applied sciences

MDPI

Article

Unlocking Efficiency in Fine-Grained Compositional Image Synthesis: A Single-Generator Approach

Zongtao Wang and Zhiming Liu *

School of Computer & Information Science, Southwest University, Chongqing 400715, China; wztdream@email.swu.edu.cn
* Correspondence: zhimingliu88@swu.edu.cn

Abstract: The use of Generative Adversarial Networks (GANs) has led to significant advancements in the field of compositional image synthesis. In particular, recent progress has focused on achieving synthesis at the semantic part level. However, to enhance performance at this level, existing approaches in the literature tend to prioritize performance over efficiency, utilizing separate local generators for each semantic part. This approach leads to a linear increase in the number of local generators, posing a fundamental challenge for large-scale compositional image synthesis at the semantic part level. In this paper, we introduce a novel model called Single-Generator Semantic-Style GAN (SSSGAN) to improve efficiency in this context. SSSGAN utilizes a single generator to synthesize all semantic parts, thereby reducing the required number of local generators to a constant value. Our experiments demonstrate that SSSGAN achieves superior efficiency while maintaining a minimal impact on performance.

Keywords: compositional learning; generative adversarial network; image synthesis; disentanglement; efficiency

check for updates

Citation: Wang, Z.; Liu, Z. Unlocking Efficiency in Fine-Grained Compositional Image Synthesis: A Single-Generator Approach. *Appl. Sci.* **2023**, *13*, 7587. https://doi.org/10.3390/app13137587

Academic Editors: Seokwon Yeom and Emanuel Guariglia

Received: 24 April 2023
Revised: 22 June 2023
Accepted: 26 June 2023
Published: 27 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

There has been significant progress in the field of digital signal processing [1–7]. With the advancement of deep learning, researchers have increasingly focused on processing digital information using neural networks, including the synthesis of images through generative models. Our world, with its diverse range of objects and scenes, is inherently compositional. Compositionality emphasizes that complex entities are composed of simpler parts or components [8,9]. In the realm of generative models for image synthesis, it is crucial to embrace this fundamental principle that underlies the fabric of our reality. By considering objects as compositions of semantic parts, we can generate images that not only possess visual appeal but also exhibit a deeper level of interpretability. This approach grants us greater control over the generation process and facilitates the creation of highly customizable and meaningful images. Unfortunately, much of the existing literature has overlooked this principle, focusing solely on visual appeal or layout manipulation without leveraging compositionality. A limited number of recent works have proposed models based on compositionality, but these studies often neglect an important factor: efficiency. Real images or scenes typically involve a significant number of semantic parts and, without taking efficiency into consideration, the generative model becomes non-scalable and impractical. Hence, it is imperative to address both compositionality and efficiency when developing generative models for image synthesis.

Generative Adversarial Networks (GANs) have emerged as the dominant approach for image generation, primarily due to their superior performance in generating high-fidelity, visually appealing images. A typical example is the style-based GAN [10], where a noise code drawn from a Gaussian distribution is mapped to a latent space. This latent code is then used to modulate each layer of the model. Although this method can generate high-fidelity images efficiently, the generation process lacks semantic meaning. This

limitation arises from the use of a single latent code to modulate the generation of the entire image, without the ability to represent the semantic meanings of its components. In our work, we propose to utilize a set of latent codes, each of which represents a specific semantic meaning, to generate images. These latent codes encode both the individual components and their relationships, allowing for the formation of complete images based on the compositional principle.

Another line of research focuses on the mapping from layout to image, exemplified by SPADE [11]. Such an approach utilizes the layout as a content latent code to modulate the style of the image. While efficient and capable of achieving semantic control, these methods require the layout as a conditional input for image generation and cannot generate images from scratch. As the semantic meaning is provided by the user—rather than learned by the model—there is no explicit modeling of semantic parts and the model lacks compositionality.

One promising new model, called SemanticStyleGAN, has recently been proposed by Yichun et al. [12]. This model utilizes one latent code to modulate a set of local generators, with each local generator representing a semantic component. Following the compositional principle, these components are combined to form the complete object. However, a significant limitation of their model lies in the need for substantial local generators to represent each semantic part, resulting in linear growth in the number of local generators. This limitation makes their model non-scalable and inefficient. In contrast, our approach goes in the opposite direction: instead of using a single latent code to modulate multiple heavy local generators, we employ a set of latent codes to modulate a single local generator. This results in an efficient model that reduces the required number of local generators to a constant.

The limitations of the discussed models can be summarized based on three criteria: whether they are semantic-aware, compositional and efficient (see Table 1). It is evident that those models proposed in the existing literature either lack semantic awareness, compositionality or efficiency. In this paper, we introduce SSSGAN to address these limitations. The primary contribution of our work is a novel method that addresses the challenging task of generating all parts using a single local generator. This innovation is crucial, due to the inherent difficulties posed by strong entanglement and mode collapse problems associated with the generation of semantic parts using a shared generator. These challenges are further elaborated on in Section 3. The resulting model is both semantic-aware and compositional, while maintaining high efficiency. We conducted a comprehensive comparative analysis between SSSGAN and a state-of-the-art model, clearly demonstrating that SSSGAN excels in efficiency while making only minimal compromises in compositional performance. Our contributions are summarized as follows:

1. We propose a novel method that enables the generation of all semantic parts using a single local generator. This method effectively addresses the efficiency challenge encountered in large-scale compositional image-synthesis tasks.
2. We introduce SSSGAN, a model that is semantic-aware, compositional and efficient. It addresses the gap in existing works by meeting all three of these criteria simultaneously.
3. We conduct extensive experiments to evaluate the performance of SSSGAN, examining its compositional properties and the characteristics of its latent space.
4. We investigate the entanglement mechanism between semantic parts and devise a method to control the intensity of this entanglement.

Table 1. Comparison of representative models with the proposed model.

Model	Semantic-Aware	Compositional	Efficient
StyleGAN [10]	✗	✗	✓
SPADE [11]	✓	✗	✓
SemanticStyleGAN [12]	✓	✓	✗
Ours	✓	✓	✓

In the following sections, we delve deeper into the topic of compositional image generation. Section 2 provides a comprehensive review of the related literature, highlighting the existing approaches and their limitations. In Section 3, we present our proposed model, SSSGAN, detailing its architecture and key components. The experimental setup and results are presented in Section 4, and we evaluate the performance and efficiency of SSSGAN against a state-of-the-art model. Finally, in Section 5, we conclude the paper by summarizing our contributions, discussing the limitations faced and outlining future research directions in this field.

2. Related Work

Here, we review the related works in three areas: Disentanglement in generative models, compositional image synthesis and layout-to-image generation. It is worth noting that disentanglement and composition are inter-related and a work may be relevant to both aspects. To ensure clarity, we categorize the models that involve a compositional step into the compositional group, and those that do not into the disentanglement group. To enhance comprehension, the reviewed works are summarized in Table 2.

Disentanglement in Generative Models. Disentanglement refers to the property of a learned representation being sensitive to changes in one generating factor while being relatively insensitive to changes in other factors [13]. Some studies promote disentanglement by regularizing the distribution of latent codes to be factorized or independent [14,15]. This regularization encourages the latent code to control specific attributes of the generated image. Other studies introduce additional information loss to enhance the influence of factorized latent codes [16,17]. However, due to the lack of explicit inductive bias, the disentanglement achieved by these models is not predictable and the phenomenon is often not apparent. In contrast, our model incorporates an explicit inductive bias that considers an object as a composition of several parts, with each part defined in a supervised manner and assigned a semantic meaning. As a result, our model achieves significantly improved disentanglement, compared to these existing models.

Some studies have focused on achieving disentanglement between specific attribute pairs, such as style and content [18,19], identity and pose [19], identity and domain [20], shape and physical attribute [21] and appearance and geometry [22–27]. In contrast, our model aims to disentangle semantic parts defined by humans. Another line of research aims to disentangle representations in the hierarchy dimension by incorporating structure priors into the model [28–31] or by utilizing structured noise codes [32]. However, as the disentanglement in these models is achieved without supervision, the resulting disentangled factors may not align well with human understanding. In contrast, the disentangled factors in our model align closely with human conception.

Compositional Image Synthesis. Composition reflects the concept that an object is composed of its components and the goal of compositional learning is to explicitly model these components and their interactions [33]. Some researchers have directly performed compositional image synthesis in pixel space [34–36]. However, these models often generate floating objects over the background or are only effective on simple datasets, due to the limited representational power of the pixel space. In contrast, our model achieves high-fidelity image synthesis on real-image datasets.

It is important to note that our world is inherently three-dimensional. Several models with a 3D prior have recently been proposed in the literature [37–44]. These models typically encode a 3D prior by incorporating explicit 3D operations such as camera view direction, rotation and translation. However, these models are still designed to decompose the image at the object level and are unable to fully capture the fine-grained part-level decomposition.

Table 2. Analysis and summary of relevant research.

