applied sciences

MDPI

Article

Conditional Generative Adversarial Networks for Domain Transfer: A Survey

Guoqiang Zhou [1,*], Yi Fan [1], Jiachen Shi [1], Yuyuan Lu [2] and Jun Shen [3,*]

[1] College of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210046, China
[2] State Key Laboratory of Polymer Physics and Chemistry, Changchun Institute of Applied Chemistry, Chinese Academy of Sciences, Changchun 130022, China
[3] School of Computing and Information Technology, University of Wollongong, Northfields Ave, Wollongong, NSW 2522, Australia
* Correspondence: zhougq@njupt.edu.cn (G.Z.); jshen@uow.edu.au (J.S.)

Abstract: Generative Adversarial Network (GAN), deemed as a powerful deep-learning-based silver bullet for intelligent data generation, has been widely used in multi-disciplines. Furthermore, conditional GAN (CGAN) introduces artificial control information on the basis of GAN, which is more practical for many specific fields, though it is mostly used in domain transfer. Researchers have proposed numerous methods to tackle diverse tasks by employing CGAN. It is now a timely and also critical point to review these achievements. We first give a brief introduction to the principle of CGAN, then focus on how to improve it to achieve better performance and how to evaluate such performance across the variants. Afterward, the main applications of CGAN in domain transfer are presented. Finally, as another major contribution, we also list the current problems and challenges of CGAN.

Keywords: conditional generative adversarial network; loss function; cycle consistency; progressive enhancement mechanism; domain transfer

Citation: Zhou, G.; Fan, Y.; Shi, J.; Lu, Y.; Shen, J. Conditional Generative Adversarial Networks for Domain Transfer: A Survey. *Appl. Sci.* **2022**, *12*, 8350. https://doi.org/10.3390/app12168350

Academic Editor: Christos Bouras

Received: 21 July 2022
Accepted: 18 August 2022
Published: 21 August 2022

1. Introduction

Since the booming development of deep learning, neural networks of various architectures have emerged in an endless stream. Domain transformation is one of the most important problems for which deep learning is qualified. Generally speaking, domain transfer is the process of moving data from one domain to another by learning the similarities between the two domains and building a bridge between them to apply the old knowledge to the new domain so that the new knowledge can be learned faster and better. It can further address the image/scene gap during training and practical application. It can change one aspect of the data while preserving others. In recent years, the domain transfer method has been applied to solving many practical problems, such as geoscientific inverse problems [1], maize residue segmentation [2], person re-identification [3], data augment in face recognition [4] and age estimation [5]. Because of the complexity of high-dimensional data, it is not easy to separate the features that need to change from those remaining unchanged. However, the emergence of Conditional Generative Adversarial Networks (CGAN) greatly improves the effect of the domain transfer. With the assistance of CGAN, domain transfer can be easily applied in different application scenarios. Therefore, this article will review the problems of domain transfer based on CGAN. In fact, CGAN could also be used for other application fields, such as data augmentation, image inpainting and so on. While their principles are similar, it is very necessary to conduct a review on CGAN. For the sake of illustration, we will first review Generative Adversarial Networks (GAN) without conditions and then expand the introduction to CGAN.

In 2014, Goodfellow et al. proposed GAN for the first time [6], providing a new method for generating realistic images. Generally speaking, GAN consists of two independent

networks: generative network and discriminative network. The generative network is used to generate an image from a random vector, while the discriminative network determines whether an image is a real image from a data set or a fake image generated by the generative network. The framework of GAN seems to be quite different from deep learning algorithms for classification [7] or regression [8], but they have some common features [9,10]. The main body of GAN is barely the generative network, while the discriminative network can be viewed as a complex loss function calculator, which needs to go through a process of training rather than only a few simple arithmetic calculations, such as Mean Square Error (MSE) [11]. On the other hand, to improve the training effect of GAN, the discriminative network is not trained to the convergent state in the process of calculating the loss function. In other words, the training of the generative network and the discriminative network are alternately carried out.

However, GAN has a serious disadvantage; that is, it cannot control the output of the generative network [12]. For example, the MINIST dataset is composed of handwritten digits from zero to nine [6]. After completing the training, a random vector is fed into GAN, and the output may be any one of zero to nine, which cannot be predicted in advance. This greatly affects the practicability of GAN. To solve this problem, CGAN was proposed in [13]. CGAN introduces a conditional variable, which is used to control the behavior of the generative network, making the output constrained to the user-specified distribution and ensuring its stability. However, conditional variables do not completely fix the generated content of the generative network. The input of the generative network still contains random vectors, and the generated results are still diverse. The stability and diversity of CGAN's generated results make it rather promising. In recent years, many institutions have carried out research on CGAN, made various improvements [14–16] and applied it to various domain transfer problems [17–21], and even spread them to various fields outside this traditional field [22,23]. At present, a large amount of surveys have summarized the models of GANs, but few of them concentrated on the topic of CGANs in domain transfer. Therefore, this paper will first summarize the research on CGAN and further discuss the problems and challenges facing it, aiming to encourage more researchers to develop CGANs. We have inspected almost 190 papers manually about CGANs, while some of them are not officially published or have similar ideas. Therefore, the final listed citations are carefully selected, including papers published in SCI journals or conferences, papers widely recognized by peer researchers (or highly cited papers) and a few novel interesting papers. In the meantime, our analysis and findings can be generalized to those new fields. Our main contributions can be summarized as follows:

1. This paper first summarizes the research on the CGAN in the domain transfer.
2. This paper summarizes the CGAN from different aspects, such as the loss function, the model variants, application fields and so on, so that readers can easily understand the situation of current research.
3. This paper discusses the remaining problems and challenges of CGAN and provides the possible future directions for the development of CGAN.

The structure of this article is as follows. In Section 2.1, the theoretical basis of CGAN will be introduced, and then in Section 2.2, we will focus on some typical improved versions of CGAN. In Section 3, the main evaluation methods will be summarized. In Section 4, applications of CGAN in various fields of domain transfer will be discussed in detail, focusing on the problems faced in specific scenarios and their corresponding solutions. In Section 5, we list some crucial problems and challenges of CGAN to date. Finally, the future development of CGAN will be analyzed in Section 6.

2. The Development Methods of CGAN

2.1. The Principle of CGAN

CGAN is directly derived from GAN. In order to better introduce CGAN, the principle of GAN is herein briefly reviewed. In GAN, there are two networks—generative network G and discriminative network D, in which G is used to establish the data distribution, and D

is used to estimate the probability that a sample comes from the dataset or G. In order to obtain the distribution, p_g, of the dataset, $\boldsymbol{x}$, G will establish a mapping from the prior noise distribution, $p_z(\boldsymbol{z})$, to the data space, $\boldsymbol{x}$; G outputs a scalar, $D(x;\theta_d)$ to represent the probability that $\boldsymbol{x}$ comes from the dataset. During the training, the parameter updates of G and D are carried out simultaneously. When adjusting the parameters of G, the goal is to minimize $\log(1 - D(G(z)))$, and when adjusting the parameters of D, the goal is converted to $\log D(G(z))$. They counteract each other and optimize the following functions in the opposite direction:

$$\begin{aligned}\min_G \max_D V(D,G) =& \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] \\ &+ \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]\end{aligned} \tag{1}$$

The generic framework of GAN could be seen in Figure 1. If the above generative network and discriminative network are conditional on some additional information $\boldsymbol{y}$, then GAN will be extended to CGAN. Here, $\boldsymbol{y}$ can be any additional information, such as category labels or other modal data. In the design of network architecture, $\boldsymbol{y}$ is often used as the input for both the generative network and discriminative network. For G, the prior noise $p_z(\boldsymbol{z})$ and the condition $\boldsymbol{y}$ are concatenated for the hidden layer; for D, sample data or generated data $\boldsymbol{x}$ and condition $\boldsymbol{y}$ are concatenated as the input. Therefore, the objective function becomes:

$$\begin{aligned}\min_G \max_D V(D,G) =& \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] \\ &+ \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))]\end{aligned} \tag{2}$$

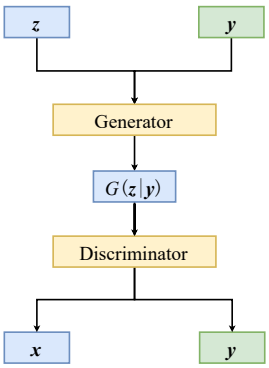


Figure 1. The generic framework of GAN. Compared with GAN, it also takes the condition $\boldsymbol{y}$ as the input of the generator and discriminator.

2.2. The Improvement of CGAN

When CGAN is applied to the computer vision field, image transformation can be completed if the condition is specified as an image. To be specific, given two types of image data (respectively, represented as domain X and domain Y), the input of the generative network will be random vector z and samples $\boldsymbol{x}$ of the domain X during the training process. After achieving the output image, we compare it with the corresponding sample $\boldsymbol{y}$ of the domain Y. However, this method faces many difficulties in practical application, so many scholars have tried to use different technologies to improve the framework or training mode. This section reviews these efforts so far.

2.2.1. The Loss Function of CGAN

As mentioned above, the main part of GAN is the generative network, while the discriminative network is only a complex loss function. The same is true of CGAN. However, maximizing the classification error of the discriminative network will only have a limited effect on CGAN. Faced with a variety of complex application scenarios, researchers

have also proposed many other loss functions for the training of CGAN. Representing the parameters in the CGAN as $\boldsymbol{\theta}$, and then these loss functions as $L_1(\boldsymbol{x};\boldsymbol{\theta})$, $L_2(\boldsymbol{x};\boldsymbol{\theta})$, ..., $L_n(\boldsymbol{x};\boldsymbol{\theta})$, then the training of CGAN becomes a multi-objective optimization problem, using the weighted sum of each loss as the final loss function,

$$L(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{i=1}^{n} \eta_i L_i(\boldsymbol{x};\boldsymbol{\theta}), (\sum_{i=1}^{n} \eta_i = 1). \tag{3}$$

Here, η_i represents the coefficient. The losses commonly used except for discriminant losses are described below.

