaeli, T. (2019). Singan: Learning a generative model from a single natural image. In: *International conference on computer vision* (pp. 4570–4580).

Shen, Y., Gu, J., Tang, X., & Zhou, B. (2020a). Interpreting the latent space of gans for semantic face editing. In: *IEEE conference on computer vision and pattern recognition* (pp. 9243–9252).

Shen, Y., Luo, P., Yan, J., Wang, X., & Tang, X. (2018). Faceid-gan: Learning a symmetry three-player gan for identity-preserving face synthesis. In: *IEEE conference on computer vision and pattern recognition* (pp. 821–830).

Shen, Y., Yang, C., Tang, X., & Zhou, B. (2020b). InterFaceGAN: Interpreting the disentangled face representation learned by GANs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2020.3034267.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In: *Workshop at international conference on learning representations*.

Torralba, A., & Oliva, A. (2003). Statistics of natural image categories. *Network: Computation in Neural Systems*, *14*(3), 391–412.

Wang, T. C., Liu, M. Y., Zhu, J. Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In: *IEEE conference on computer vision and pattern recognition* (pp. 8798–8807).

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba A (2010) Sun database: Large-scale scene recognition from abbey to zoo. In: *2010 IEEE computer society conference on computer vision and pattern recognition* (pp. 3485–3492). IEEE.

Xiao, T., Hong, J., & Ma, J. (2018) Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In: *European conference on computer vision* (pp. 168–184).

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., & Sun, J. (2018). Unified perceptual parsing for scene understanding. In: *Proceedings of the European conference on computer vision (ECCV)* (pp. 418–434).

Yao, S., Hsu, T. M., Zhu, J. Y., Wu, J., Torralba, A., Freeman, B., & Tenenbaum, J. (2018). 3D-aware scene manipulation via inverse graphics. In: *Advances in neural information processing systems* (pp. 1887–1898).

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In: *Advances in neural information processing systems* (pp. 3320–3328).

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., & Xiao, J. (2015) Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365.

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

Zhang, W., Zhang, W., & Gu, J. (2019). Edge-semantic learning strategy for layout estimation in indoor environment. *IEEE Transactions on Cybernetics*, *50*(6), 2730–2739.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2015). Object detectors emerge in deep scene cnns. In: *International conference on learning representations*.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2017). Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(6), 1452–1464.

Zhu, J., Shen, Y., Zhao, D., & Zhou, B. (2020). In-domain gan inversion for real image editing. In: *European conference on computer vision*.

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *International conference on computer vision* (pp. 2223–2232).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.