References	Achievement	Disadvantage	Comparison with Our Model
[14,15]	Increased disentanglement by encouraging latent codes to be factorized	Non-obvious, non-predictable disentanglement	Disentanglement factor has semantic meaning
[16,17]	Use of mutual information loss to boost factorized latent codes for improved disentanglement	Non-obvious, non-predictable disentanglement	Disentanglement factor has semantic meaning
[18,19]	Style–content disentanglement	Not fine-grained	Part-level disentanglement
[19]	Identity–pose disentanglement	Not fine-grained	Part-level disentanglement
[20]	Identity–domain disentanglement	Not fine-grained	Part-level disentanglement
[21]	Shape and physical attribute disentanglement	Not fine-grained	Part-level disentanglement
[22–27]	Appearance–geometry disentanglement	Not fine-grained	Part-level disentanglement
[28–31]	Structure disentanglement by customizing model architecture, unsupervised	Disentanglement factor does not align well with human conception	Disentanglement factor aligns with human conception
[32]	Structure disentanglement by use of structured noise codes, unsupervised	Disentanglement factor does not align well with human conception	Disentanglement factor aligns with human conception
[34–36]	Compositional image synthesis in pixel space	Floating object over background or feasible on simple dataset	Composition in hidden space, high quality on real-image dataset
[37–44]	3D-aware compositional image synthesis	Decomposes images into object and background, not fine-grained	Part-level compositional image synthesis
[45,46]	Decomposes image into foreground shape, foreground appearance and background	Not fine-grained	Part-level compositional image synthesis
[47]	Part-level compositionality, unsupervised	Decomposition quality is low	With supervision, high-quality decomposition
[12]	Part-level compositionality, high-quality for image and decomposition	Need substantial amount of heavy local generators, inefficient	Uses single local-generator, efficient
[48–54]	Sketch-to-image mapping	Semantic-agnostic, need sketches as input	Semantic-aware, no need for input
[11,55–58]	Semantic layout-to-image mapping, semantic aware	Object-level, need semantic layout as input	Part-level, no need for input, generation from scratch
[59,60]	Semantic layout-to-image mapping, part-level	Need layout as input	No need for input, generation from scratch
[60–62]	Semantic layout-to-image mapping, supports both part- or object-level layout	Needs layout as input	No need for input, generation from scratch
[55,62,63]	Semantic layout-to-image mapping, utilizing multi-scale methods to better encode semantic component	Needs layout as input	No need for input, generation from scratch

Some studies have explored compositional learning by decomposing objects into shape and appearance representations [45,46]; however, these are still not as fine-grained as that in our model. To the best of our knowledge, only a limited number of studies have achieved compositional learning at the semantic part level [12,47]. Among them, SemanticStyleGAN [12] stands out for its high-quality compositional image synthesis ability. However, SemanticStyleGAN sacrifices efficiency for performance by employing separate local generators for each semantic part. Additionally, it requires explicit factorization of the latent code into base,

shape and texture codes, resulting in a complex and inefficient model. In contrast, our model overcomes these limitations by using a single local generator for all parts and eliminates the need for latent code factorization, making it more efficient and simple.

Layout-to-Image Generation. Many studies have focused on mapping layouts to realistic images, which is valuable for controllable image editing and creation. Some studies have utilized sketches as input, which lack semantic meaning [48–54]. Other studies have employed segmentation masks. The granularity of the semantic component varies from object level [11,55–58] to part level [59,60]. Certain models support both object- and part-level semantic layouts [60–62]. Some researchers have utilized multi-scale methods during training to capture the varying granularity of semantic components [55,62,63]. While these models achieve high-fidelity image synthesis and provide control over the semantic component(s), they share a common drawback, when compared to our model: they require a layout as input. As a result, they are unable to generate images from scratch and their models are not compositional.

3. Methods

Our method is based upon SemanticStyleGAN. In order to understand the proposed approach in this work and the reason why SemanticStyleGAN sacrifices efficiency for performance, it is necessary to comprehend the challenges involved in image composition at the semantic part level. These challenges can be identified in two aspects. First, at the part level, the components are typically strongly entangled with each other, making it difficult to achieve compositional learning, necessitating the disentanglement of the parts as a prerequisite. Second, the parts themselves are usually heterogeneous objects originating from different distributions, posing a challenge for GANs to model due to the notorious mode collapse problem [64]. So, how does SemanticStyleGAN address these challenges and what is the trade-off? At the core of SemanticStyleGAN is a set of local generators responsible for generating the corresponding features for each semantic part—in other words, it utilizes a separate generator for each semantic part. This design naturally resolves the second difficulty mentioned earlier, as each local generator only needs to generate one semantic part from a single distribution. Moreover, this design also aids in addressing the first challenge, as there is no interaction between the parts within the local generator, given that each local generator has its own parameters. While SemanticStyleGAN has good performance, the drawback of this design is apparent: as it requires a dedicated local generator for each semantic part, the number of local generators increases linearly with the number of semantic parts, making the model impractical when dealing with a large number of semantic parts. Therefore, addressing this issue is crucial for achieving large-scale compositional image synthesis at the semantic part level.

So, the following question arises: can we achieve similar performance with SemanticStyleGAN by using fewer generators or even a single local generator? To do so, we face the same challenges mentioned above. To tackle these challenges, we drew inspiration from the auto-regressive model [65], in which the image is generated in a pixel-by-pixel manner, with each pixel being sampled conditioned on the previously sampled pixels. Our main idea is to apply a similar approach to model the relationships between the noise codes. By doing so, the sampled noise codes will contain the dependency information. Despite processing the noise codes with a shared local generator, it is possible for the GAN to utilize this dependency information to encode each part feature in its own distinct way. Therefore, this design has the potential to overcome the aforementioned challenges. The key distinction between the original auto-regressive model and our approach lies in the generation process. While the original auto-regressive model generates images pixel-by-pixel, our approach generates them part-by-part. Additionally, in the original auto-regressive model, no semantic meaning is assigned to each pixel. However, in our approach, our model learns the semantic meaning of each part's noise code during training. The necessity of the proposed approach is further examined in Section 4.2.4.

The architecture of SSSGAN is depicted in Figure 1. In particular, SSSGAN is built upon SemanticStyleGAN [12], consisting of three main modules: A set of local generators, a render network and a dual-branch discriminator. SSSGAN introduces modifications to SemanticStyleGAN in three key aspects. First, it incorporates a LSTM module to model the relationships between the parts and generate the noise codes. Second, it employs a single local generator for generating all the parts. Third, unlike SemanticStyleGAN—which factorizes the latent code into a base code, shape code, and texture code to enhance the encoding of part relationships—our model leverages the LSTM module to capture the part dependencies, rendering the manual factorization of latent codes unnecessary. Instead, we use a single latent code to modulate the local generator. To offer a comprehensive understanding of the information processing flow in SSSGAN, we provide brief descriptions of the modules inherited from SemanticStyleGAN as needed. For more detailed information, please refer to the original paper [12].

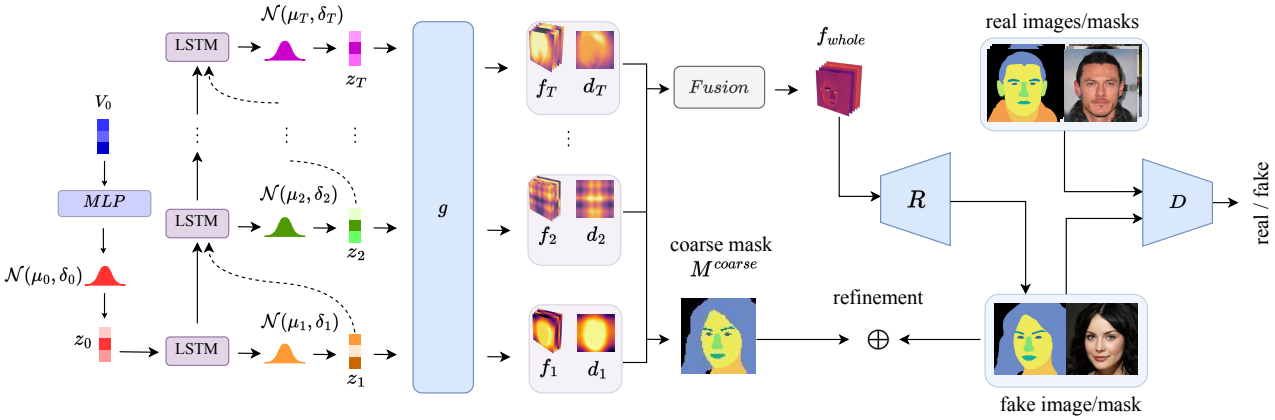


Figure 1. Overview of the SSSGAN architecture: The LSTM module models noise code relationships, the shared generator g generates part features, the *Fusion* module fuses part features, the render net R renders features to images and the discriminator D provides adversarial loss.

Noise Code Generation. As SSSGAN uses a single local generator for the generation of all parts, we need to explicitly model their joint distribution $P(\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_K)$, where $\mathbf{q}_k$, $k = 1, \cdots, K$ is a random variable representing the kth part. Instead of directly modeling this joint distribution, inspired by the auto-regressive model [65], we factorize this joint distribution into the product of a sequence of conditional distributions, as defined below:

$$P(\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_K) = \prod_{k=1}^{K} P(\mathbf{z}_k | \mathbf{z}_1, \cdots, \mathbf{z}_{k-1}). \tag{1}$$

We start the generation of the noise codes from a learnable constant vector V_0, which is mapped to a tuple $(\mu_0, \log var_0)$ by a two-layer MLP; here, μ_0 and var_0 represent the mean and variance of the Gaussian distribution, respectively. The logarithm operation is performed to ensure that the variance is a positive value. The initial input to the LSTM layer is sampled from the diagonal Gaussian $\mathcal{N}(\mu_0, \delta_0)$ using the re-parameterization trick:

$$z_0 = \epsilon_0 \odot \delta_0 + \mu_0, \tag{2}$$

where $\odot$ denotes element-wise product, $\epsilon_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\delta_0 = \sqrt{var_0}$. $\mathcal{N}(\mathbf{0}, \mathbf{I})$ represent the multivariate Gaussian distribution with zero mean $\mathbf{0}$ and identity covariance matrix $\mathbf{I}$.