1. Content loss: This loss is primarily used to measure the gap between the generated image and the user's expectations. For example, when training CGANs whose task is to transfer gray-level images to color images, the general strategy is to first render a set of color images grayscale and then use the paired gray level image $\boldsymbol{y}$ and color image $\boldsymbol{x}$ as training data. During the training process, when the generative network inputs gray level images and outputs color images, users usually expect the output results to be as close as possible to the corresponding color image in the training set. Therefore, the loss is expressed as the difference between the values of each pixel in the generated images and the training data, which is generally characterized by mean square error (MSE),

$$L_{\text{content}}(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{CHW} \sum_{i=1}^{C} \sum_{j=1}^{H} \sum_{k=1}^{W} (G(\boldsymbol{y})_{ijk} - \boldsymbol{x}_{ijk})^2. \tag{4}$$

   Here C, H and W, respectively, represent the number of channels, height and width of the image, $\cdot_{ijk}$ represents the i-th channel, j-th row and k-th column of a sample. However, it is worth noting that excessive increases in this loss will lead to image blurring.
2. Perception loss: Although content loss can monitor the content of the generated image, it gives more consideration to the underlying semantics of the image due to its pixel-by-pixel differential mechanism. In order to monitor the high-level semantic consistency between the generated image and the actual image, perception loss is often used. In essence, it is still the MSE of two tensors, except that the two tensors are no longer about the image itself, and instead, the feature map is derived from the images with a certain model. For example, ESRGAN [14] takes the generated image and the actual image as the input of the VGG19 network [24], respectively, and then calculates the MSE of hidden variables of each layer in the middle of the VGG19 network. Generally, the expression of the loss is:

$$\begin{aligned} &L_{\text{perception}}(\boldsymbol{x};\boldsymbol{\theta}) \\ &= \frac{1}{CHW} \sum_{i=1}^{C} \sum_{j=1}^{H} \sum_{k=1}^{W} (F(G(\boldsymbol{y}))_{ijk} - F(\boldsymbol{x})_{ijk})^2. \end{aligned} \tag{5}$$

   Here, similarly, C, H and W, respectively, represent the number of channels, height and width of the image, $\cdot_{ijk}$ represents the i-th channel, j-th row and k-th column of a sample, and F can be any model.
3. Hidden variable loss: Both content loss and perception loss are evaluated by the final images generated by the generated network. However, when the network is deep, the idea of end-to-end learning can swiftly cause the problem of gradient disappearance or gradient explosion. For this reason, we would consider monitoring hidden variables in the network. Since hidden variables do not have a dataset as the "standard answer", it is necessary to design the network architecture in a reasonable way to construct two hidden variables with the same semantics. For example, in the task of improving image resolution, if the generative network is designed to resemble

an auto-encoder network, two hidden variables on the symmetry of the bottleneck can be considered to represent the same semantic information, and the difference between them can be designed as the loss of hidden variables. The expression of the loss is

$$L_{\text{latent}}(\boldsymbol{x};\boldsymbol{\theta}) = D_f(l_1 \| l_2), \tag{6}$$

where l_1 and l_2 are the distributions of two hidden variables, and D_f is f-divergence. The specific type of f-divergence can be selected according to different tasks.

4. Category loss: In some scenarios, the user does not require the generated image to be similar to an expected image, as long as it has the correct category. For example, after training with MNIST, the user requires the network to generate the specified number but does not require the handwriting characteristics of the number. For these requirements, many CGAN models put the generated images into a pre-trained classifier. Therefore, the expression of this kind of loss is similar to the common classification problem, and cross-entropy loss is commonly used. The expression of the loss is

$$L_{\text{class}}(\boldsymbol{x};\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{z},c}[-\log D_{\text{cls}}(c|G(\boldsymbol{z},\boldsymbol{y}))], \tag{7}$$

where D_{cls} represents the classifier and c represents the category to which the image belongs.

The above four kinds of losses are only representing typical usage in CGAN, and there is no clear boundary between them. Their differences are mainly reflected in independent variables, calculation methods, etc. Table 1 summarizes them as a whole.

Table 1. The comparison of different loss functions.

Loss Function	**Content Loss [25]**	**Perception Loss [14]**	**Hidden Variable Loss [26]**	**Category Loss [27]**
Source of independent variables	The output of generative network and training data		Two latent variables in the network	The output of generation network and condition
Data type of independent variables	Image tensors		Distribution	
Whether to use additional networks	No	Yes	No	Yes
Metric	Error (usually MSE)		Divergence	Divergence (usually cross-entropy loss)
Contents of supervision	Low-level semantics of generated images	High-level semantics of generated images	States of generative network	Class of generated images

In most of the existing results, all the above losses are not applied simultaneously. The usual practice is to select 2–3 losses to build the loss function according to the specific requirements of the task. Table 2 summarizes the usage of these losses in some typical works. As can be seen from Table 2, most CGAN models use content loss. That said, in most application scenarios, supervision of the low-level semantics is necessary.

Table 2. The usage of loss functions in various methods.

Methods	Adversarial Loss	Content Loss	Perception Loss	Hidden Variable Loss	Category Loss
Laplacian Pyramid [28]	✓	✓			
SRGAN [29]	✓	✓			
Amortised MAP Inference [30]	✓				
ESRGAN [14]	✓	✓	✓		
Pixel-Level Domain Transfer [31]	✓				✓
Markovian GAN [32]	✓		✓		
Generative Visual Manipulation [33]	✓	✓	✓		
ICGAN [34]	✓	✓			
pix2pix [35]	✓	✓			
Scribbler [36]	✓	✓		✓	
CycleGAN [15]	✓	✓			
DiscoGAN [26]	✓	✓			
DualGAN [37]	✓	✓			
pix2pixHD [25]	✓			✓	
MUNIT [38]	✓	✓			
StarGAN V2 [39]	✓	✓			✓
ExprGAN [40]	✓	✓	✓		✓
AttGAN [19]	✓	✓			✓
STGAN [41]	✓	✓	✓		
SynCGAN [42]	✓	✓			✓
StackGAN [16]	✓				
StackGAN++ [43]	✓			✓	
MirrorGAN [44]	✓		✓	✓	
DM-GAN [45]	✓		✓		✓

2.2.2. Cycle Consistency

The premise of the successful training of the above image transformation model is that the data between the two domains have certain relationships. However, in many actual tasks, such as converting a picture of a man to a picture of a woman with the same face, such a requirement is difficult to meet. In this case, without the criteria to judge the quality of the generated data, many of the loss functions mentioned above lack data sources. That is to say, it is impossible to improve the network performance only by optimizing the loss function. The objective of optimization should be the network framework rather than the loss function. For this reason, some researchers used the idea of cyclic consistency to solve this problem and designed different CGAN frameworks, typical of which included CycleGAN [15], StarGAN [27], DTN [46] and XGAN [47].

CycleGAN

The framework of CycleGAN is shown in Figure 2. It consists of two generative networks and two discriminative networks. Let G become a generative network converting an image from domain X to domain Y. After the original input $\boldsymbol{x}$ obtains the output $\hat{\boldsymbol{Y}}$ through G, if $\hat{\boldsymbol{Y}}$ is fed back into F, the original input $\boldsymbol{x}$ should be retrieved. This is equivalent to having the image loop back to the starting point and keeping it consistent. Its total loss consists of four parts: firstly, two losses formed by two discriminative networks will guide the generative networks to generate more realistic images for the target domain, which is consistent with the loss function corresponding to raw GAN; secondly, two loop losses will guide the generative networks to generate images that are as close as possible to the input image. Its loss function is

$$\begin{aligned} L_{\text{cyc}}(G,F) =& \mathbb{E}_{\boldsymbol{x}\sim P_{\text{data}}(\boldsymbol{x})}[\|F(G(\boldsymbol{x}))-\boldsymbol{x}\|_1] \\ &+ \mathbb{E}_{\boldsymbol{y}\sim P_{\text{data}}(\boldsymbol{y})}[\|G(F(\boldsymbol{y}))-\boldsymbol{y}\|_1], \end{aligned} \tag{8}$$

where $\|\cdot\|_1$ is l_1 norm.

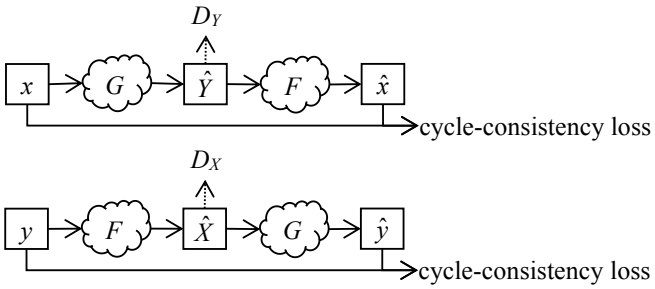


Figure 2. The framework of CycleGAN. It consists of two generative networks and two discriminative networks. G is a converter from domain X to Y, and F is the reverse. The cycle-consistency loss monitors whether the data are distorted in the transformation of X-Y-X and Y-X-Y.

StarGAN

CycleGAN uses the idea of cycle consistency to realize the conversion of images on two domains. However, in this method, a target domain requires a large number of generative networks, namely, n domains require $\frac{1}{2}n(n-1)$ generative networks. In addition to high complexity, inadequate utilization of data may lead to the unsatisfactory quality of generated images, and it is impossible to train domains together from different data sets. Therefore, reference [27] proposes StarGAN to solve the problem of multi-domain transformation, which only trains a single generator with data from multiple domains to ensure the generated images contain different domain styles.

The training process of StarGAN is shown in Figure 3. It also uses two constraints, conditional generative adversary and cyclic consistency. Figure 3a represents the training of the discriminative network. The real images and the generated images are transferred to the discriminative network, which needs to determine the authenticity of all images and their corresponding domains. Figure 3b represents the training of the generative network, where the generator receives both the input images and the target domain label. Figure 3c represents the process of cycle consistency. It will take the image generated by the previously generated network with the original domain label as the input of the generated network. Figure 3d represents the training of the discriminator, which aims to distinguish whether the image is real or not and whether in the correct domain.

In short, in order to realize the mutual transformation of multi-domain images, the generative network must receive two kinds of information, the input images and the target domain to be transformed. To this end, the target domain information is equivalent to the conditional constraint of the generated network. Cycle consistency is used to avoid a situation where the generated images and the input images have nothing to do with each other.

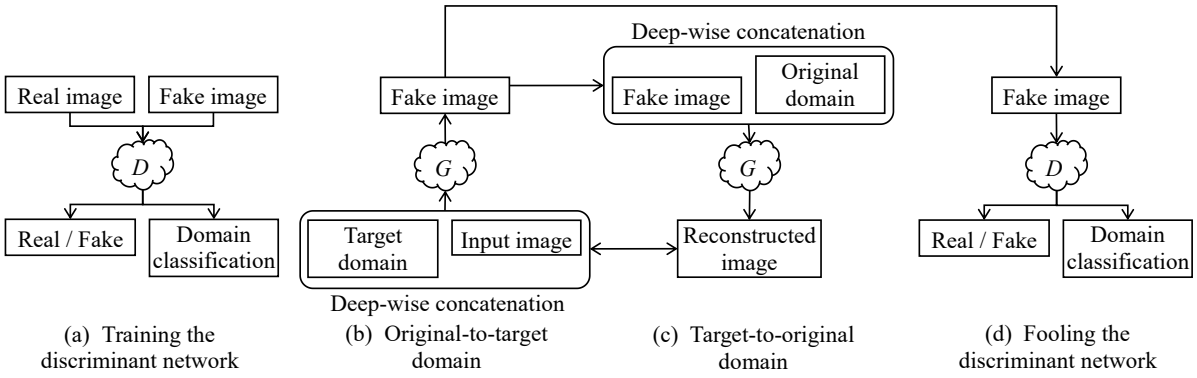


Figure 3. The framework of StarGAN. (**a**) *D* is composed of two modules and receives the real images and fake images. One is to distinguish real images from fake ones, and the other is to classify them into different domains. (**b**) *G* receives the target domain and input data and generates fake images. (**c**) *G* reconstructs the raw images using the fake image and original domain information. (**d**) *D* distinguishes the source of the image and its domain. A well-trained *G* can make *D* fail to complete this task.

