electronics

MDPI

Article

aRTIC GAN: A Recursive Text-Image-Conditioned GAN

Edoardo Alati, Carlo Alberto Caracciolo, Marco Costa, Marta Sanzari, Paolo Russo * and Irene Amerini

Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25, 00185 Rome, Italy; alati@diag.uniroma1.it (E.A.); caracciolo@diag.uniroma1.it (C.A.C.); costa@diag.uniroma1.it (M.C.); sanzari@diag.uniroma1.it (M.S.); amerini@diag.uniroma1.it (I.A.)

* Correspondence: paolo.russo@diag.uniroma1.it

Abstract: Generative Adversarial Networks have recently demonstrated the capability to synthesize photo-realistic real-world images. However, they still struggle to offer high controllability of the output image, even if several constraints are provided as input. In this work, we present a Recursive Text-Image-Conditioned GAN (aRTIC GAN), a novel approach for multi-conditional image generation under concurrent spatial and text constraints. It employs few line drawings and short descriptions to provide informative yet human-friendly conditioning. The proposed scenario is based on accessible constraints with high degrees of freedom: sketches are easy to draw and add strong restrictions on the generated objects, such as their orientation or main physical characteristics. Text on its side is so common and expressive that easily enforces information otherwise impossible to provide with minimal illustrations, such as objects components color, color shades, etc. Our aRTIC GAN is suitable for the sequential generation of multiple objects due to its compact design. In fact, the algorithm exploits the previously generated image in conjunction with the sketch and the text caption, resulting in a recurrent approach. We developed three network blocks to tackle the fundamental problems of catching captions' semantic meanings and of handling the trade-off between smoothing grid-pattern artifacts and visual detail preservation. Furthermore, a compact three-task discriminator (covering global, local and textual aspects) was developed to preserve a lightweight and robust architecture. Extensive experiments proved the validity of aRTIC GAN and show that the combined use of sketch and description allows us to avoid explicit object labeling.

Keywords: Conditional GAN; Image-to-Image Translation; Text-to-Image Synthesis; multi-conditional image generation

check for updates

Citation: Alati, E.; Caracciolo, C.A.; Costa, M.; Sanzari, M.; Russo, P.; Amerini, I. aRTIC GAN: A Recursive Text-Image-Conditioned GAN. *Electronics* **2022**, *11*, 1737. https://doi.org/10.3390/electronics11111737

Academic Editors: Stefanos Kollias, Leonardo Galteri and Claudio Ferrari

Received: 27 April 2022
Accepted: 23 May 2022
Published: 30 May 2022

1. Introduction

In the last decade, deep learning (DL) algorithms have been capable of generating photo-realistic images and videos, useful in a wide range of applications, such as computer graphics, digital design and art generation. In this context, the scientific community is exploring the possibility of controlling image synthesis by feeding auxiliary inputs, such as category labels, descriptive text, hand-drawn sketches, semantic maps and many others. Generative Adversarial Networks (GANs), first introduced by Goodfellow et al. [1], represent nowadays the state-of-the-art solution. Despite their success, GANs are affected by training instabilities and are sensitive to hyperparameters configuration. The high-dimensional image space of such networks exacerbates the complexity of generating high-resolution images in opposition to low-resolution and simulated data, such as MNIST [2], Fashion MNIST [3] and CoDraw [4]. Recent models introduced by Zhang et al. [5,6] have proven to achieve excellent results exploiting a multi-stage approach, in which several networks cooperate in sequence at different resolutions. Nonetheless, the complexity in the training phase and the limitations in generating multiple objects inside the same image are still key issues.

In order for the image generating system to be effective, bridging the gap between high-level concepts, such as sketches and text descriptions, and pixel-level details is required.

Depending on the nature of such constraints, all conditional GAN models can be gathered into three main groups: *Image-to-Image Translation*, *Text-to-Image Synthesis* and *Style Transfer*. *Image-to-Image Translation* (I2IT) offers hard spatial constraints exploitation at the expense of color-pattern-level information and an increasing sketch complexity for fine detail generation, resulting in the suppression of chromatic and style variety. Isola et al. [7] and Wang et al. [8] represent exhaustive examples of the described behaviors. On the other hand, *Text-to-Image Synthesis* (T2IS) employs written captions, which are flexible and informative but hard to handle when coping with detailed shape descriptions. This may lead to high color variations but little or no shape adjustments. In general, this problem is solved using complex cascade networks to capture every possible semantic meaning, e.g., [9]. However, this scenario results in elaborated models, restraining and limiting the possibilities of generating multiple objects in the same image. *Style Transfer* (ST) works in a different context, allowing the network to homogeneously apply a requested style on a given input image. This kind of conditioning lacks fine control over the final generated image, as the model performs a translation rather than a proper generation. An important accomplishment is pictured by Zhu et al. [10].