At step t, the LSTM layer takes two vectors as input: one is the noise code z_{t-1} sampled at step $t-1$ and the other is the hidden state of the LSTM layer h_{t-1}, which aggregates the interaction of the parts up to step $t-1$. The output layer of the LSTM is a single linear layer that maps the hidden state h_t to a tuple $(\mu_t, \log var_t)$. Similarly, the noise code z_t is

sampled from the diagonal Gaussian $\mathcal{N}(\mu_t, \delta_t)$ using the re-parameterization trick, defined as follows:

$$z_t = \epsilon_t \odot \delta_t + \mu_t, \tag{3}$$

where $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\delta_t = \sqrt{var_t}$.

In our experiment, we set the maximum step T to the part count K, which means that we generate one part at each step.

Part Feature Generation and Composition. The noise codes $\{z_k\}_{k=1}^K$ are mapped to a set of latent codes $\{w_k\}_{k=1}^K$ by a group of equal linear layers [66]. As mentioned above, we do not factorize the latent code as in SemanticStyleGAN and use one latent code to modulate the generator; that is, SemanticStyleGAN uses a separate local generator for the generation of each part, while we use a shared local generator for the generation of all the parts. The architecture of the local generator is shown in Figure 2. The local generator is a stack of modulated convolutional layers [66] and the *toDepth* and *toFeat* modules are mainly single modulated convolutional layers with fixed style. The local generator g takes the Fourier feature p and the kth latent code w_k as input, then produces a pseudo depth map $d_k \in \mathbb{R}^{H_f \times W_f \times 1}$ and a feature map $f_k \in \mathbb{R}^{H_f \times W_f \times D_f}$, where H_f, W_f and D_f are the spatial height, spatial width and channel dimension, respectively. The mapping is defined as:

$$g : (p, w_k) \mapsto (f_k, d_k). \tag{4}$$

The main logic of the composition process is to use the pseudo depth map d_k as a weighting factor for the feature map f_k. Specifically, the depth maps are squeezed to a channel-wise probability function by the softmax function, defined as:

$$M_k^{coarse}(i,j) = \frac{exp(d_k(i,j)/\tau)}{\sum_{k'}^{K} exp(d_{k'}(i,j)/\tau)}, \tag{5}$$

where τ is a temperature hyper-parameter with default setting of 1. Then, the feature maps $\{f_k\}_{k=1}^K$ are fused using weighted summation, defined as follows:

$$f_{whole} = \sum_{k}^{K} M_k^{coarse} \odot f_k, \tag{6}$$

where $\odot$ represents element-wise multiplication.

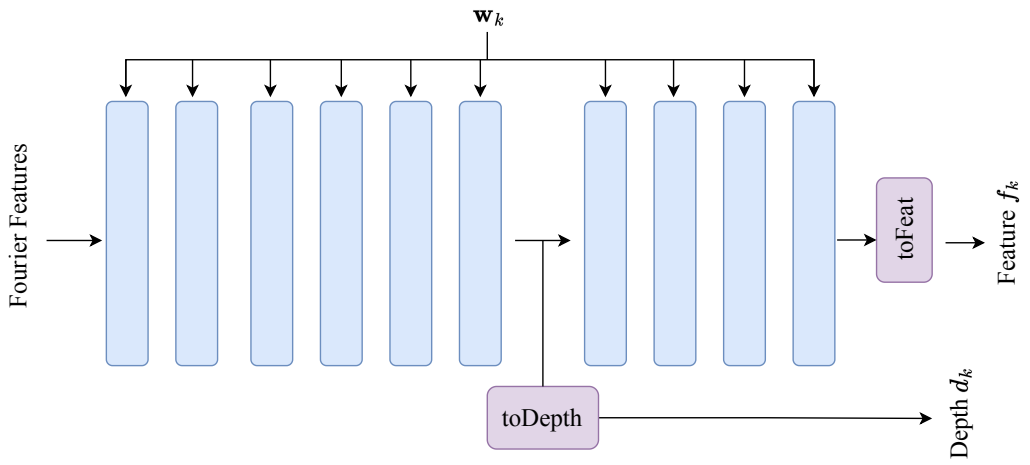


Figure 2. The architecture of the local generator. The blue blocks are modulated convolutional layers and the purple blocks are both a single modulated convolutional layer with fixed style.

Image Rendering. Following SemanticStyleGAN [12], the whole feature map f_{whole} is rendered into realistic images using a render net R. The render net R is actually the generator of StyleGAN [66]. As our goal is to render the feature and generate the image, R

should only depend on the whole feature f_{whole}. To achieve this, the style of the generator is fixed. The render net is defined as follows:

$$R : f_{whole} \mapsto (\tilde{x}, M^{fine}), \tag{7}$$

where $\tilde{x} \in \mathbb{R}^{H \times W \times 3}$ and $M^{fine} \in [0,1]^{H \times W \times K}$ are the fake image and the fake segmentation mask, respectively; and H and W represent the image height and width, respectively.

The segmentation signal directly interacts with the fine mask M^{fine}, but only indirectly interacts with the coarse mask M^{coarse} through the pseudo depth maps. As M^{coarse} is crucial for decomposing the part features, a mask loss is used to strengthen the supervision signal over M^{coarse}, defined as:

$$\mathcal{L}_{mask} = \|upsample(M^{coarse}) - M^{fine}\|_2, \tag{8}$$

where *upsample* denotes the bilinear interpolation function.

Discriminator and Loss. As the output of the render net is a synthesized image and a segmentation mask, a dual branch discriminator D is used to enhance the regularization of the segmentation mask. This discriminator is a variation of the discriminator in StyleGAN [66], with the addition of a *toSeg* branch, which consists of a stack of equal convolution layers [66]. In the discriminator, the image feature and the segmentation feature are added for further processing. The benefit of this design is that a separate $R1$ regularization term [67] $\mathcal{L}_{R1_{seg}}$ can be applied specifically to the segmentation mask.

As there is no constraint on the values of the noise codes, we empirically observed that the values of the noise codes tend to grow very large during training, leading to training instability. To address this issue, we introduce a Kullback–Leibler divergence term, defined as follows:

$$\mathcal{L}_{KL} = \frac{1}{T} \sum_{t=1}^{T} -D_{KL}(P(\mathbf{z}_t|\mathbf{z}_1, \cdots, \mathbf{z}_{t-1}) \| \mathcal{N}(\mathbf{0}, \mathbf{I})). \tag{9}$$

The final loss is

$$\mathcal{L}_{all} = \mathcal{L}_{stylegan2} + \mathcal{L}_{mask} + \mathcal{L}_{R1_{seg}} + \mathcal{L}_{KL}, \tag{10}$$

where $\mathcal{L}_{stylegan2}$ represents the loss used in StyleGAN2 [10].

4. Experiments and Results

4.1. Dataset and Training Method

Our main results were obtained using the CelebAMask-HQ dataset [68], which is a large-scale face image dataset consisting of 30,000 high-resolution images accompanied by fine-grained segmentation masks for facial attributes. This dataset is well-suited to our fine-grained compositional image synthesis task. To ensure a fair benchmark comparison with SemanticStyleGAN, we adopted the same pre-processing method. Specifically, we used the first 28,000 facial images and their corresponding segmentation masks for training. The segmentation masks include thirteen semantic parts: Background, skin, eyes, eyebrows, nose, mouth, ears, hair, neck, clothing, eyeglasses, hats and earrings. Prior to training, we resized both the images and segmentation masks to 256 $\times$ 256 pixels and normalized them within the range of $[-1, 1]$.

We also attempted to use the same optimization settings as SemanticStyleGAN, whenever possible, to ensure a fair comparison. For training, we employed the Adam optimizer with $\beta = (0, 0.99)$. In SemanticStyleGAN, a single learning rate of 0.002 was used. However, we observed that this value is too high when applied to the LSTM module with the Adam optimizer, as it leads to gradient spikes and disrupts the training process. Consequently, we introduced a separate learning rate of 0.0001 exclusively for the LSTM module to mitigate this issue.

To regularize the training process, we applied path regularization [10] on the generator and $R1$ regularization on the discriminator every 4 and 16 iterations, respectively. The

weight factors for $\mathcal{L}_{mask}$ and $\mathcal{L}_{KL}$ were set to 100 and 0.0001, respectively. As we explicitly modeled the relationships between the parts using the LSTM module, we observed that style mixing had a detrimental effect on the training. This is expected, as style mixing assumes independent styles for the parts, which contradicts the function of the LSTM module. Therefore, we did not utilize style mixing during training.

Our model was trained using a single RTX3090 GPU, with a batch size of 16 and gradient accumulation every two iterations, resulting in an effective batch size of 32. We set the maximum number of training steps to 150,000, with each iteration taking approximately 4 s.