DTN

The cycle transformation in CycleGAN and StarGAN is relatively simple; in order to make this process dig out more information, DTN [46] introduces perceptual loss and strives to produce images that are consistent with the target on high-level semantics. This strategy allows DTN to achieve linear regression of significantly different images. The DTN framework is shown in Figure 4.

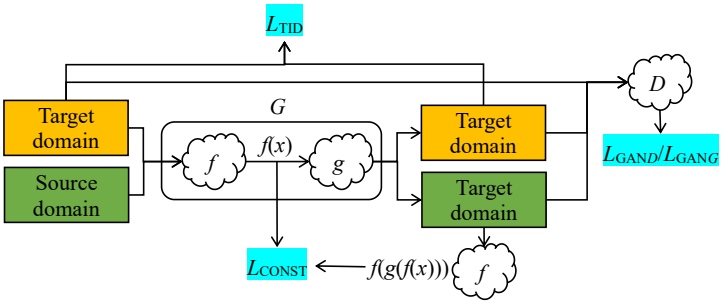


Figure 4. The framework of DTN. The generative network *G* is composed of feature extractor f and image generative network g, where f will extract the image features of the input image, and g is used to generate the image. As to loss function, L_{const} supervises the difference of features between two domains, L_{TID} supervises the difference between output and input, and $L_{\text{GAND}}/L_{\text{GANG}}$ distinguishes the authenticity of the images.

Normally, DTN consists of a generative network and a discriminative network. Among them, the generative network is composed of feature extractor f and image generator g. f is used to extract the image features of the input image, which is actually the semantic style of the image, and then g is used to generate the images.

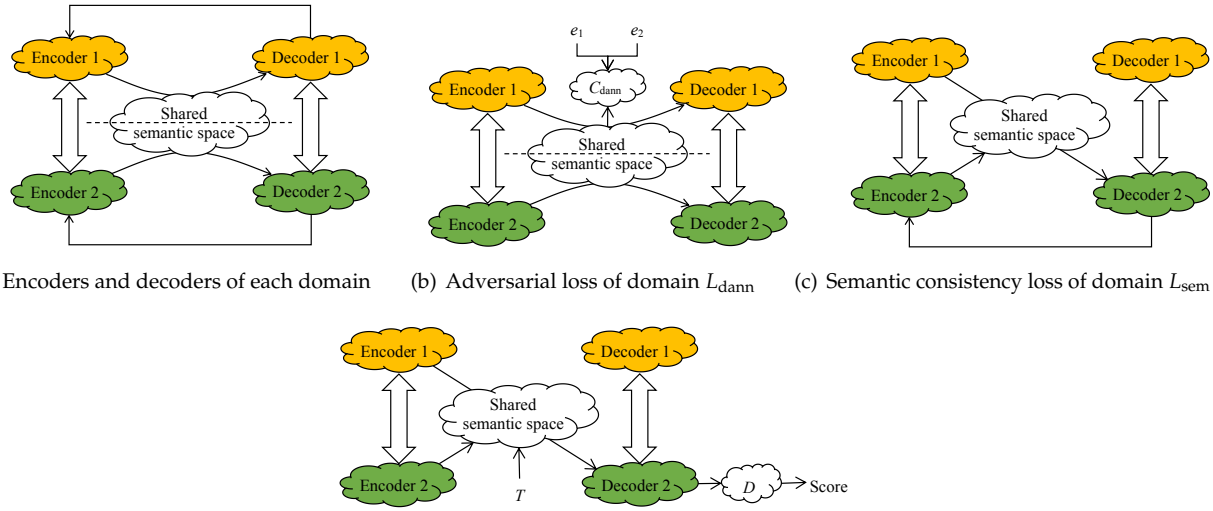
Two types of images will be taken as the input data: the source domain images (e.g., real face) and the target domain images (e.g., cartoon face). Firstly, the network will generate target domain images with similar semantic feature information. In order to ensure the generated images have a semantic similarity with source domain images, DTN will send the generated images to feature extractor f again to render it for extraction of the feature, and then compare this feature with that of the source domain images. The difference between them will be set as the loss function, i.e., L_{CONST}, which will be minimized to guarantee the feature matching. Secondly, in order to facilitate the generative network capturing the characteristics of target domain images, the target domain images will be

transferred to themselves by the generative network, minimizing the difference between them, i.e., L_{TID}.

For the discriminative network, it takes in three different types of images: real source domain images, target domain images generated from the target domain and target domain images generated from the source domain. The discriminant network will judge whether the input image is realistic, which is rated as a high score or a low score and vice versa.

XGAN

Generally speaking, in order to simplify the calculation of the consistent cyclic loss, most researchers usually encode the related variables to a fixed length, while this method does not accommodate the generality when there is a large domain displacement between two domains. Therefore, XGAN [47] establishes two auto-encoders in two domains, and both domains have an encoder and an decoder, respectively, as shown in Figure 5a, to solve this problem. Specifically, Encoder 1 and Decoder 1 were combined as an auto-encoder in domain x to obtain the semantic characteristics of the images here. Domain y has the same constitution (Encoder 2 and Decoder 2). It should be noted that the last several layers of Encoder 1 and Encoder 2 are associated with each other so that the associate semantic features can be produced when encoders encode the images in their own domain. The same occurs to Decoder 1 and Decoder 2 to allow the two domains to generate relevant semantic features.



(a) Encoders and decoders of each domain (b) Adversarial loss of domain L_{dann} (c) Semantic consistency loss of domain L_{sem}

(d) Adversarial loss

Figure 5. The framework of XGAN. Generally speaking, it is composed of two encoders and two decoders. As for the loss function, L_{dann} assures the quality of the encoding, and L_{sem} preserves the semantics after domain transfer. Specifically, in (**a**), each domain in XGAN has their own encoder-and-decoder. In (**b**), the domain adversarial loss is used to moderate the differences in semantic features between two domains. In (**c**), the semantic consistency loss is used to ensure that the semantic features generated have the same meanings in different domains. (**d**) Is the whole architecture of XGAN.

In order to strengthen the semantic relationship in two domains, XGAN defines the domain adversarial loss, L_{dann}, to embed the image features of two domains into the same subspace and moderate the differences in semantic features between two domains, as shown in Figure 5b. From this, we could also see that the images in different domains are still encoded by their own encoders to obtain the semantic features, which would be then decoded by their decoders to restore the image. The binary classification model C_{dann} is trained to classify the resource of the semantic features in the current semantic space, namely from Encoder 1 or Encoder 2. Simultaneously, Encoder 1 and Encoder 2 are aimed to prevent C_{dann} from being identified by decreasing the accuracy of the classification,

which could, in turn, enable them to represent images with similar semantic features in their own domains.

However, the above mechanism could only ensure that the semantic features generated between the images of the two domains are similar in phenotype but cannot indicate that they have the same meanings in different domains. For example, suppose that the semantic vectors generated by the encoders in the two domains are similar, where domain X means people with black hair and big eyes are a little angry and domain Y means people with golden hair and small eyes are laughing happily—they represent two totally different internal meanings. Therefore, XGAN defines semantic consistency loss, i.e., L_{sim}, to solve this problem, and its intuitive form is shown in Figure 5c. It can be seen that the images in domain X will be obtained with semantic features by encoder 1 first, and then be decoded by decoder 2 in the domain Y. Now the restored image will be sent to encoder 2 to obtain semantic features by encoder 2. Minimizing the difference between the restored image and the original input image will make the semantic features in the two domains require relevance in meaning. In a word, the architecture of XGAN is shown in Figure 5d.

Furthermore, the main purpose of this subsection is to allow readers to gain a generic understanding of CGAN and some typical variants of CGAN as much as possible. While most typical related works are discussed in this paper, we can not list many other variants of CGAN to uphold the focus of our contribution in this paper. On the other hand, some other variants of CGAN could be found in some relevant papers [48–54] if a reader is truly interested in an exhaustive search of all CGAN variants.

2.2.3. Progressive Enhancement Mechanism

CGAN can be used to generate images, but these images are usually 64×64 or 128×128, making it difficult to obtain the desired effect with larger images. One intuitive reason is that more information needs to be learned to generate large images, and it is difficult for GAN to generate a large amount of information within one breath so that GAN will produce some unnatural phenomena, such as distorted or blurred outputs, when generating large images. A progressive enhancement mechanism can gradually generate high-resolution images from low-resolution images, which has a good application prospect in the generation of high-resolution images. Hence, herein, we will briefly introduce some typical methods based on progressive enhancement mechanisms, such as StackGAN [16], StackGAN++ [43] and PGGAN [55].

StackGAN

The core idea of StackGAN is to use two generative networks and two discriminative networks to generate the final image hierarchically. As shown in Figure 6, StackGAN contains two layers, each of which is composed of a generative network and a discriminative network. The first layer is mainly responsible for producing small blurred images so that the generative network's "attention" is focused on the edges of the image and the overall structure rather than the details. Then, the fuzzy image generated by the first layer with the overall structure of the image is passed to the generative network of the next layer, which generates the image on the basis of the fuzzy image of the first layer. For the generative network in the second layer, it directly obtains the general information of the image edges, shapes and so on and does not have to pay any effort to generate the overall content. Instead, it focuses its "attention" on the details of the image, thus achieving the purpose of generating large-size images.

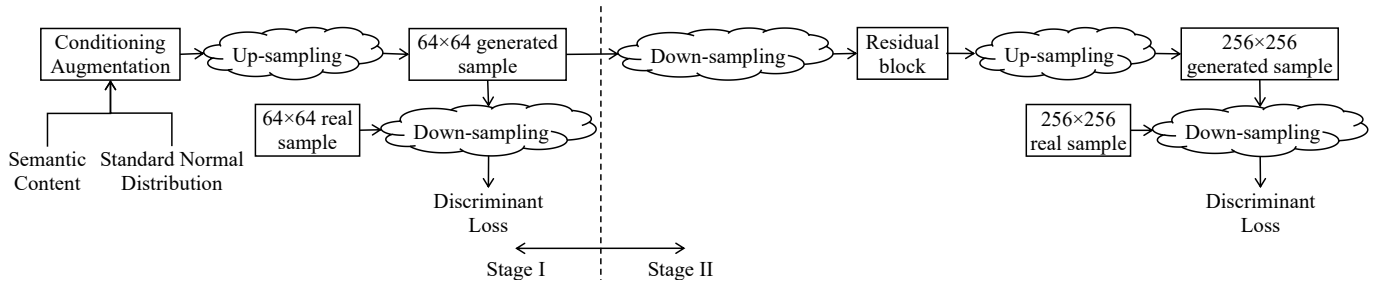


Figure 6. The framework of StackGAN. The generation process is composed of two stages. In Stage I, the generator generates the low-resolution images with rough shapes and colors from semantic content and standard normal distribution. In Stage II, the high-resolution images are formed by fine-tuning according to the results of Stage I.