Two of the aforementioned categories are prone to be affected by *Mode Collapsing*. This phenomenon is characterized by the generation of samples containing either the same color and texture patterns (I2IT) or shapes (T2IS), pointing out a major limitation in the mono-conditional generation scenario. On the other hand, ST is virtually unaffected by this issue, although it requires a well-formed image provided as a basis for the translation, limiting the expressiveness of the model.

In aRTIC GAN, we decide to address the *Mode Collapsing* problem by creating a multi-conditional scenario in which I2IT and T2IS are combined, contrasting each other's style suppression. In particular, we proposed as visual input the use of simple and minimal hand-drawn sketches and descriptive captions related to colors and patterns. These two conditioning are both user-friendly and simple to retrieve, differently from previous works that combine text caption only with position [11] and parts [12] information. Moreover, the concurrent use of simple sketches and text descriptions takes advantage of non-overlapping information: the illustration imposes hard spatial constraints, making the model able to identify pose and species, while the caption is in charge of pointing out the coloring.

In addition to the initial problem, we explored the possibilities of multi-object generation. One major problem we encountered is represented by different visual defects, such as checkerboard artifacts and the loss of detail, which may characterize the generated image. These issues are derived directly from the output morphology and must be solved by custom adjustments; thus, we opted for a sequential model, as opposed to the more common concurrent approach. The proposed method exploits a recurrent module in conjunction with a third visual conditioning represented by an initial background. In detail, this is used by the first generated object, and the resulted output is provided as a new scene for the following generation step. A visual example of the overall generation procedure is provided in Figure 1.

aRTIC GAN displays a compact design, composed of a set of blocks in charge of capturing the proper semantic, leading to a generation process enriched with detail enhancement or suppression (see Figure 2). This pipeline forwards the given inputs along a realistic inpainting procedure up to the intermediate and final results. The simplicity of this configuration allows the model to be called recursively over a series of different pairs of sketches and descriptions, while the rendering capabilities enable high levels of realism, limiting the generation possibilities only to the human imagination and artistic touch. The contribution of our method is threefold:

(i) aRTIC GAN exploits two specifically designed refinement blocks (Section 3.2) to deal with image artifacts and fine detail enforcement as opposed to the multi-stage generation approaches. This structure aims to achieve a small parameter count, much lower than the aforesaid counterparts, while still obtaining high-quality performance.

(ii) In order to use a unique discriminator for single-stage generation, our discriminator produces different outputs at the same time. This design allows us to simultaneously analyze text consistency and image quality at several levels without weighing down excessively the overall complexity. The model and the novel losses are described, respectively, in Sections 3.3 and 3.4.

(iii) The use of sketches and text descriptions improves performance while reducing the *Mode Collapsing* effects, since each constraint influences and dampens the variation suppression problem caused by the other input. As an additional effect, the generator appears to better discern elements from multiple domains and to generate them accordingly, boosting even more the all-round realism quality and the detail enhancement (Section 5.3).

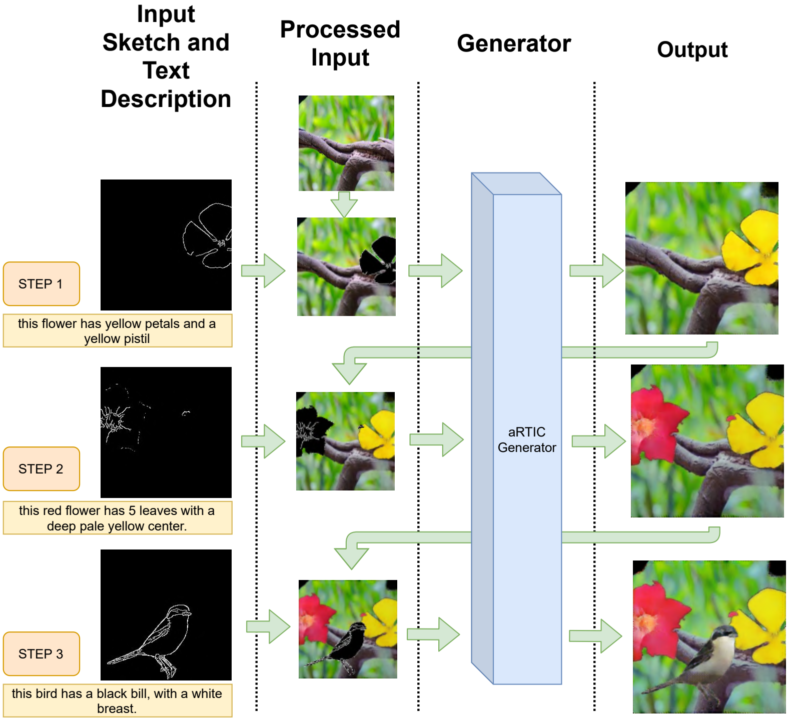
Input Sketch and Text Description
Processed Input
Generator
Output
STEP 1
this flower has yellow petals and a yellow pistil
STEP 2
this red flower has 5 leaves with a deep pale yellow center.
STEP 3
this bird has a black bill, with a white breast.
aRTIC Generator