4.2. Results

4.2.1. Image Quality Evaluation

The quality of the generated images is a crucial aspect for a generative model. In Figure 3, we show some generated images produced with a resolution of 256 × 256 and with a truncation of 0.7. It is evident that these images have high perceptual quality and the corresponding segmentation masks accurately align with their intended semantic meanings. To quantitatively assess the image quality, two widely used metrics—-namely, the Fréchet Inception Distance (FID) [69] and Inception Score (IS) [70]—were employed. A lower FID value or a higher IS value indicates higher image quality. The FID and IS scores are summarized in Table 3. Notably, SSSGAN achieved the lowest FID value of 6.19, albeit with a slightly lower IS value of 3.15. It should be acknowledged that quantitative evaluation of image quality remains an open problem; overall, based on the results, we can conclude that SSSGAN achieves comparable performance in generating high-quality images.

The semantic parts of a face exhibit strong correlations with each other. To generate a high-fidelity facial image, it is essential to arrange the latent codes for these parts in a cohesive manner. Our findings demonstrated that the LSTM module of SSSGAN effectively captures the relationships between the parts, enabling a single local generator to encode the facial images using these latent codes.

Table 3. Quantitative evaluation of the quality of the synthesized images.

Method	Data	Compositional	FID ↓	IS ↑
SemanticGAN [1]	img & seg	✗	7.50	3.51
SemanticStyleGAN [1]	img & seg	✓	6.42	3.21
SSSGAN	img & seg	✓	6.19	3.15

[1] These data are from [12].

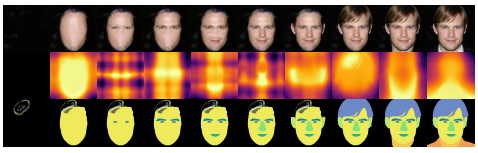


Figure 3. Examples of images and segmentation masks generated by SSSGAN.

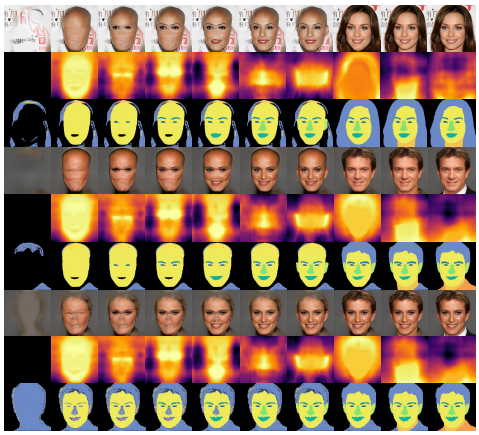
4.2.2. Compositional and Disentanglement Properties

Sequential Part Composition. Sequential Part Composition (SPC) is a task that involves sequentially adding components to an object for composition. This task serves as an intuitive demonstration of both the model's compositional ability and its disentanglement property. In terms of compositional property, a desirable model should accurately add each semantic part. Regarding the disentanglement property, there should be minimal interference with previously added components when a new component is introduced.

As SSSGAN utilizes a single local generator for all parts, it is possible that there are stronger interactions between the features of different parts, resulting in a higher degree of entanglement. The evaluation results for the SPC task are depicted in Figure 4. It is evident that SSSGAN successfully added the correct semantic part at each step, indicating its compositional ability. When comparing it with SemanticStyleGAN (Figure 4a), a stronger entanglement between the part features in SSSGAN can be observed; for example, in Figure 4b, the lip color and even the gender changed in the first and second samples, respectively. Furthermore, it can be observed that the most significant change occurred when the hair was added. This observation can be attributed to the fact that the hair is a prominent part of the human face, occupying a large portion of the image's pixels. Consequently, more neural activity is dedicated to representing its features. Given the limited resources, SSSGAN learned to exert greater control over facial attributes through interactions with the larger part (i.e., the hair).



(**a**) SemanticStyleGAN



(**b**) SSSGAN

Figure 4. Sequential part composition: (**a**) SemanticStyleGAN, and (**b**) SSSGAN. Each sample consists of three rows: The synthesized image, the pseudo depth map and the segmentation mask.

Although there was entanglement between the parts in SSSGAN, it occurred in a coherent manner. For instance, consider the skin: while its appearance changes after the addition of hair, a shared coarse structure remains before and after the change, indicating that the coarse structure is disentangled. This aspect may initially seem like a drawback, in terms of image attribute editing; however, we argue that entanglement also offers an advantage, in terms of representation efficiency. This entanglement enables the model to re-use learned features to represent a wider range of attributes. This ability becomes particularly crucial under limited computational resources.

Entanglement Mechanism. The next question was the following: how does this entanglement occur? As we did not employ any style-mixing tricks during training, the only source of disentanglement came from the supervision of the segmentation mask. We speculate that information leakage occurred in the spatial dimension, from one part to another. Figure 5 illustrates the proposed mechanism of entanglement between the parts. The coarse mask in Equation (5) was computed using a softmax activation function. It is possible that SSSGAN learns to decrease the intensity of the pseudo depth map to create a softer mask. Taking the situation in Figure 5 as an example, when the temperature is higher, the segmentation mask becomes softer. As a result, the features of part B may leak

to parts A and C, enabling part B to influence the attributes of parts A and C through their interaction.

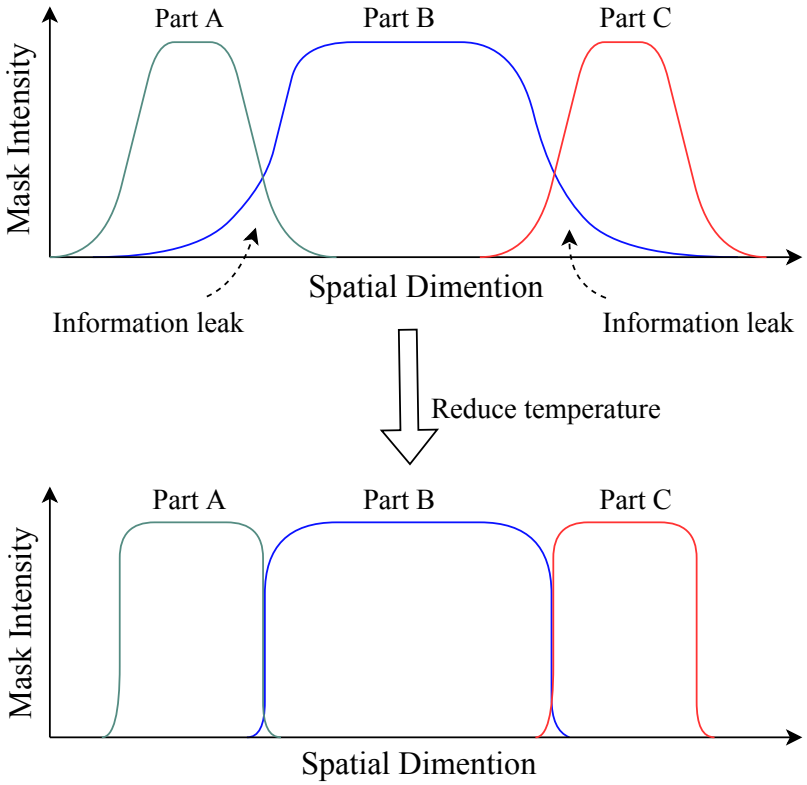


Figure 5. Illustration of the proposed entanglement mechanism.

Conversely, when we lower the temperature in Equation (5), the coarse mask becomes harder, making it more challenging for information to leak to neighboring parts. To validate our hypothesis, we decreased the temperature in Equation (5) from the default value of 1.0 to 0.01 and generated the second sample in Figure 4b again. The result is shown in the last sample of Figure 4b. It can be observed that, through the sequential composition process, there was minimal change in previous results when a new part was added and the gender of the individuals remained unchanged. While lowering the temperature can achieve better disentanglement, SSSGAN tended to generate femininity in males under lower temperatures. This outcome suggests a trade-off between disentanglement and diversity in SSSGAN, which can be controlled by the temperature variable. Higher temperatures increase the intensity of entanglement but yield more diverse images, while lower temperatures reduce entanglement intensity but also decrease the diversity of the generated images.

To quantitatively assess the level of disentanglement in the SPC task, we propose a metric called Sequential Mean Square Error (SMSE), as defined in Equation (11).

$$\mathrm{SMSE}_i = \frac{\sum_{h=1}^{H}\sum_{w=1}^{W}\left[\left(\sum_{j=1}^{i-1} M^i_{j,h,w}\right)\left(I^i_{h,w} - I^{i-1}_{h,w}\right)\right]^2}{\sum_{h=1}^{H}\sum_{w=1}^{W}\sum_{j=1}^{i-1} M^i_{j,h,w}}, \tag{11}$$

where h, w represent the spatial height and width indices, respectively; i denotes the sequence index; $M^i_{j,h,w} \in \{0, 1\}$ represents the mask value at location (h, w) of the jth part in step i; and $I^i_{h,w}$ represents the pixel value at location (h, w) in step i. The SMSE metric quantifies the intensity of change in the original image when the ith part is added. A lower

SMSE value indicates a better performance, in terms of disentanglement, meaning that the added part has a lesser impact on the rest of the image.

Figure 6 displays the quartiles of the SMSE values at different temperatures for each semantic part. For quantitative comparison, Table 4 presents the SMSE values and their corresponding standard deviations. Each row in the table corresponds to a specific part, while each column represents the SMSE values at different temperatures. The lowest SMSE value for each part is highlighted in bold font in the table. From Figure 6, it can be observed that the skin and hair exhibited relatively larger SMSE values, indicating higher entanglement for larger parts. Additionally, SSSGAN with the default temperature of 1.0 exhibited higher SMSE values, compared to SemanticStyleGAN. However, by reducing the temperature in Equation (5) for SSSGAN, the SMSE values were effectively reduced. For instance, as shown in Table 4, the skin's SMSE value for SSSGAN with a temperature of 0.01 was 0.022 ± 0.036, significantly lower than that for SemanticStyleGAN (0.078 ± 0.101). This result indicates that the level of entanglement can be significantly reduced by lowering the temperature. This finding is in alignment with the proposed entanglement mechanism illustrated in Figure 5.