StackGAN++

StackGAN++ is an improvement of StackGAN. Compared with StackGAN, it is no longer layered, in which the generative network adopts a tree structure and is composed of multiple generative networks that can generate different images. Each generative network corresponds to a discriminative network respectively to achieve coherent image generation.

The framework of StackGAN++ is shown in Figure 7, where the left part is the architecture of the generative network, and the right part is the architecture of the discriminative network.

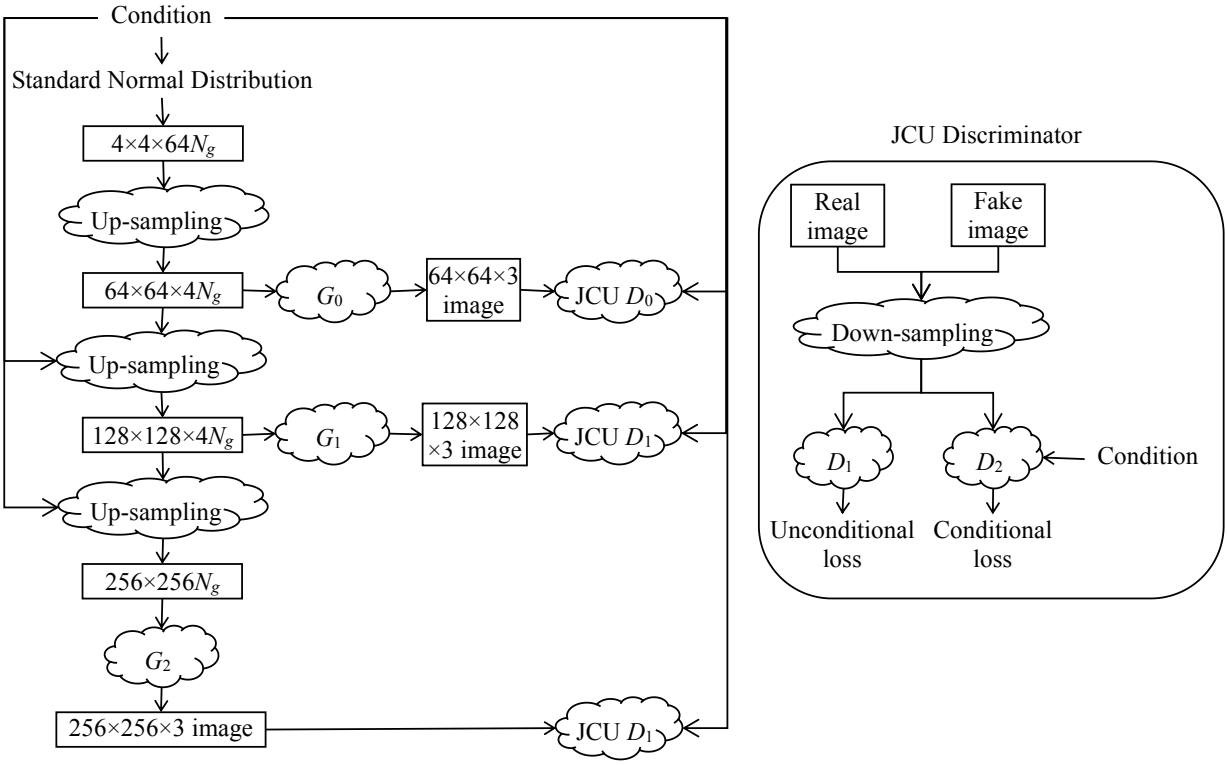


Figure 7. The framework of StackGAN++. The main difference against StackGAN is using more than one generative network and discriminative network. These networks form a tree, different branches of which correspond to different scales in the same scenario.

The generative network is a tree structure made up of three small generative networks that can produce images of different sizes. It can be seen from Figure 7 that the first generative network is responsible for generating a $64 \times 64 \times 3$ image, the second generative network generates a $128 \times 128 \times 3$ image from the first generative network, and the last generative network generates a $256 \times 256 \times 3$ image from the second generative network.

PGGAN

Although StackGAN can generate high-resolution images at the 256×256 level, the resolution is still hard to reach at 1024×1024. As another progressive enhancement network, PGGAN is expected to generate higher resolution images, and its framework is shown in Figure 8.

As can be seen from Figure 8, PGGAN only generates a 4×4 image at the beginning, but with the increase in network architecture, the resolution of the generated image is gradually improved. This incremental training enables the network to focus its attention on the general distribution of the image at the beginning. As the training proceeds, GAN's attention shifts from the general distribution of the image to the details of the image, thus avoiding the training difficulties caused by GAN in the process of learning all the information about the image at the same time.

Notably, during PGGAN training, the generative network and discriminative network are equivalent to mirror structures of each other, both of which grow simultaneously, and all existing components of the generative network and discriminative network are trainable throughout the training process. When a new layer is added to the original generative network and discriminative network, the previous layer is still trainable except for training the new layer, rather than only training the new layer.

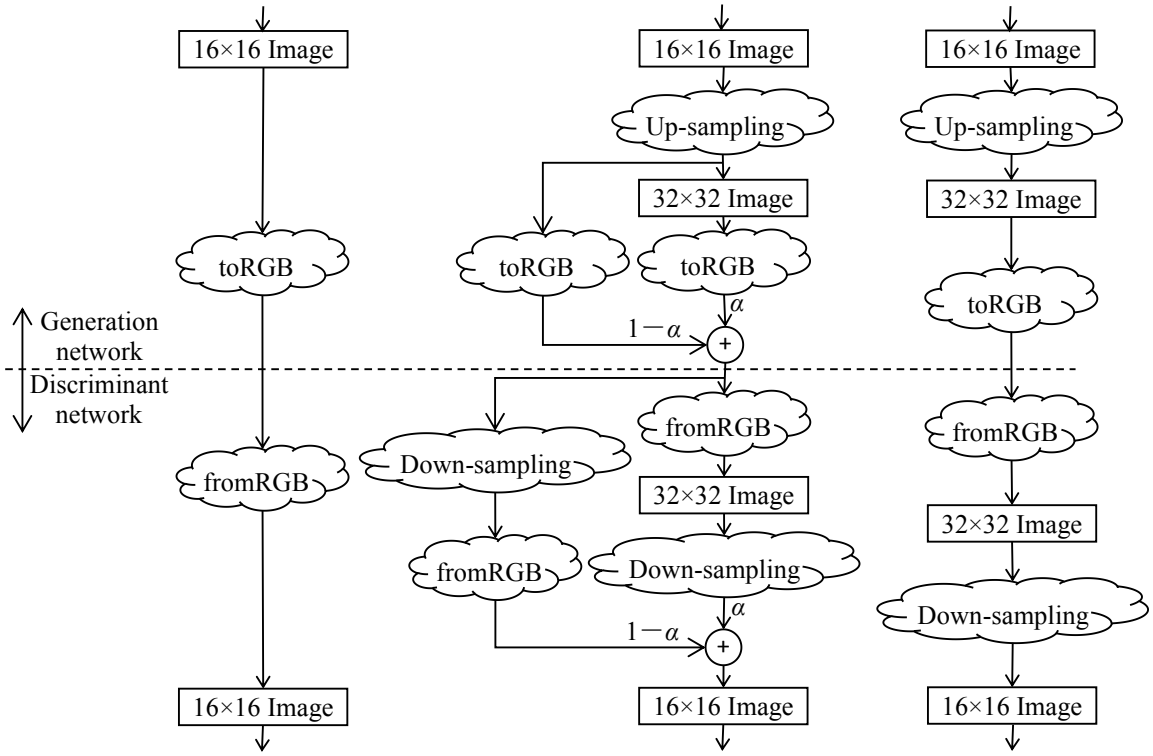


Figure 8. The framework of PGGAN. α will increase continuously during the training process, thus realizing a slow increase in the degree of upsampling. Note that this figure represents only one layer of the network, and the actual training process needs to stack several similar architectures.

3. The Evaluation Method of CGAN

In order to evaluate the performance of CGAN, it is necessary to design a fair evaluation method. In the field of image classification, the accuracy of image classification on the test set is almost the most convincing index. However, CGAN's main task in computer vision is to edit or synthesize images, which is a process of "creation of art", so it is difficult to use a unified index for evaluation. Hence, in order to evaluate the performance of CGAN as comprehensively as possible, different methods are used in different works. Generally speaking, these methods can be divided into two categories: automatic evaluation and manual evaluation.

3.1. Automatic Evaluation

Automatic evaluation refers to the evaluation index that can be calculated directly by using the formula without human participation. Some indexes in the original GAN, such as Inception Score (IS) [56] and Fréchet Inception Distance (FID) [57], can also be used to evaluate CGAN to measure the fidelity of the generated images [23,45,58–60]. Compared with the classical GAN, for CGAN, the expected image is usually given in some task, and the similarity between the generated image and the expected image is also an important index, such as the Target Attributes Recognition Rate (TARR), Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) [41,61,62].

As noted, CGAN's task is to generate correct images rather than only arbitrary images. We have also mentioned the diversity of the CGAN loss function in Section 2.2.1, and each loss function corresponds to whether the generated image meets the requirements in some aspects. For example, category loss is used to monitor whether the generated image conforms to the category specified in the input conditions. Therefore, the classification accuracy of the generated image can also be used as an evaluation index for CGAN. The hypothesis that the classifier trained on the real image can be well applied to the classification of the generated image once the quality of the generated image is good enough is employed here. For example, the indicator Dice introduced by [35,63] uses the pre-trained image segmentation model, FCN-8s, to segment the generated image and then calculates the distance between the segmentation result and the original semantic mask. The principles of [58,61] are similar. In addition to the discriminant method developed by category loss, reference [64] calculates the average distance of a group of random samples in the feature space, and also the Cosine Similarity (CSIM) of embedded vectors proposed in [65] has a similar effect.

For some cross-modal tasks, a more indirect approach is often required. For example, reference [45] mainly solves the task of converting text to image using R-precision. First, it needs 100 candidate texts, including R ground truth text and $100 - R$ random text. Then, it queries these candidate texts according to the generated images and counts the number of texts related to the generated images in the previous R text, that is r, and the value r/R is namely the R-precision.

The advantage of automatic evaluation is that it does not need manual intervention and can save on manpower. In addition, it can derive quantitative metrics for the quality of the generated data. However, its disadvantages are also obvious. On the one hand, due to the complexity of data generated, the evaluation content is diverse, and there is no absolutely reasonable index. For example, IS can reflect the proximity of the distribution between the generated data and the real data, but it cannot monitor the problem of overfitting. When the model can only generate data from the training set but cannot create data, IS may still remain at a high level. If one application relies too much on automatic evaluation, the problem of disconnection between the evaluation metrics and the actual effect will occur. On the other hand, as many evaluation metrics are involved in the pre-training model, the uncertainty of evaluation results will also increase.