Figure 1. Theproposed generation process of aRTIC GAN. The model is recursively fed with sketches composed of a few lines, short yet informative descriptions and either a background or the previous generated image.

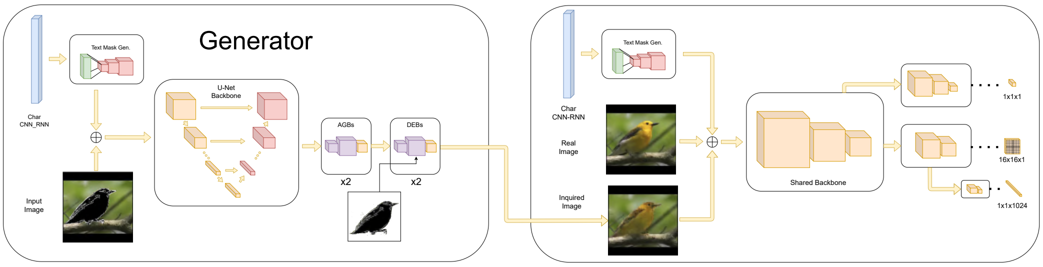

Figure 2. The overall architecture of aRTIC GAN. The orange blocks represent the convolutional layers and the red ones symbolize the transposed convolutional layers, while the green and purple represent the fully connected and bilinear upsampling blocks, respectively.

All the introduced novelties have been validated on two well-known datasets containing birds (CUB-200 [13]) and flowers (Oxford Flower 102 [14]) images. The correspondent text descriptions together with their embeddings (i.e., their vector representation), outlining attributes such as appearances and colors, are provided in a different dataset collected by Reed et al. [12].

The paper is organized as follows: Section 2 presents the state-of-the-art work, describing the main generation approaches; Section 3 discusses the main novelties of aRTIC GAN with respect to network blocks, losses and architecture; Section 4 describes the implementation details of the proposed method, from the input preparation to the network setup and training procedure; Section 5 shows the experiments and results, investigating comparative and ablation studies; Section 6 draws the conclusions of our work.

2. Related Works

Generating photo-realistic images is a challenging and significant task for many applications, such as computer graphics, digital design and art generation. Over the last few years, great progress has been achieved in this direction with the emergence of deep generative models. Among them, GANs [1] stood out for their capability to achieve several outstanding results, relentlessly improving as [15–19]. Moreover, the efficiency of GAN-based image processing has been proven in many other research areas such as object classification [20] or signal restoration [21].

Realistic image generation can be divided into five main categories with respect to conditioning types and modalities:

Image-to-Image Translation has significantly improved over the last years since the development of Pix2Pix by Zhu et al. [7], where conditional generative adversarial network is explored to learn a mapping from input to output images. CycleGAN was introduced by Zhu et al. [10] to convert the source domain into a different target domain in the absence of explicit paired example images. Wang et al. [8] proposed an improvement to generate high-definition images by progressively adding new generators. Chen et al. [22] presented an alternative approach by exploiting a cascade of refinement blocks. A recent turning point has been reached by Park et al. [23], who introduced a novel method based on semantic spatially adaptive normalization. Recently, Park et al. [24] proposed an approach based on patchwise contrastive learning and adversarial learning, while [25] explored a hierarchical tree structure to organize labels and a new translation process. A peculiar approach was developed by [26] which exploited a rich dataset collected through Artbreeder [27] to output a single image from a graph-like structure. Finally, Dai et al. [28] learned a sequence of invertible mappings which led to a flow-version of popular GANs, such as StarGAN, AGGAN and CyCADA, with similar performances but half of the training parameters.