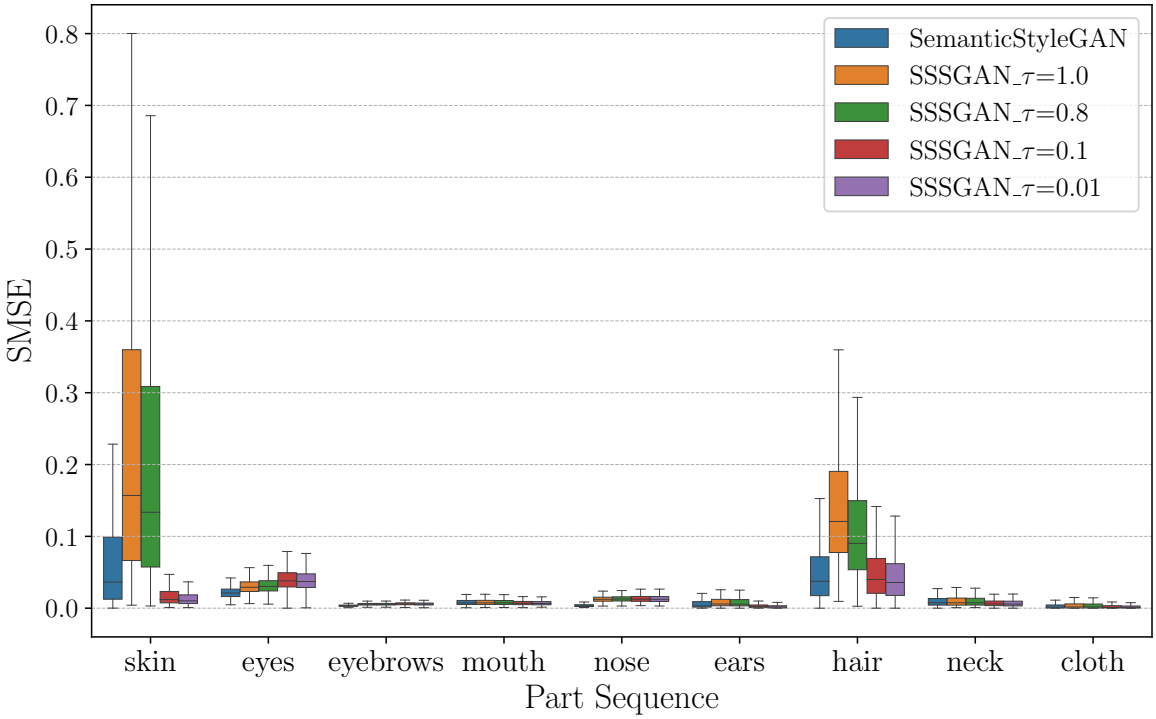


Figure 6. Quartiles of SMSE for the semantic parts. The statistical calculation is based on 10,000 samples.

Table 4. SMSE mean value and standard deviation for SemanticStyleGAN and SSSGAN at different temperatures.

Model	**SemanticStyleGAN**	**SSSGAN**			
Temperature	**1.0**	**1.0**	**0.8**	**0.1**	**0.01**
skin	0.078 ± 0.101	0.244 ± 0.232	0.214 ± 0.210	0.028 ± 0.044	$\mathbf{0.022 \pm 0.036}$
eyes	$\mathbf{0.022 \pm 0.009}$	0.031 ± 0.011	0.032 ± 0.012	0.041 ± 0.017	0.040 ± 0.016
eyebrows	$\mathbf{0.004 \pm 0.002}$	0.006 ± 0.003	0.006 ± 0.002	0.006 ± 0.002	0.006 ± 0.002
mouth	0.009 ± 0.006	0.009 ± 0.005	0.009 ± 0.005	$\mathbf{0.008 \pm 0.004}$	$\mathbf{0.008 \pm 0.004}$
nose	$\mathbf{0.004 \pm 0.003}$	0.013 ± 0.005	0.014 ± 0.005	0.014 ± 0.006	0.014 ± 0.007
ears	0.008 ± 0.012	0.012 ± 0.016	0.011 ± 0.016	0.005 ± 0.010	$\mathbf{0.004 \pm 0.010}$
hair	0.053 ± 0.051	0.149 ± 0.104	0.115 ± 0.090	0.052 ± 0.044	$\mathbf{0.046 \pm 0.041}$
neck	0.011 ± 0.012	0.013 ± 0.017	0.012 ± 0.016	0.008 ± 0.007	$\mathbf{0.008 \pm 0.007}$
cloth	0.004 ± 0.006	0.005 ± 0.009	0.005 ± 0.009	0.003 ± 0.005	$\mathbf{0.002 \pm 0.004}$

4.2.3. Noise and Latent Space Property

As SSSGAN utilizes a shared local generator for generating all the parts, we can analyze the properties of the noise codes and latent codes directly, without considering the local generator. In particular, we can visualize the noise codes and latent codes through low-dimensional embeddings. Figure 7 presents the two-dimensional embedding results using the t-SNE algorithm [71]. It can be observed that the noise codes were not well-separated, with clusters formed by the background, neck and clothes being close together. In contrast, the latent codes were well-grouped and did not overlap. These results suggest that the latent space of SSSGAN is flatter (or more linear) than the noise space. Consequently, the t-SNE algorithm finds it easier to separate the parts in the latent space, compared to the noise space. Furthermore, we noticed that the noise codes for consecutive semantic parts, such as the eyes, eyebrows and mouth, were closely grouped. This finding indicates that the LSTM module of SSSGAN can capture the relationships between these parts well.

Given that t-SNE is designed to capture local patterns within a distribution, we also examined the relationships between the noise codes and latent codes by measuring the pairwise cosine similarity. The results, illustrated in Figure 8, revealed a distinct pattern: the cosine similarity between parts was significantly stronger in the latent space, compared to the noise space. This suggests that the vectors representing the parts are more closely distributed, possibly forming a smoother manifold, in the latent space. In contrast, the distribution of part vectors in the noise space appeared to be less structured. Additionally, in the noise space, we observed slightly stronger cosine similarity between consecutive parts, indicating that the LSTM module can successfully capture the interdependence of adjacent parts, in line with our earlier analysis. Furthermore, in the latent space, we observed a stronger cosine similarity among facial components such as the eyes, eyebrows, mouth and nose. This observation indicates stronger correlations between these features, suggesting that they are distributed closer to each other in the manifold.

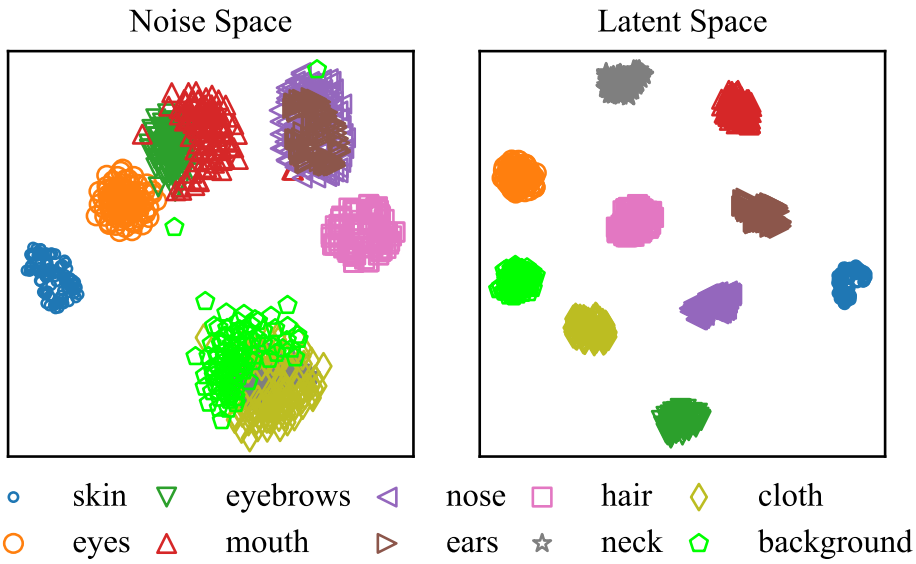


Figure 7. Noise code and latent code visualization using the t-SNE algorithm [71].