3.2. Manual Evaluation

Manual evaluation is a supplement to automatic evaluation, which is helpful for finding out some factors that are not considered enough in the process of designing automatic indexes. One of the simplest methods is to select some typical samples from the generated data for human analysis [23,61,64]. However, this simple analysis method is only representative of the authors' or recruited analysts' opinions, with a large degree of randomness.

In order to make the evaluation results more objective, it is often necessary to invite several evaluators to evaluate independently. For example, in [64], several groups of images will be displayed, each of which contains a real image and a generated image, and the evaluator needs to choose the more likely real image based on his own judgment. There are several improved versions of this evaluation method. From the perspective of the evaluator, if the image involves professional content, the evaluator should be a domain expert [23]. From the perspective of the measurement scale, if we desire the evaluators to pay more attention to the overall image rather than local, the display time of the image can be limited (such as 1 second), and when the display time is depleted, the image immediately disappears. At this time, the evaluator needs to rely on the impression of the two images to choose the proper one [35]. From the point of view of evaluation results, if we want the evaluators to understand the attributes of real images to a deeper extent, there are mainly two schemes. In one way, we can demonstrate the answer to the evaluator at the beginning of the evaluation as feedback [35]. For another, we can display two real images and a generated image once at a time so that the evaluator will have more opportunities to perceive the attributes of the real image [65]. In addition, comparative experiments can also be conducted, in which the image generated by another algorithm replaces the real image, and the evaluator will choose the better image to attain the conclusion on which algorithm is better.

Similar to automatic evaluation, in the task of text-to-image generation, there are corresponding manual evaluation methods in order to determine whether the generated image is conditional on the text. For example, in [60], a number of text descriptions are randomly selected for each category of data, and images are generated using different algorithms. The evaluator sorts these images from text-related to non-related, and then the average ranking of each algorithm is counted and compared.

The manual evaluation effectively alleviates the problem that the evaluation index is out of line with the actual effect, and it is easier to grasp the overall effect of the generated data rather than the local performance. However, its disadvantages are also apparent, the main one of which is the consumption of more manpower. In addition, due to the influence of personal preferences, the evaluation results also have larger uncertainty.

As a summary, we list the descriptions of various evaluation methods in Table 3. Because of the advantages and disadvantages of automatic evaluation and manual evaluation, they are often used simultaneously in many scenarios, playing a complementary role. When the result of the manual evaluation is significantly worse than that of automatic evaluation, it is necessary to consider whether the automatic evaluation metrics cannot reflect all aspects of the generated data. On the contrary, it is necessary to pay attention to whether the evaluators in the manual evaluation method are deceived by some appearance of the generated data, that is, fail to pay attention to some defects reflected by the automatic evaluation metrics.

Table 3. A summary of various evaluation methods.

Type	Index	Description
Automatic evaluation	IS [23,45,56,58–60]	Mapping the evaluation task to a classifier.
	FID [23,45,57–60]	Using the inception network to extract features from an intermediate layer.
	TARR [61]	Computing the recognition rate of attributes on synthetic data by a model trained on real data.
	PSNR [41]	Computing an expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation.
	SSIM [62]	Measuring the similarity between the two images.
	Dice [35,58,61,63–65]	Measuring the gap between the semantic segmentation result and the real result.
	R-Precious [45]	Retrieving the relevant text for a given image query.
Manual evaluation	Sample Analysis [23,61,64]	Analyzing some typically generated images qualitatively.
	Evaluator Tests [23,35,64,65]	Inviting some reviewers to identify the real image and the generated image.
	Simulative Recommendation [60]	Ranking the images from different sources under the given conditions.

4. The Applications of CGAN

The main task of CGAN is to creatively generate new data based on the given data, and the generated data should contain as much information as possible from the given data. From this perspective, the essential role played by CGAN is to convert information from one form of representation to another. This is the reason why CGAN is mainly used for domain transfer. It should also be noted that CGAN not only plays its role in the field of domain transfer but also has the ability to conduct data augmentation in supervised learning [42,66,67], as we discussed in earlier sections. Furthermore, CGAN can even be found to succeed in some discrimination tasks [68] and emerging fields, as reported in [22,23]. Actually, there were also traditional models in the early stage, but here we only focus on the methods to use the CGAN model. The following is a timely summary of our review of the most notable works in different scenarios.

Firstly, we give an overview of the application areas of CGAN. For any task, it is necessary to include input data and output data. The input data can be images, labels, text, and so on, yet the images are the most common. In particular, if the input datum is also an image, the task can be considered an image transformation. From the perspective of humans' cognition process, the elements of an image can be roughly divided into the overall contents, partial contents, styles, colors, resolution, etc. The semantic levels of an image can also be divided into the visual layer (low level), object layer (middle level) and concept layer (high level). The low-level semantics are the style, color, and resolution of an image; the middle-level semantics include attributes and features, which are the state of an object at a certain moment in the image; the concept layer is the high-level, which is the content shown by images to humans. The task of image conversion is to change the operation of one or a few of these elements. Table 4 lists the categories of common tasks.

Table 4. The classification of tasks involving CGAN.

Task Type	Task Description	Semantic Level	Typical Papers
Super resolution	Converting image from low resolution to high resolution	Low-level	[14]
Texture transformation	Converting photos to oil paintings		[15]
Color transformation	Converting grayscale image to color image		[69]
	Coloring in a line drawing		[33]
Environment transformation	Time conversion of landscape photo	Middle-level	[70]
	Season conversion of landscape photo		[15]
Attribute edition	Object style editing		[38]
	Object outline editing		[33]
	Facial attribute editing		[71]
Image synthesis	Converting the semantic mask to a photo		[72]
	Converting sketch to photo		[73]
	Converting normal map to satellite map		[35]
	Converting thermal to photo		[74]
Image inpainting	Repairing the image with some missing pixels	High-level	[75]
Data augmentation	Generating more data from existing data		[66]
Image creation	Generating image from text description		[58]

4.1. Super Resolution (SR)

High-resolution images are widely used in the production industry and daily life. However, due to the large size of their corresponding files, it presents great challenges for stable storage and efficient transmission. Therefore, a feasible method is to use the low-resolution images in the process of image storage and transmission, and when the image needs to be employed, the high-resolution image will be restored according to the low-resolution image. Therefore, Single Image Super Resolution (SISR) has become a research hot spot in the field of computer vision [28].

SRGAN was the first work to solve the SR problem via a CGAN model [30]. In SRGAN, the generative network consists of several jump connections to synthesize the low-level and high-level semantics, and two sub-pixel convolutional layers are used to improve the image resolution. In terms of the loss function, SRGAN uses perceptual loss in addition to adversarial loss, where the perceptual loss is calculated by the feature map of the VGG network (Oxfords Visual Geometry Group's pre-training network) [29].

In recent years, Curriculum Learning (CL) [76] has received more and more attention, the main idea of which is to feed the training data to the model step-by-step rather than all at once. It has achieved great success in many tasks such as natural language processing [77,78], image recognition [79] and even image generation [80]. Inspired by curriculum learning, the ProGanSR has been proposed, and it indicates that the learning direction should be gradually transformed from a small level to a large level. ProGanSR uses an asymmetric pyramid structure to achieve efficiency. Each pyramid consists of a dense compression unit and a sub-pixel convolutional layer to double up the input resolution. Mahapatra et al. proposed a local significance diagram defining the importance of each pixel, which could be used in the adversarial loss on classical MSE [81].

Alongside the mentioned papers, numerous papers used CGAN to achieve SR [17,82–84]. Some other works on image style transfer mainly focused on the architecture of the CGAN, such as [26,31–37,72,85]. Recently, a novel model named MSG U-Net architecture is proposed based on [85], where the network is trained by allowing the flow of gradients from multiple-discriminators to a single generator at multiple scales [86].

4.2. Image Style Transfer

The task of image style transfer is to transform one type of image into another style of image, including black and white to color, line coloring, texture transformation (for example, oil painting generated from photos), and so on. Similar to SR, it falls into the category of low-level semantic image generation but locally requiring high-level semantic support.

Isola et al. [35] proposed Pix2Pix as a universal model for solving different image-to-image tasks in the supervised setting, i.e., the training data would include the source image and its corresponding destination image. Wang et al. [25] extended the framework to generate high-resolution images.

As a basis, CycleGAN [15] has made significant progress in unsupervised settings by introducing cyclic consistency loss to ensure that the destination image correctly retains domain invariant features, such as pose, of the source image. Although CycleGAN learns deterministic one-to-one mapping, it fails to capture the multi-modal attributes of the image distribution. MUNIT [38] recognizes this limitation and extends the framework to learn multi-modal mapping, which can generate different and real translation output. However, it trains different encoder–decoder models for each domain, so it is not easy to scale up to multiple domains. StarGAN [27,39] solves this problem and proposes a unified model for multi-domain translation. FUNIT [62] tries to generalize images from unseen domains using some reference images from the target domain, but it requires fine-grained class labels during training and cannot model unspecified changes within the class.

Due to the in-depth research on image style transfer, CGAN has been applied to other fields. For example, in the case of Geographic Information Systems (GIS), CGAN is used to transfer satellite images to maps [18,87]. In the case of Computer Graphics, reference [88] uses CGAN to generate the SyntheticFur dataset by transferring the style of photos and [89] uses it to denoise blurred images.

4.3. Feature Synthesis

Image feature synthesis is similar to style transfer in the sense that it edits the image, but it changes the semantic information of the image. For ordinary GAN, it is difficult to obtain hidden variables. For this reason, ICGAN is proposed in [34], where the encoder is first used to convert the image into high-level semantics, and CGAN takes the high-level semantics as the condition to generate the edited image. In recent years, researchers have mainly been concerned about the difficulty in collecting paired data in feature synthesis. Accordingly, among many works, reference [38] assumes that the latent space of the images can be decomposed into content space and style space, where content space is shared by the source domain and target domain, and style space is owned by two domains, respectively. Meanwhile, reference [90] further applies the idea of cyclic consistency in feature synthesis tasks and puts forward a cross-cycle consistency loss.

As a specific type of data with high recognition and wide application, face photos have been studied by a large number of researchers. The reason for its high recognition lies in numerous features [71]. Without a doubt, it is also an important subject to transform face features with CGAN. The authors of [91] explicitly represent face features with vectors, which satisfy orthogonality (each component controls different features independently, without affecting each other), richness (that vectors can express enough features), and controllability (where users can achieve satisfactory results by specifying this vector). The authors of [92] further refined the generated image by using unlabeled data. However, the foundation theory and common improvement methods of CGAN can only be used for typical image modification, such as adding happy, sad, angry and other basic expressions to

the face, but it cannot customize arbitrary complex expressions. To this end, reference [40] introduces the concept of "expression intensity", which can combine complex expressions of sadness with small amounts of anger and similar expressions. In addition to the design of the overall network architecture, the design of network components has also been carried out. For example, reference [61] introduces a Graph Convolutional Network (GCN) network, with the graph structure representing the features, the nodes of the graph representing components of the feature, and the edges of the graph representing the relationship between the components. Recently, the attention mechanism has been further employed in this field [19,41,61].