Text-to-Image Synthesis was first pursued by Reed et al. [29], in which embeddings containing visual-relevant features are obtained taking advantage of the popular text embedding technique Char CNN-RNN [12]. Subsequently, Zhang et al. [5,6] introduced StackGAN in which the employment of multiple consecutive generator-discriminator pairs is explored. Xu et al. [9] developed AttnGAN, a model able to transform text description into spatial attention associating single words with image regions. The usage of a BI-LSTM text encoder rather than Char CNN-RNN allows the model to focus on both the general caption meaning and the single semantic word meaning, obtaining impressive results. The drawback is a very high complexity due to the fact that every semantic aspect part has to be controlled by a different component. Finally, Hong et al. [30] and Wang et al. [31] focused on the spatial constraints of the generated image. Other recent notable works are DM-GAN [32], MirrorGAN [33] and DF-GAN [34], using, respectively, a dynamic memory module to refine fuzzy images (DM-GAN), a mirror structure to model T2I and I2T subjects (MirrorGAN) and a single-stage architecture composed of a deep text–image fusion block and a target-aware discriminator (DF-GAN). Finally, Li et al. [35] proposed a lightweight GAN model with a novel word-level discriminator providing fine-grained training feedback; the corresponding generator is able to correctly focus on specific visual attributes while using a small number of parameters.

Style Transfer aims at transferring the style from one image onto another synthesising a texture from a source image preserving the semantic content of a target image. Gatys et al. [36] exploited one of the first attempts of texture modeling with deep learning. The multiple-domain transfer problem was widely addressed in [37–40], deepening the analysis on cross-domain relations, style merging and translation control, even in non-visual scenarios. This type of approach was explored in more specific fields, such as face modification, resulting, for example, in a family of architectures specialized in *age progression and regression*, such as CAAE [41] and its further development CAAE++ [42], C-GAN (Contextual GAN) [43], IPCGAN [44], idGAN [45] and CAN-GAN [46]. Recently, An et al. [47] explored the content leak issue in state-of-the-art methods and addressed it using neural flows, while [48] developed an adaptive attention normalization procedure to apply the attention mechanism on a per-point basis. A novel application of Laplacian pyramids was proposed by [49], which transfers global style patterns with a drafting network and further refines the local details with a revision network by hallucinating the previous output with a residual image.

The **Multi-Conditioned GANs** approach, although very demanding, nowadays results fundamentally in dampening the problem of *Mode Collapsing*; otherwise, it is much more difficult to deal with the "standard" mono-conditional methods. Reed et al. [12] proposed an architecture to generate objects from text descriptions and bounding boxes or object part locations. Bounding boxes determine the object region but cannot provide any information about the appropriate pose. On the other hand, the posture can be precisely described by selecting single parts' locations, although this methodology is significantly unnatural and time-consuming. H.Park et al. [11] introduced a model improving the generation process by focusing on object masks, while Dean et al. [50] developed a cascade GAN exploiting both class labels and audio signal inputs. Recently, astonishing results were achieved by T. Park et al. [23], combining semantic maps in combination with input semantic layouts. This method displays great visual performance, even showing capabilities of concurrent multiple-object generation. However, it lacks fine-detail control over the single object.

Multiple-Object Generation has been mainly addressed via a concurrent approach, characterized by a simultaneous spawning of multiple objects. Examples of concurrent generation are provided by [23,51–54]. Turkoglu et al. [55] proposed oppositely a sequential generation method based on a recursive approach, in which one single object is generated at a time from a segmentation map to deal with the occlusion artifacts problem. Finally, the model proposed by El et al. [56] produces sequentially simple shapes in a simulator [4], inpainting new geometrical objects in the scene.

3. Method

aRTIC GAN, as a Generative adversarial network (GAN), consists of a generator G and a discriminator D competing in a two-player minimax game: the discriminator tries to distinguish real training data from synthetic images, and the generator tries to fool the discriminator. aRTIC GAN takes as input an RGB image containing a background and the inpainted sketch (see Section 4.2) together with a text description embedding (Section 4.2.4). When dealing with the task of generating multiple objects, as opposed to multi-stage methods such as [5,6,9], our approach focuses on a single-stage generation process resulting in a much more compact design as shown in Figure 1. Specifically, at each step, the image produced as output by the generator is merged with the sketch for the consecutive generation process, resulting in a recurrent approach due to the exploitation of the same generator and discriminator.