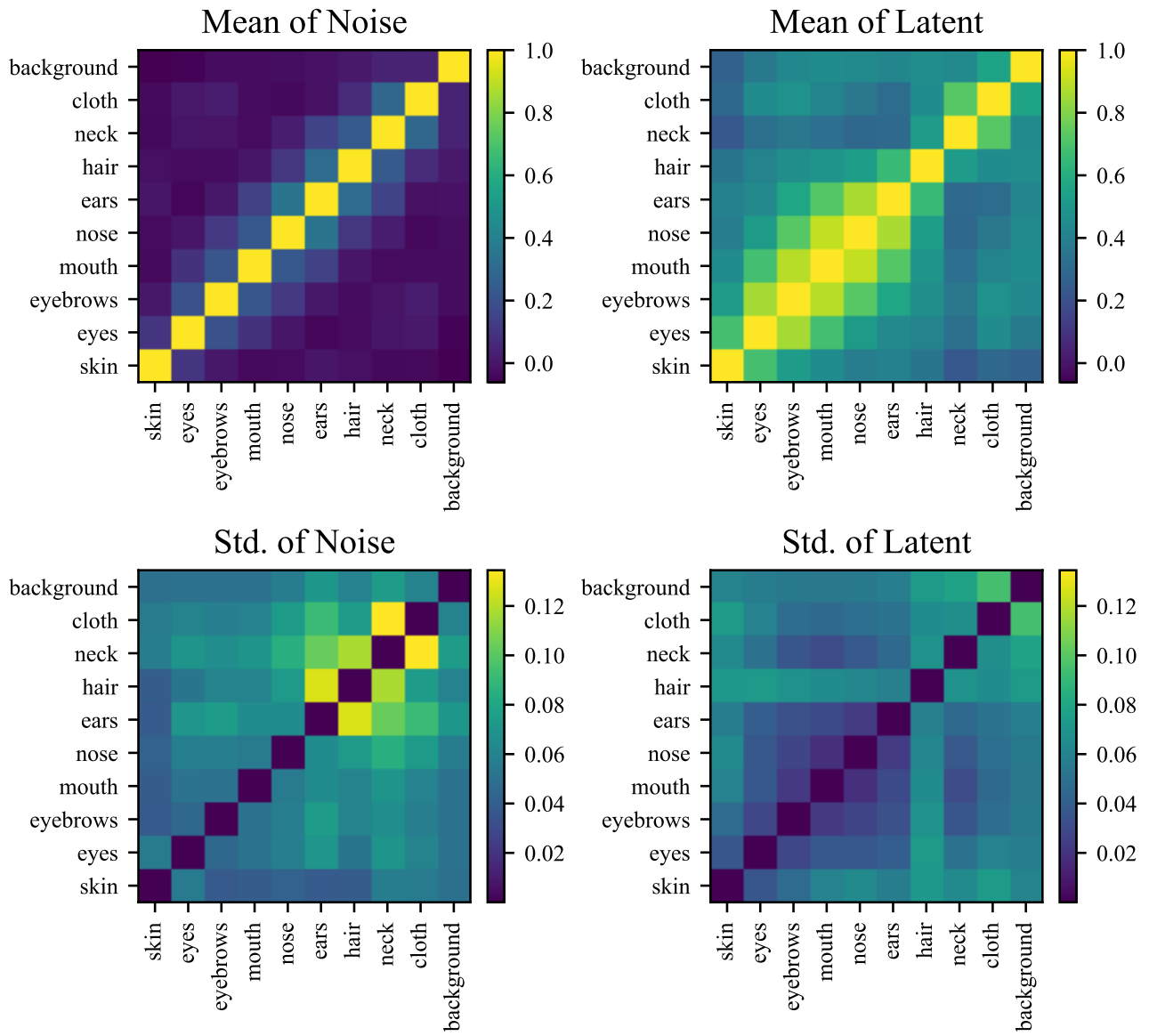


Figure 8. Correlation analysis between noise codes and latent codes using cosine similarity. The stochastic calculation is based on 10,000 samples.

4.2.4. Necessity of the LSTM Module

SSSGAN includes an LSTM module to explicitly capture the relationships between the noise codes, which plays a crucial role in the subsequent compositional synthesis steps. While GANs have the capability to generate diverse images, one might question the need for the LSTM module. Could SSSGAN achieve similar performance without modeling these relationships? To address this question, we conducted an ablation study by removing the LSTM module, resulting in a simplified version called SSSGAN-s. In SSSGAN-s, the noise codes for the parts were independently sampled from a Gaussian distribution, while other aspects remained the same as in SSSGAN. The experimental results on the CelebAMask-HQ dataset are presented in Figure 9, revealing that SSSGAN-s failed to generate the semantic mask, indicating a lack of awareness of the semantic part structure. To further examine how SSSGAN-s encodes the part noise code, we evaluated its performance on the sequential part composition task, as shown in Figure 10. It is evident that SSSGAN-s treated each part code as a whole face, completely disregarding the semantic meaning of each part. These findings clearly demonstrate the necessity of the LSTM module in SSSGAN, validating the effectiveness and novelty of our approach.

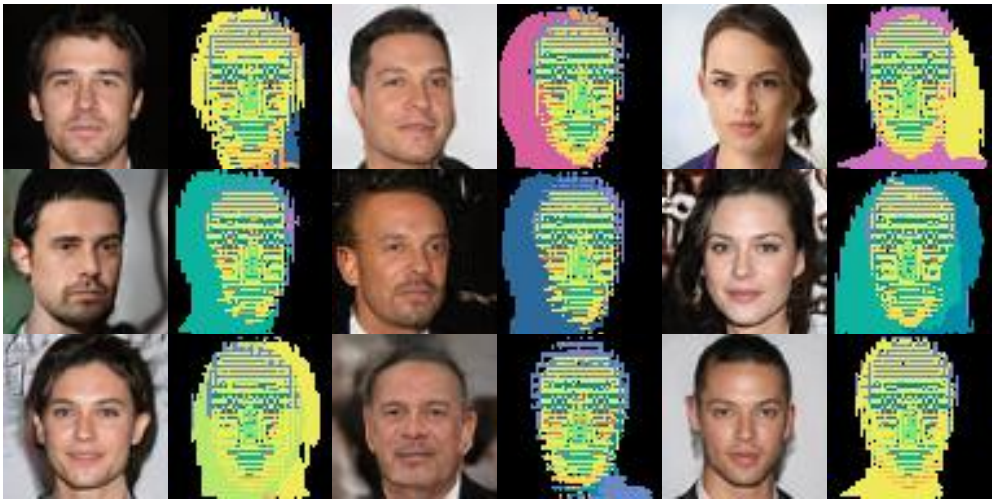


Figure 9. Images and segmentation masks generated by SSSGAN-s without the LSTM module.

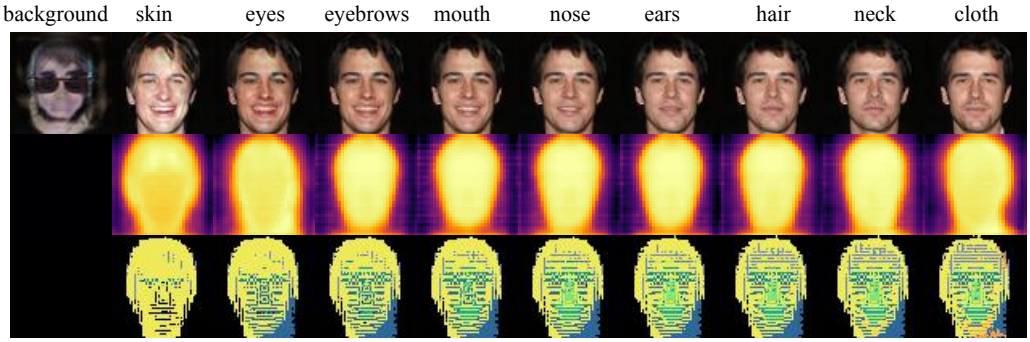


Figure 10. Sequential part composition results of SSSGAN-s without the LSTM module.

4.2.5. Performance on Other Domains

To demonstrate the versatility of our model beyond facial datasets, we conducted an evaluation on the DeepFashion dataset [72]. This dataset comprises 30,000 fashion images and we utilized the segmentation data provided in [60]. For supervision, we considered seven semantic parts; namely, background, skin, hair, face, upper clothes, pants and shoes. The experiment followed the same settings as our main results. The sequential part composition outcomes are presented in Figure 11. It is evident that the components were successfully combined one-by-one, starting from the background. At each step, the added part aligned with its semantic meaning and no apparent entanglement was observed. These results indicate that our model also demonstrated impressive performance on this dataset, in terms of compositional synthesis and disentanglement.