4.4. Image Inpainting

In the process of image storage, some pixels may be lost due to various reasons. In order to restore the original appearance of the image as far as possible, certain methods are needed for image inpainting. Intuitively, pixels in an image have a great correlation with their neighboring pixels but less correlation with distant pixels. Therefore, traditional methods similar to interpolation have been widely used [93–95]. However, if there are continuously missing pixels in the image, the significant effect of the interpolation method will be greatly limited, especially when the missing part contains the whole object, which will render the interpolation method to be completely unpredictable. Under this circumstance, it is a great alternative to employ CGAN [75,96–99].

However, using CGAN to solve image inpainting also faces many new problems. For example, reference [96] can only fill in the center of the image, not the area of variable sizes. When the missing pixels form a connected region, the boundary of the missing region can still be seen on the inpainted image [97]. In addition, problems such as model collapse and overfitting are also quite serious. In order to solve these problems, reference [100] optimized the overall framework and super parameters of CGAN. A multi-stage generation method is introduced to create high-quality images by gradually adding layers to the generative network and discriminative network in [55]. Following this work, reference [101] uses the adaptive normalization layer [102] to control the visual features of images at different scales on the basis of [55]. Some studies also attempt to improve from the perspective of loss functions, such as Wasserstein distance [20], least squares [20] and energy-based GANs [103]. In addition, similar to the SR, here we can also adapt the curriculum learning method [104].

4.5. Pose Transfer

Pose transfer is another kind of image domain transfer task. For an object, given a photo taken from an angle and a target angle, the model needs to generate an image of the object observed from the target angle. Intuitively, when an object is viewed from a particular angle, a considerable area of the object's surface is occluded, and if the target angle involves these areas, the model needs to be drawn from scratch. That is to say, the trained model should have a considerable amount of prior knowledge about the characteristics of the object, meaning that the model for pose transfer is often oriented to specific objects. Reference [105] discusses the pose transfer of a figure image. To solve the deviation between pixels, a deformable jump connection is introduced into the generator. In addition, in order to ensure that the details of the generated image match with the original image, a nearest-neighborhood loss function is proposed. Reference [106] further considers the pose transfer with a background, which uses a multi-branch reconstruction network to fuse the foreground, background and pose information into embedded features and then combine these features to re-compose the input image itself. The training process involves a progressive enhancement mechanism.

Another target of pose transfer is a vehicle. Because vehicles are more regular shapes than figures, reference [107] integrates the prior geometry of 3D space into the model, forming a semi-parametric approach. Specifically, the symmetries of regular geometry and piecewise planarity are integrated into the composition of the new pose, and the Image

Completion Network (ICN) is used to generate the final image. Reference [108] further uses the law of perspective transformation, decomposes the vehicle into several quadrilateral planes, and processes the images on each plane respectively, obtaining a more realistic image in the overall structure.

In general, the application of CGAN in the pose transfer is still limited and heavily depends on prior knowledge in specific fields. There is still a long way to go to realize automatic and adaptive pose transfer. In the future, classifier ensemble techniques [21,109–112] may be a feasible approach to tackle this challenge.

4.6. Image Generation from Text Descriptions

Using CGANs to generate images according to text description was first proposed by Reed et al. [113]. It first encodes the text description as a vector representation and then uses the representation as a condition to train CGAN. In order to make the generated image more in line with the user's needs, Reed et al. added the position constraint of the object in the image to the CGAN condition [114]. In detail, besides the text description, the expected coordinates of some objects will also be fed into the CGAN. Afterward, Zhang et al. proposed StackGAN [16] and StackGAN++ [43], which have been detailed in this paper earlier. In addition, HDGAN [115] adopts a multi-stage strategy similar to StackGAN++.

The previous research in this area only used global sentence embedding as a condition to train CGAN, and the granularity was quite rough. AttnGAN [116] introduced the attention [117] module and added word-level conditions, making the generated image closely related to the description. MirrorGAN [44] borrows the idea of cycle consistency to re-convert the generated image into text and supervise the distance between the re-converted text and the original text during the training process. DMGAN [45] uses a dynamic memory module to screen out which text contents are more important to the generation of images and is dedicated to improving the resolution of relevant parts of images. SD-GAN [60] uses two networks that work together to make the generated images consistent in various descriptions.

Similar to the task of image style transfer, some researchers also proposed some effective and efficient network architectures [58–60,115,118–120].

4.7. Other Fields

Dehazegan [121] first applied conditional GAN into the field of single image dehazing and explored the connection among CGAN, image dehazing, and differentiable programming, which advanced the theories and application of these fields. DATNet [122] proposed a new architecture for addressing sequence labeling, which employed two variants: datnet-f and datnet-p to explore the effective feature fusion between high resources and low resources and further proposed a novel generalized resource adversarial discriminator (GRAD) to boost model generalization. In cross-modal retrieval, TANSS [123] proposed ternary adversarial networks with self-supervision to solve the problem of inconsistency between the seen classes in the source set and unseen classes in the target set. In semantic segmentation, reference [124] proposed a weak supervision adversarial domain adaptation to improve the segmentation performance from synthetic data to real scenes, which consists of three deep neural networks. We summarized all CGANs in different application fields in Table 5.

Table 5. A summary of CGANs in various application fields.

Application Fields	Task Description	Typical Papers
Super Resolution (SR)	Converting image from low resolution to high resolution	[17,26,28–37,72,76–86]
Image style transfer	Transforming one type of image into another style of image, such as color transformation, texture transformation, etc.	[15,18,25,27,35,38,39,62,87–89]
Feature synthesis	Combining separate features to make composite features	[19,34,38,40,41,61,71,90–92]
Image inpainting	Restoring the original appearance of the image	[20,55,75,93–104]
Pose transfer	Generating images of the object observed from other angles	[21,105–112]
Image generation from text descriptions	Generating images according to text description	[16,43–45,58–60,113–120]
Other fields	Such as: single image dehazing, sequence labeling, inconsistency, etc.	[121–124]

5. The Remaining Problems and Challenges

Although CGAN has made great progress in recent years, compared with many other sub-fields of artificial intelligence, it still faces many problems and challenges. Here are a few insights gained from our literature review.

- **Conditions are ignored.** Many CGAN models easily ignore the conditions and generate data that fail to meet the conditions in some practical applications. Currently, there are maybe two causes for this problem. First, the characteristics of samples under different conditions in the training set are not significantly different; that is, the model cannot accurately classify samples according to conditions. Second, the network architecture is not reasonable. For example, the conditions representing high-level semantics of data only act on the first half of the network, resulting in only affecting low-level semantics of data.
- **Proper coding of conditions.** In order to make the CGAN model easily trainable, conditions need to be coded. However, how to have the reasonable encoding of conditions is a problem that needs further study in the field of CGAN because the commonly used Convolution Neural Network (CNN) may not contain the details that essentially correspond to the characteristics of the generated image.
- **The high calculation cost.** In CGAN, in addition to learning the distribution of generated images as unconditional GAN, the relationship between input conditions and generated images should also be learned, which further aggravates the burden of the model. Therefore, from this perspective, it is necessary to reduce the complexity of the model. One possible approach to this problem is model compression. However, the current research on how to implement model compression for CGAN is still very preliminary. For example, reference [104] only reduces the capacity of the model by screening the convolution kernel, which plays a rather limited role actually.
- **The strong coupling with data.** The training effectiveness of CGAN often depends heavily on the given data. For example, in the image style conversion task, let us consider a scenario where the image time needs to be converted from noon to evening. If the training data are only natural scenery photos, it is still difficult for CGAN to convert city photos after the training, which restricts the application of CGAN. This problem is difficult to solve by moderating the model alone. Therefore, a variety of standard data sets is the viable solution to this problem. At present, in terms of

scientific research, some websites such as Kaggle have provided many types of data sets, but they are far from satisfying the research needs.

- **The strong coupling with the task.** Compared with the unconditional GAN, CGAN can be used to complete more tasks. However, the network used to solve one task is difficult to apply directly to another task. Despite that quite a few researchers summarized some design principles, such as the tricks summarized in Section 2.2, most of them only guided the overall framework of the network rather than a specific architecture inside the generative network and the discriminative network, which might have a great difference in different tasks.
- **Neural Architecture Search (NAS)** It should be noted that in recent years, Neural Architecture Search (NAS) technology has made a lot of achievements in the field of image classification [125,126], which is expected to help with the automatic design of network architecture. However, the complexity of CGAN means that there is still a long way to go, and even NAS technology can be employed.

Furthermore, it should be reminded that, currently, CGAN is only a branch of GAN, and the performance improvement methods of CGAN [127–130] are not significantly different from the traditional GAN. From the perspective of the application, CGAN's application scope is still very limited, except for computer vision. For example, in the field of natural language processing, published research results are comparatively much less than in the computer vision [131,132]. The reason is the complexity of language itself and the discreteness of text data. In the future, overcoming these difficulties and designing a more efficient model will be the main tasks for natural language processing based on CGAN.

6. Conclusions and Prospects

With the rapid development of artificial intelligence technologies, GAN has been widely applied in the field of data generation. In order to control the output results of GAN, researchers added conditions to the input of GAN to solve the ill-conditioned constraint problem, and CGAN has been investigated extensively in recent years.

This paper focuses on the research of CGAN in domain transfer and introduces the development process of CGAN from the perspectives of: the principle of CGAN, the development of loss function, the model variants of CGAN, evaluation methods and application fields. These studies fully demonstrate that the CGAN model has great potential and research value.

However, due to the short life of CGAN to date, it is still in the initial stage of development, and the relevant theories and applications are far from maturity. Meanwhile, we have also discussed and listed the future development directions of CGAN in Section 5. In summary, more investigations on CGAN are necessary for consolidating its development, and we hope this paper will contribute to researchers interested in applying CGAN in different domains.