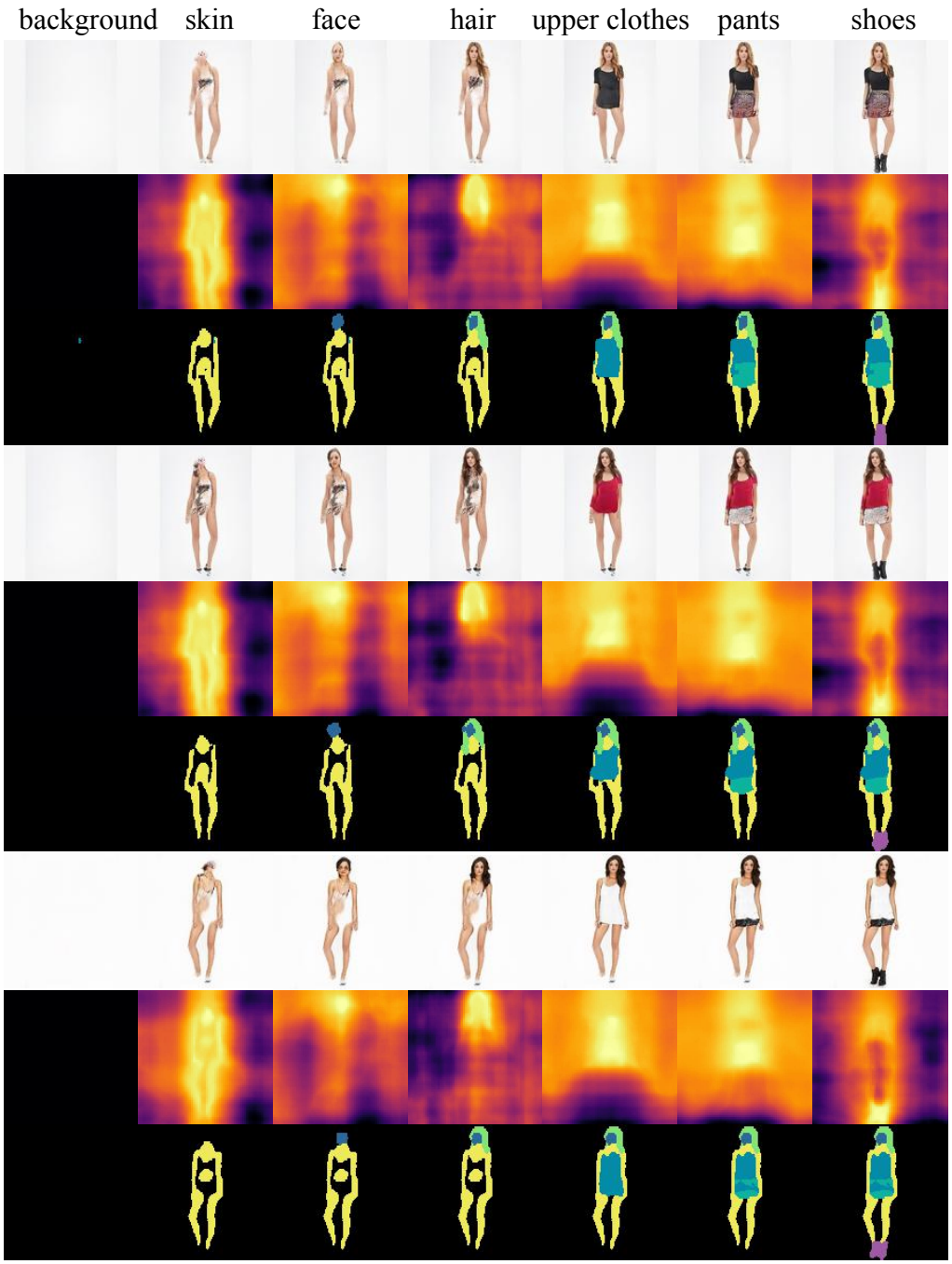


Figure 11. Sequential part composition results on DeepFashion dataset.

5. Conclusions

In this paper, we introduced SSSGAN, an efficient model for fine-grained compositional image synthesis. SSSGAN achieves improved efficiency by utilizing a shared generator to generate all of the semantic components. In order to accomplish this, we needed to address the challenges associated with entanglement and mode collapse. To overcome these challenges, we presented a novel method that explicitly models the relationships between the noise codes. Each noise code in SSSGAN controls the generation of a specific semantic part, which are then combined to form a complete object. Our model demonstrated impressive performance in generating high-quality images, providing a high level of control over the generation process for each semantic part, albeit with a slightly increased level of entanglement. We believe that efficiency is crucial for large-scale,

fine-grained compositional image synthesis and our work presents a fundamental methodology for advancing this field. However, there are still several challenges that need to be addressed in future research. First, improving the disentanglement performance while maintaining image diversity is crucial. Incorporating style-mixing techniques into our method (which is currently incompatible) could potentially achieve this goal. Second, our current model supports the generation of images with a fixed component count. However, it is important to extend its capability to handle dynamic component counts, which are common in real-world objects and scenes. This can potentially be achieved by explicitly modeling the component count in the composition process. Finally, similar to SemanticStyleGAN, SSSGAN relies on full supervision, and so would be valuable for developing models with similar performance having lower supervision requirements.

Author Contributions: Conceptualization, Z.W. and Z.L.; methodology, Z.W.; software, Z.W.; validation, Z.W.; formal analysis, Z.W.; investigation, Z.W. and Z.L.; resources, Z.L.; data curation,Z.W.; writing—original draft preparation, Z.W.; writing—review and editing, Z.W. and Z.L.; visualization, Z.W.; supervision, Z.L.; project administration, Z.L.; funding acquisition, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China grant number 62032019 and 61732019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code will be made publicly available upon acceptance at https://gitee.com/zongtao_wang/sssgan-pytorch, accessed on 23 April 2023.

Acknowledgments: We would like to express our gratitude to the anonymous reviewers for their valuable feedback, which greatly contributed to improving our research paper. Additionally, we acknowledge the GPU support provided by the Center for Research and Innovation in Software Engineering (RISE).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

GAN	Generative Adversarial Network
MLP	Multi-Layer Perceptron
SSSGAN	Single-Generator SemanticStyleGAN
SPC	Sequential Part Composition
SMSE	Sequential Mean Square Error
LSTM	Long Short-Term Memory
FID	Fréchet Inception Distance
IS	Inception Score

References

1. Guido, R.C.; Pedroso, F.; Contreras, R.C.; Rodrigues, L.C.; Guariglia, E.; Neto, J.S. Introducing the Discrete Path Transform (DPT) and Its Applications in Signal Analysis, Artefact Removal and Spoken Word Recognition. *Digit. Signal Process.* **2021**, *117*, 103158. [CrossRef]
2. Guariglia, E.; Silvestrov, S. Fractional-Wavelet Analysis of Positive definite Distributions and Wavelets on $\mathcal{D}'(\mathbb{C})$. In *Engineering Mathematics II*; Springer Proceedings in Mathematics & Statistics Series; Springer: Berlin/Heidelberg, Germany, 2016; pp. 337–353.
3. Yang, L.; Sun, H.; Zhong, C.; Meng, Z.; Luo, H.; Li, X.; Tang, Y.Y.; Lu, Y. Hyperspectral image classification using wavelet transform-based smooth ordering. *Int. Wavelets Multiresolution Inf. Process.* **2019**, *17*, 1950050:1–1950050:18. [CrossRef]
4. Guariglia, E. Harmonic Sierpinski Gasket and Applications. *Entropy* **2018**, *20*, 714. [CrossRef] [PubMed]
5. Guariglia, E. Primality, Fractality and Image Analysis. *Entropy* **2019**, *21*, 304. [CrossRef] [PubMed]

6. Zheng, X.; Tang, Y.Y.; Zhou, J. A Framework of Adaptive Multiscale Wavelet Decomposition for Signals on Undirected Graphs. *IEEE Trans. Signal Process.* **2019**, *67*, 1696–1711. [CrossRef]
7. Berry, M.V.; Lewis, Z.V.; Nye, J.F. On the Weierstrass-Mandelbrot fractal function. *Proc. R. Soc. Lond. Math. Phys. Sci.* **1980**, *370*, 459–484.
8. Osherson, D.N.; Smith, E.E. On the Adequacy of Prototype Theory as a Theory of Concepts. *Cognition* **1981**, *9*, 35–58. [CrossRef]
9. Lake, B.M.; Ullman, T.D.; Tenenbaum, J.B.; Gershman, S.J. Building Machines That Learn and Think Like People. *Behav. Brain Sci.* **2017**, *40*, e253. [CrossRef]
10. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and Improving the Image Quality of StyleGAN. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 8110–8119.
11. Park, T.; Liu, M.Y.; Wang, T.C.; Zhu, J.Y. Semantic Image Synthesis With Spatially-Adaptive Normalization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 2337–2346.
12. Shi, Y.; Yang, X.; Wan, Y.; Shen, X. SemanticStyleGAN: Learning Compositional Generative Priors for Controllable Image Synthesis and Editing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 11254–11264.
13. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: a Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef]
14. Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; Lerchner, A. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
15. Kim, H.; Mnih, A. Disentangling by Factorising. In Proceedings of the International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 2649–2658.
16. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; Volume 29, pp. 2172–2180.
17. Lin, Z.; Thekumparampil, K.; Fanti, G.; Oh, S. InfoGAN-CR and ModelCentrality: Self-supervised Model Training and Selection for Disentangling GANs. In Proceedings of the International Conference on Machine Learning (ICML) 2020, Virtual, 13–18 July 2020; Volume 119, pp. 6127–6139.
18. Kazemi, H.; Iranmanesh, S.M.; Nasrabadi, N. Style and Content Disentanglement in Generative Adversarial Networks. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 7–11 January 2019; pp. 848–856.
19. Tran, L.; Yin, X.; Liu, X. Disentangled Representation Learning GAN for Pose-Invariant Face Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1415–1424.
20. Liu, A.H.; Liu, Y.C.; Yeh, Y.Y.; Wang, Y.C.F. A Unified Feature Disentangler for Multi-Domain Image Translation and Manipulation. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*; NIPS: La Jolla, CA, USA, 2018; Volume 31, pp. 2595–2604.
21. Medin, S.C.; Egger, B.; Cherian, A.; Wang, Y.; Tenenbaum, J.B.; Liu, X.; Marks, T.K. MOST-GAN: 3D Morphable StyleGAN for Disentangled Face Image Manipulation. *AAAI Conf. Artif. Intell.* **2022**, *36*, 1962–1971. [CrossRef]
22. Skafte, N.; ren Hauberg, S. Explicit Disentanglement of Appearance and Perspective in Generative Models. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS),*; NIPS: La Jolla, CA, USA, 2019; Volume 32, pp. 1016–1026.
23. Lorenz, D.; Bereska, L.; Milbich, T.; Ommer, B. Unsupervised Part-Based Disentangling of Object Shape and Appearance. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 10955–10964.
24. Xing, X.; Gao, R.; Han, T.; Zhu, S.C.; Wu, Y.N. Deformable Generator Networks: Unsupervised Disentanglement of Appearance and Geometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1162–1179. [CrossRef] [PubMed]
25. Liu, L.; Jiang, X.; Saerbeck, M.; Dauwels, J. EAD-GAN: A Generative Adversarial Network for Disentangling Affine Transforms in Images. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–11. [CrossRef] [PubMed]
26. Tewari, A.; R, M.B.; Pan, X.; Fried, O.; Agrawala, M.; Theobalt, C. Disentangled3D: Learning a 3D Generative Model with Disentangled Geometry and Appearance From Monocular Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 1516–1525.
27. Nguyen-Phuoc, T.; Li, C.; Theis, L.; Richardt, C.; Yang, Y.L. HoloGAN: Unsupervised Learning of 3D Representations from Natural Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7588–7597.
28. Sønderby, C.K.; Raiko, T.; Maaløe, L.; Sønderby, S.R.K.; Winther, O. Ladder Variational Autoencoders. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*; NIPS: La Jolla, CA, USA, 2016; Volume 29, pp. 3738–3746.
29. Zhao, S.; Song, J.; Ermon, S. Learning Hierarchical Features from Deep Generative Models. In Proceedings of the International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; Volume 70, pp. 4091–4099.
30. Li, Z.; Murkute, J.V.; Gyawali, P.K.; Wang, L. Progressive Learning and Disentanglement of Hierarchical Representations. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.