Author Contributions: Conceptualization, J.S. (Jun Shen); Formal analysis, G.Z.; Validation, Y.L. Writing—original draft, Y.F.; Writing—review & editing, J.S. (Jiachen Shi). All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the National Natural Science Foundation of China (21790340 and 21674113), the National Program on Key Basic Research Project (2020YFA0713600), and Shen's contribution to this paper is supported partially by the ARC Discovery Project DP180101051.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Laloy, E.; Linde, N.; Jacques, D. Approaching geoscientific inverse problems with vector-to-image domain transfer networks. *Adv. Water Resour.* **2021**, *152*, 103917. [CrossRef]
2. Li, L.; Li, J.; Lv, C.; Yuan, Y.; Zhao, B. Maize residue segmentation using Siamese domain transfer network. *Comput. Electron. Agric.* **2021**, *187*, 106261. [CrossRef]
3. Liu, H.; Guo, F.; Xia, D. Domain adaptation with structural knowledge transfer learning for person re-identification. *Multimed. Tools Appl.* **2021**, 80, 29321–29337. [CrossRef]
4. Liu, J.; Li, Q.; Zhang, P.; Zhang, G.; Liu, M. Unpaired domain transfer for data augment in face recognition. *IEEE Access* **2020**, *8*, 39349–39360. [CrossRef]
5. Al-Shannaq, A.; Elrefaei, L. Age estimation using specific domain transfer learning. *Jordanian J. Comput. Inf. Technol. (JJCIT)* **2020**, *6*, 122–139.
6. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
7. Suh, Y.; Han, B.; Kim, W.; Lee, K.M. Stochastic class-based hard example mining for deep metric learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7251–7259.
8. Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 14454–14463.
9. Mao, J.; Wang, H.; Spencer, B.F., Jr. Toward data anomaly detection for automated structural health monitoring: Exploiting generative adversarial nets and autoencoders. *Struct. Health Monit.* **2021**, *20*, 1609–1626. [CrossRef]
10. Xia, Y.; Zhang, L.; Ravikumar, N.; Attar, R.; Piechnik, S.K.; Neubauer, S.; Petersen, S.E.; Frangi, A.F. Recovering from missing data in population imaging–Cardiac MR image imputation via conditional generative adversarial nets. *Med. Image Anal.* **2021**, *67*, 101812. [CrossRef]
11. Wen, P.; Zhang, S.; Du, S.; Qu, B.; Song, X. A Full Mean-Square Analysis of CNSAF Algorithm For Noncircular Inputs. *J. Frankl. Inst.* **2021**, *358*, 7883–7899. [CrossRef]
12. Wang, Z.; She, Q.; Ward, T.E. Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–38. [CrossRef]
13. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
14. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Change Loy, C. Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 63–79.
15. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
16. Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D.N. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In Proceedings of the IEEE international Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5907–5915.
17. Zhao, L.; Liang, J.; Bai, H.; Wang, A.; Zhao, Y. Simultaneously Color-Depth Super-Resolution with Conditional Generative Adversarial Network. *arXiv* **2017**, arXiv:1708.09105.
18. Vaishali, I.; Rishabh, S.; Pragati, P. Image to Image Translation: Generating maps from satellite images. *arXiv* **2021**, arXiv:2105.09253.
19. He, Z.; Zuo, W.; Kan, M.; Shan, S.; Chen, X. AttGAN: Facial Attribute Editing by Only Changing What You Want. *IEEE Trans. Image Process.* **2019**, *28*, 5464–5478. [CrossRef] [PubMed]
20. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein gan. *arXiv* **2017**, arXiv:1701.07875.
21. Alzubi, O.A.; Alzubi, J.A.; Alweshah, M.; Qiqieh, I.; Al-Shami, S.; Ramachandran, M. An optimal pruning algorithm of classifier ensembles: Dynamic programming approach. *Neural Comput. Appl.* **2020**, *32*, 16091–16107. [CrossRef]
22. Perraudin, N.; Marcon, S.; Lucchi, A.; Kacprzak, T. Emulation of cosmological mass maps with conditional generative adversarial networks. *arXiv* **2020**, arXiv:2004.08139.
23. Kamran, S.A.; Hossain, K.F.; Tavakkoli, A.; Zuckerbrod, S.L. Fundus2Angio: A Novel Conditional GAN Architecture for Generating Fluorescein Angiography Images from Retinal Fundus Photography. *arXiv* **2020**, arXiv:2005.05267.
24. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
25. 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, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8798–8807.
26. Kim, T.; Cha, M.; Kim, H.; Lee, J.K.; Kim, J. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1857–1865.
27. Choi, Y.; Choi, M.; Kim, M.; Ha, J.W.; Kim, S.; Choo, J. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8789–8797.

28. Denton, E.; Chintala, S.; Szlam, A.; Fergus, R. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. *arXiv* **2015**, arXiv:1506.05751 .
29. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.P.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *arXiv* **2016**, arXiv:1609.04802.
30. Sonderby, C.K.; Caballero, J.; Theis, L.; Shi, W.; Huszar, F. Amortised MAP Inference for Image Super-resolution. In Proceedings of the International Conference on Learning Representation, Toulon, France, 24–26 April 2017.
31. Yoo, D.; Kim, N.; Park, S.; Paek, A.S.; Kweon, I.S. Pixel-level domain transfer. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 517–532.
32. Li, C.; Wand, M. Precomputed real-time texture synthesis with markovian generative adversarial networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 702–716.
33. Zhu, J.Y.; Krähenbühl, P.; Shechtman, E.; Efros, A.A. Generative visual manipulation on the natural image manifold. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 597–613.
34. Perarnau, G.; De Weijer, J.V.; Raducanu, B.; Alvarez, J.M. Invertible Conditional GANs for image editing. *arXiv* **2016**, arXiv:1611.06355.
35. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
36. Sangkloy, P.; Lu, J.; Fang, C.; Yu, F.; Hays, J. Scribbler: Controlling Deep Image Synthesis with Sketch and Color. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6836–6845.
37. Yi, Z.; Zhang, H.; Tan, P.; Gong, M. Dualgan: Unsupervised dual learning for image-to-image translation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2849–2857.
38. Huang, X.; Liu, M.Y.; Belongie, S.; Kautz, J. Multimodal Unsupervised Image-to-Image Translation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 172–189.
39. Choi, Y.; Uh, Y.; Yoo, J.; Ha, J.W. Stargan v2: Diverse image synthesis for multiple domains. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8188–8197.
40. Ding, H.; Sricharan, K.; Chellappa, R. ExprGAN: Facial Expression Editing with Controllable Expression Intensity. In Proceedings of the Association for the Advance of Artificial Intelligence (AAAI), New Orleans, LA, USA, 2–3 February 2018; pp. 6781–6788.
41. Liu, M.; Ding, Y.; Xia, M.; Liu, X.; Ding, E.; Zuo, W.; Wen, S. Stgan: A unified selective transfer network for arbitrary image attribute editing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3673–3682.
42. Dey, S.; Das, S.; Ghosh, S.; Mitra, S.; Chakrabarty, S.; Das, N. SynCGAN: Using learnable class specific priors to generate synthetic data for improving classifier performance on cytological images. *arXiv* **2020**, arXiv:2003.05712.
43. Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D.N. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1947–1962. [CrossRef]
44. Qiao, T.; Zhang, J.; Xu, D.; Tao, D. Mirrorgan: Learning text-to-image generation by redescription. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1505–1514.
45. Zhu, M.; Pan, P.; Chen, W.; Yang, Y. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5802–5810.
46. Taigman, Y.; Polyak, A.; Wolf, L. Unsupervised cross-domain image generation. *arXiv* **2016**, arXiv:1611.02200.
47. Royer, A.; Bousmalis, K.; Gouws, S.; Bertsch, F.; Mosseri, I.; Cole, F.; Murphy, K. Xgan: Unsupervised image-to-image translation for many-to-many mappings. In *Domain Adaptation for Visual Understanding*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 33–49.
48. Ma, J.; Xu, H.; Jiang, J.; Mei, X.; Zhang, X.P. DDcGAN: A dual-discriminator conditional generative adversarial network for multi-resolution image fusion. *IEEE Trans. Image Process.* **2020**, *29*, 4980–4995. [CrossRef] [PubMed]
49. Zhang, Y.; Liu, S.; Dong, C.; Zhang, X.; Yuan, Y. Multiple cycle-in-cycle generative adversarial networks for unsupervised image super-resolution. *IEEE Trans. Image Process.* **2019**, *29*, 1101–1112. [CrossRef]
50. Ma, Y.; Zhong, G.; Liu, W.; Wang, Y.; Jiang, P.; Zhang, R. ML-CGAN: Conditional Generative Adversarial Network with a Meta-learner Structure for High-Quality Image Generation with Few Training Data. *Cogn. Comput.* **2021**, *13*, 418–430. [CrossRef]
51. Liu, R.; Ge, Y.; Choi, C.L.; Wang, X.; Li, H. Divco: Diverse conditional image synthesis via contrastive generative adversarial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 16377–16386.
52. Han, L.; Min, M.R.; Stathopoulos, A.; Tian, Y.; Gao, R.; Kadav, A.; Metaxas, D.N. Dual Projection Generative Adversarial Networks for Conditional Image Generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QB, Canada, 11 October 2021; pp. 14438–14447.

53. Ueda, Y.; Fujii, K.; Saito, Y.; Takamichi, S.; Baba, Y.; Saruwatari, H. HumanACGAN: Conditional generative adversarial network with human-based auxiliary classifier and its evaluation in phoneme perception. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Virtual, 6–12 June 2021; pp. 6468–6472.
54. Wang, Z. Learning Fast Converging, Effective Conditional Generative Adversarial Networks with a Mirrored Auxiliary Classifier. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Montreal, QB, Canada, 11 October 2021; pp. 2566–2575.
55. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
56. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 2234–2242.
57. 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, Long Beach, CA, USA, 4–9 December 2017; pp. 6626–6637.
58. Stap, D.; Bleeker, M.; Ibrahimi, S.; ter Hoeve, M. Conditional Image Generation and Manipulation for User-Specified Content. *arXiv* **2020**, arXiv:2005.04909.
59. Souza, D.M.; Wehrmann, J.; Ruiz, D.D. Efficient Neural Architecture for Text-to-Image Synthesis. *arXiv* **2020**, arXiv:2004.11437.
60. Yin, G.; Liu, B.; Sheng, L.; Yu, N.; Wang, X.; Shao, J. Semantics disentangling for text-to-image generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2327–2336.
61. Bhattarai, B.; Kim, T. Inducing Optimal Attribute Representations for Conditional GANs. *arXiv* **2020**, arXiv:2003.06472.
62. Liu, M.; Huang, X.; Mallya, A.; Karras, T.; Aila, T.; Lehtinen, J.; Kautz, J. Few-Shot Unsupervised Image-to-Image Translation. *arXiv* **2019**, arXiv:1905.01723.
63. Chen, J.; Li, Y.; Ma, K.; Zheng, Y. Generative Adversarial Networks for Video-to-Video Domain Adaptation. In Proceedings of the AAAI, New York, NY, USA, 7–12 February 2020; pp. 3462–3469.
64. Zhu, J.Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A.A.; Wang, O.; Shechtman, E. Toward multimodal image-to-image translation. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 465–476.
65. Zakharov, E.; Shysheya, A.; Burkov, E.; Lempitsky, V. Few-shot adversarial learning of realistic neural talking head models. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9459–9468.
66. Antoniou, A.; Storkey, A.; Edwards, H. Data augmentation generative adversarial networks. *arXiv* **2017**, arXiv:1711.04340.
67. Abdollahi, A.; Pradhan, B.; Sharma, G.; Maulud, K.N.A.; Alamri, A. Improving Road Semantic Segmentation Using Generative Adversarial Network. *IEEE Access* **2021**, *9*, 64381–64392. [CrossRef]
68. Ji, Y.; Zhang, H.; Wu, Q.J. Saliency detection via conditional adversarial image-to-image network. *Neurocomputing* **2018**, *316*, 357–368. [CrossRef]
69. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
70. Laffont, P.Y.; Ren, Z.; Tao, X.; Qian, C.; Hays, J. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Trans. Graph. (TOG)* **2014**, *33*, 1–11. [CrossRef]
71. Zhang, Z.; Song, Y.; Qi, H. Age progression/regression by conditional adversarial autoencoder. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5810–5818.
72. 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, Long Beach, CA, USA, 15–20 June 2019; pp. 2337–2346.
73. Eitz, M.; Hays, J.; Alexa, M. How do humans sketch objects? *ACM Trans. Graph. (TOG)* **2012**, *31*, 1–10. [CrossRef]
74. Hwang, S.; Park, J.; Kim, N.; Choi, Y.; So Kweon, I. Multispectral pedestrian detection: Benchmark dataset and baseline. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1037–1045.
75. Yeh, R.A.; Chen, C.; Lim, T.Y.; Hasegawajohnson, M.; Do, M.N. Semantic Image Inpainting with Perceptual and Contextual Losses. *arXiv* **2016**, arXiv:1607.07539.
76. Bengio, Y.; Louradour, J.; Collobert, R.; Weston, J. Curriculum learning. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 41–48.
77. Kocmi, T.; Bojar, O. Curriculum Learning and Minibatch Bucketing in Neural Machine Translation. In Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP 2017), Varna, Bulgaria, 2–8 September 2017; pp. 379–386.
78. Platanios, E.A.; Stretcu, O.; Neubig, G.; Poczos, B.; Mitchell, T. Competence-based Curriculum Learning for Neural Machine Translation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 1162–1172.
79. Sarafianos, N.; Giannakopoulos, T.; Nikou, C.; Kakadiaris, I.A. Curriculum learning for multi-task classification of visual attributes. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 2608–2615.