31. Kaneko, T.; Hiramatsu, K.; Kashino, K. Generative Adversarial Image Synthesis with Decision Tree Latent Controller. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6606–6615.
32. Alharbi, Y.; Wonka, P. Disentangled Image Generation through Structured Noise Injection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2020; pp. 5134–5142.
33. Goyal, A.; Bengio, Y. Inductive biases for deep learning of higher-level cognition. *R. Soc. Math. Phys. Eng. Sci.* **2022**, *478*, 20210068. [CrossRef]
34. Arandjelović, R.; Zisserman, A. Object Discovery with a Copy-Pasting GAN. *arXiv* **2019**, arXiv:cs/1905.11369.
35. Azadi, S.; Pathak, D.; Ebrahimi, S.; Darrell, T. Compositional GAN: Learning Image-Conditional Binary Composition. *Int. J. Comput. Vis.* **2020**, *128*, 2570–2585. [CrossRef]
36. Sbai, O.; Couprie, C.; Aubry, M. Surprising Image Compositions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 3926–3930.
37. Burgess, C.P.; Matthey, L.; Watters, N.; Kabra, R.; Higgins, I.; Botvinick, M.; Lerchner, A. MONet: Unsupervised Scene Decomposition and Representation. *arXiv* **2019**, arXiv:cs/1901.11390
38. Greff, K.; Kaufman, R.L.; Kabra, R.; Watters, N.; Burgess, C.; Zoran, D.; Matthey, L.; Botvinick, M.; Lerchner, A. Multi-Object Representation Learning with Iterative Variational Inference. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 10–15 June 2019; Volume 97, pp. 2424–2433.
39. Liao, Y.; Schwarz, K.; Mescheder, L.; Geiger, A. Towards Unsupervised Learning of Generative Models for 3D Controllable Image Synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5871–5880.
40. Nguyen-Phuoc, T.H.; Richardt, C.; Mai, L.; Yang, Y.; Mitra, N. BlockGAN: Learning 3D Object-aware Scene Representations from Unlabelled Images. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*; NIPS: La Jolla, CA, USA, 2020; Volume 33, pp. 6767–6778.
41. Niemeyer, M.; Geiger, A. GIRAFFE: Representing Scenes As Compositional Generative Neural Feature Fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 11453–11464.
42. Li, N.; Eastwood, C.; Fisher, R. Learning Object-Centric Representations of Multi-Object Scenes from Multiple Views. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*; NIPS: La Jolla, CA, USA, 2020; Volume 33, pp. 5656–5666.
43. Anciukevicius, T.; Lampert, C.H.; Henderson, P. Object-Centric Image Generation with Factored Depths, Locations and Appearances. *arXiv* **2020**, arXiv:cs/2004.00642.
44. Henderson, P.; Lampert, C.H. Unsupervised Object-Centric Video Generation and Decomposition in 3D. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*; NIPS: La Jolla, CA, USA, 2020; Volume 33, pp. 3106–3117.
45. Singh, K.K.; Ojha, U.; Lee, Y.J. FineGAN: Unsupervised Hierarchical Disentanglement for Fine-Grained Object Generation and Discovery. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 6490–6499.
46. Schwarz, K.; Liao, Y.; Niemeyer, M.; Geiger, A. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*; NIPS: La Jolla, CA, USA, 2020; Volume 33, pp. 20154–20166.
47. Kwak, H.; Zhang, B.T. Generating Images Part by Part with Composite Generative Adversarial Networks. *arXiv* **2016**, arXiv:cs/1607.05387.
48. Chen, W.; Hays, J. SketchyGAN: Towards Diverse and Realistic Sketch to Image Synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 9416–9425.
49. Lu, Y.; Wu, S.; Tai, Y.W.; Tang, C.K. Image Generation from Sketch Constraint Using Contextual GAN. In Proceedings of the European Conference on Computer Vision (ECCV), Salt Lake City, UT, USA, 18–22 June 2018; pp. 205–220.
50. Zhao, J.; Xie, X.; Wang, L.; Cao, M.; Zhang, M. Generating Photographic Faces From the Sketch Guided by Attribute Using GAN. *IEEE Access* **2019**, *7*, 23844–23851. [CrossRef]
51. Ghosh, A.; Zhang, R.; Dokania, P.K.; Wang, O.; Efros, A.A.; Torr, P.H.S.; Shechtman, E. Interactive Sketch & Fill: Multiclass Sketch-to-Image Translation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1171–1180.
52. Chen, S.Y.; Su, W.; Gao, L.; Xia, S.; Fu, H. DeepFaceDrawing: Deep Generation of Face Images from Sketches. *ACM Trans. Graph.* **2020**, *39*, 72:1–72:16. [CrossRef]
53. Richardson, E.; Alaluf, Y.; Patashnik, O.; Nitzan, Y.; Azar, Y.; Shapiro, S.; Cohen-Or, D. Encoding in Style: A StyleGAN Encoder for Image-to-Image Translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 2287–2296.
54. Wang, S.Y.; Bau, D.; Zhu, J.Y. Sketch Your Own GAN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 14050–14060.
55. Chen, Q.; Koltun, V. Photographic Image Synthesis with Cascaded Refinement Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1511–1520.

56. Liu, X.; Yin, G.; Shao, J.; Wang, X.; Li, h. Learning to Predict Layout-to-image Conditional Convolutions for Semantic Image Synthesis. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*; NIPS: La Jolla, CA, USA, 2019; Volume 32, pp. 568–578.
57. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-To-Image Translation With Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
58. Tang, H.; Xu, D.; Sebe, N.; Wang, Y.; Corso, J.J.; Yan, Y. Multi-Channel Attention Selection GAN With Cascaded Semantic Guidance for Cross-View Image Translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 2417–2426.
59. Wang, Y.; Qi, L.; Chen, Y.C.; Zhang, X.; Jia, J. Image Synthesis via Semantic Composition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, Canada, 11–17 October 2021; pp. 13749–13758.
60. Zhu, Z.; Xu, Z.; You, A.; Bai, X. Semantically Multi-Modal Image Synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 5467–5476.
61. Zhu, P.; Abdal, R.; Qin, Y.; Wonka, P. SEAN: Image Synthesis With Semantic Region-Adaptive Normalization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 5104–5113.
62. Wang, T.C.; Liu, M.Y.; Zhu, J.Y.; Tao, A.; Kautz, J.; Catanzaro, B. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8798–8807.
63. Chen, A.; Liu, R.; Xie, L.; Chen, Z.; Su, H.; Yu, J. SofGAN: A Portrait Image Generator with Dynamic Styling. *ACM Trans. Graph.* **2022**, *41*, 1–26. [CrossRef]
64. Thanh-Tung, H.; Tran, T. On Catastrophic Forgetting and Mode Collapse in Generative Adversarial Networks. *arXiv* **2020**, arXiv:cs/1807.04015.
65. van den Oord, A.; Kalchbrenner, N.; Kavukcuoglu, K. Pixel Recurrent Neural Networks. In Proceedings of the International Conference on Machine Learning (ICML), New York City, NY, USA, 19–24 June 2016; Volume 48, pp. 1747–1756.
66. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4401–4410.
67. Mescheder, L.M.; Geiger, A.; Nowozin, S. Which Training Methods for GANs do Actually Converge? In Proceedings of the International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 3478–3487.
68. Lee, C.; Liu, Z.; Wu, L.; Luo, P. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, Washington, USA, 14–19 June 2020; pp. 5548–5557.
69. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*; NIPS: La Jolla, CA, USA, 2017; Volume 30, pp. 6626–6637.
70. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X.; Chen, X. Improved Techniques for Training GANs. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*; NIPS: La Jolla, CA, USA, 2016; Volume 29, pp. 2226–2234.
71. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
72. Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; Tang, X. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1096–1104.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.