80. Zhang, H.; Hu, Z.; Luo, C.; Zuo, W.; Wang, M. Semantic image inpainting with progressive generative networks. In Proceedings of the 26th ACM International Conference on Multimedia, Seoul, Korea, 22–26 October 2018; pp. 1939–1947.
81. Mahapatra, D.; Bozorgtabar, B. Retinal Vasculature Segmentation Using Local Saliency Maps and Generative Adversarial Networks for Image Super Resolution. *arXiv* **2017**, arXiv:1710.04783.
82. Sanchez, I.; Vilaplana, V. Brain MRI super-resolution using 3D generative adversarial networks. *arXiv* **2018**, arXiv:1812.11440.
83. Rangnekar, A.; Mokashi, N.; Ientilucci, E.J.; Kanan, C.; Hoffman, M.J. Aerial Spectral Super-Resolution using Conditional Adversarial Networks. *arXiv* **2017**, arXiv:1712.08690.
84. Chen, Y.; Shi, F.; Christodoulou, A.G.; Xie, Y.; Zhou, Z.; Li, D. Efficient and accurate MRI super-resolution using a generative adversarial network and 3D multi-level densely connected network. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Granada, Spain, 16–20 September 2018; pp. 91–99.
85. Liu, M.Y.; Breuel, T.; Kautz, J. Unsupervised image-to-image translation networks. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 700–708.
86. Kumarapu, L.; Shiv, R.D.; Baddam, K.; Satya, R.V.K. Efficient High-Resolution Image-to-Image Translation using Multi-Scale Gradient U-Net. *arXiv* **2021**, arXiv:2105.13067.
87. Wang, Y.; Bittner, K.; Zorzi, S. Machine-learned 3D Building Vectorization from Satellite Imagery. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2021), Nashville, TN, USA, 19–25 June 2021.
88. Le, T.; Poplin, R.; Bertsch, F.; Toor, A.S.; Oh, M.L. SyntheticFur dataset for neural rendering. *arXiv* **2021**, arXiv:2105.06409.
89. Kim, H.J.; Lee, D. Image denoising with conditional generative adversarial networks (CGAN) in low dose chest images. *Nucl. Instrum. Methods Phys. Res. Sect. A* **2020**, *954*, 161914. [CrossRef]
90. Lee, H.Y.; Tseng, H.Y.; Huang, J.B.; Singh, M.; Yang, M.H. Diverse image-to-image translation via disentangled representations. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 35–51.
91. Kaneko, T.; Hiramatsu, K.; Kashino, K. Generative attribute controller with conditional filtered generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6089–6098.
92. Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; Webb, R. Learning from simulated and unsupervised images through adversarial training. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2107–2116.
93. Barnes, C.; Shechtman, E.; Finkelstein, A.; Goldman, D.B. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* **2009**, *28*, 24. [CrossRef]
94. Barnes, C.; Shechtman, E.; Goldman, D.B.; Finkelstein, A. The generalized patchmatch correspondence algorithm. In Proceedings of the European Conference on Computer Vision, Crete, Greece, 5–11 September 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 29–43.
95. Darabi, S.; Shechtman, E.; Barnes, C.; Goldman, D.B.; Sen, P. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Trans. Graph. (TOG)* **2012**, *31*, 1–10. [CrossRef]
96. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2536–2544.
97. Iizuka, S.; Simoserra, E.; Ishikawa, H. Globally and locally consistent image completion. *ACM Trans. Graph.* **2017**, *36*, 107. [CrossRef]
98. Liu, G.; Reda, F.A.; Shih, K.J.; Wang, T.C.; Tao, A.; Catanzaro, B. Image inpainting for irregular holes using partial convolutions. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 85–100.
99. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. Free-form image inpainting with gated convolution. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 4471–4480.
100. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
101. 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, Long Beach, CA, USA, 15–20 June 2019; pp. 4401–4410.
102. Gatys, L.A.; Ecker, A.S.; Bethge, M. A Neural Algorithm of Artistic Style. *J. Vis.* **2016**, *16*, 326. [CrossRef]
103. Zhao, J.; Mathieu, M.; Lecun, Y. Energy-based Generative Adversarial Network. In Proceedings of the International Conference of Learning Representation (ICLR), Toulon, France, 24–26 April 2017.
104. Hedjazi, M.A.; Genç, Y. Learning to Inpaint by Progressively Growing the Mask Regions. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea, 27–28 October 2019; pp. 4591–4596.
105. Siarohin, A.; Sangineto, E.; Lathuiliere, S.; Sebe, N. Deformable gans for pose-based human image generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3408–3416.
106. Ma, L.; Sun, Q.; Georgoulis, S.; Van Gool, L.; Schiele, B.; Fritz, M. Disentangled person image generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 99–108.
107. Palazzi, A.; Bergamini, L.; Calderara, S.; Cucchiara, R. Warp and Learn: Novel Views Generation for Vehicles and Other Objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *14*, 2216–2227. [CrossRef]

108. Lv, K.; Sheng, H.; Xiong, Z.; Li, W.; Zheng, L. Pose-based view synthesis for vehicles: A perspective aware method. *IEEE Trans. Image Process.* **2020**, *29*, 5163–5174. [CrossRef]
109. Sethuraman, J.; Alzubi, J.A.; Manikandan, R.; Gheisari, M.; Kumar, A. Eccentric methodology with optimization to unearth hidden facts of search engine result pages. *Recent Patents Comput. Sci.* **2019**, *12*, 110–119. [CrossRef]
110. Alzubi, O.A.; Alzubi, J.A.; Tedmori, S.; Rashaideh, H.; Almomani, O. Consensus-based combining method for classifier ensembles. *Int. Arab J. Inf. Technol.* **2018**, *15*, 76–86.
111. Al-Najdawi, N.; Tedmori, S.; Alzubi, O.A.; Dorgham, O.; Alzubi, J.A. A frequency based hierarchical fast search block matching algorithm for fast video communication. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 447–455. [CrossRef]
112. Alzubi, J.A.; Jain, R.; Kathuria, A.; Khandelwal, A.; Saxena, A.; Singh, A. Paraphrase identification using collaborative adversarial networks. *J. Intell. Fuzzy Syst.* **2020**, *39*, 1021–1032. [CrossRef]
113. Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; Lee, H. Generative Adversarial Text to Image Synthesis. In Proceedings of the International Conference on Machine Learning, New York City, NY, USA, 19–24 June 2016; pp. 1060–1069.
114. Reed, S.E.; Akata, Z.; Mohan, S.; Tenka, S.; Schiele, B.; Lee, H. Learning what and where to draw. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 217–225.
115. Zhang, Z.; Xie, Y.; Yang, L. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6199–6208.
116. Xu, T.; Zhang, P.; Huang, Q.; Zhang, H.; Gan, Z.; Huang, X.; He, X. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1316–1324.
117. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2048–2057.
118. Hong, S.; Yang, D.; Choi, J.; Lee, H. Inferring semantic layout for hierarchical text-to-image synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7986–7994.
119. Zhao, B.; Meng, L.; Yin, W.; Sigal, L. Image generation from layout. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8584–8593.
120. Agrawal, S.; Venkitachalam, S.; Raghu, D.; Pai, D. Directional GAN: A Novel Conditioning Strategy for Generative Networks. *arXiv* **2021**, arXiv:2105.05712.
121. Zhu, H.; Peng, X.; Chandrasekhar, V.; Li, L.; Lim, J.H. DehazeGAN: When Image Dehazing Meets Differential Programming. In Proceedings of the IJCAI, Stockholm, Sweden, 13–19 July 2018; pp. 1234–1240.
122. Zhou, J.T.; Zhang, H.; Jin, D.; Peng, X. Dual adversarial transfer for sequence labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 434–446. [CrossRef]
123. Xu, X.; Lu, H.; Song, J.; Yang, Y.; Shen, H.T.; Li, X. Ternary adversarial networks with self-supervision for zero-shot cross-modal retrieval. *IEEE Trans. Cybern.* **2019**, *50*, 2400–2413. [CrossRef]
124. Wang, Q.; Gao, J.; Li, X. Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes. *IEEE Trans. Image Process.* **2019**, *28*, 4376–4386. [CrossRef]
125. Elsken, T.; Metzen, J.H.; Hutter, F. Neural architecture search: A survey. *arXiv* **2018**, arXiv:1808.05377
126. Wistuba, M.; Rawat, A.; Pedapati, T. A survey on neural architecture search. *arXiv* **2019**, arXiv:1905.01392.
127. Wang, Y.; Chen, Y.C.; Zhang, X.; Sun, J.; Jia, J. Attentive Normalization for Conditional Image Generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5094–5103.
128. Odena, A.; Buckman, J.; Olsson, C.; Brown, T.B.; Olah, C.; Raffel, C.; Goodfellow, I. Is Generator Conditioning Causally Related to GAN Performance. *arXiv* **2018**, arXiv:1802.08768 .
129. Brock, A.; Donahue, J.; Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
130. Zand, J.; Roberts, S. Mixture Density Conditional Generative Adversarial Network Models (MD-CGAN). *arXiv* **2020**, arXiv:2004.03797.
131. Yu, L.; Zhang, W.; Wang, J.; Yu, Y. Seqgan: Sequence generative adversarial nets with policy gradient. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
132. Fedus, W.; Goodfellow, I.; Dai, A.M. MaskGAN: Better Text Generation via Filling in the _____. *arXiv* **2018**, arXiv:1801.07736.