In this Section we introduce the main novelties of aRTIC GAN, focusing on original network blocks, loss definitions and overall architecture composition.

3.1. Text Mask Generator

The cross-modal input of aRTIC GAN Generator is carried out through the **Text Mask Generator Block (TMG)** (shown in Figure 3), who is in charge of combining sketches and captions. First of all, text descriptions are encoded with the robust embedding method exploited by Reed et al. [29], whose performances are widely demonstrated in several works [5,6,9,11]. After that the text embedding is combined with the desired object sketch, represented here as a binary mask (in Section 4.2.2 are given details regarding the binary mask generation).

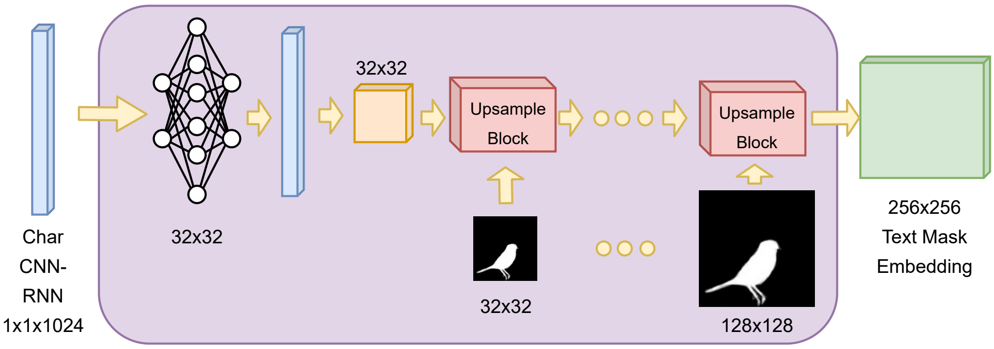

Figure 3. The Text Mask Generator structure exploits the binary masks, corresponding to the sketch area, into the *transposed convolutional* layers.

The composition of text embedding and object binary mask could in principle be performed by a single fully connected (FC) layer, resulting in a prohibitive parameters number. To overcome this issue and to preserve the separation between text and objects, we propose a series of progressive *upsampling blocks* fed at each level with the binary mask describing the desired location and rough shape of the object. Such a mechanism of spatial enforcement is reproduced in the generator and the discriminator so that each text channel is independently trained in both models. In Section 5.2 we present an overview of the configuration and performance of the TMG Block.

The TMG output is concatenated to the RGB input, i.e., an initial background image in case of the first object, or the previously generated scene in case of the subsequent ones, and fed to the aRTIC GAN Generator successive block (in Figure 2 is depicted the overall generation process).

3.2. Refinement Blocks

As shown in Figure 4, we dealt with two main issues to achieve photo-realistic images: large monochromatic areas are affected by grid and checkerboard artifacts, while finer details are insufficiently highlighted. To deal with these undesired effects, the generator is equipped with two refinement modules: the Anti-Grid Block (AGB) and the Detail Enforcement Block (DEB), respectively illustrated in Figures 5 and 6. The idea of grid-pattern removal in GANs is not novel [57,58], but our approach differs in the development of a specific modular block which can be applied several times in a row (e.g., 2 consecutive ones in the proposed architecture). The contributions of these two modules are further discussed in Section 5.4.

The **Anti-Grid Block (AGB)** is in charge of removing the grid-pattern artifacts due to small dimension kernels during the decoder steps of the generator. Our implementation is inspired by the ones proposed by Odena et al. [57] and Sugawara et al. [58]. The model performs an upsample operation by a factor of 2 via a *bilinear function*, and then a *convolutional layer* composed of three 8 × 8 kernels and a *hyperbolic tangent* activation function are employed.

As analyzed in detail in Section 5.4, the AGB effectively learns a smoothing function with improved performance with respect to classical smoothing algorithms [59,60] operating over the whole image and causing an excessive blur effect.

The **Detail Enforcement Block (DEB)** plays a key role in object generation by exploiting a bilinear upsampling and a convolutional downscaling combination analogous to the one in the AGB. The DEB input is given by the concatenation of the initial sketch with the AGB output. The objects' spatial constraints are enforced using 4×4 convolutional kernels, obtaining two significant improvements: the avoidance of small detail removal performed by the AGB (e.g., little bird eyes in Figure 4) and the enhancement of sharper image elements. The latter effect comes from mimicking a well-known behavior exploited in previous works dealing with stacked approaches, in which sharp details are generated by the last G-D pairs [5,6,9,50].

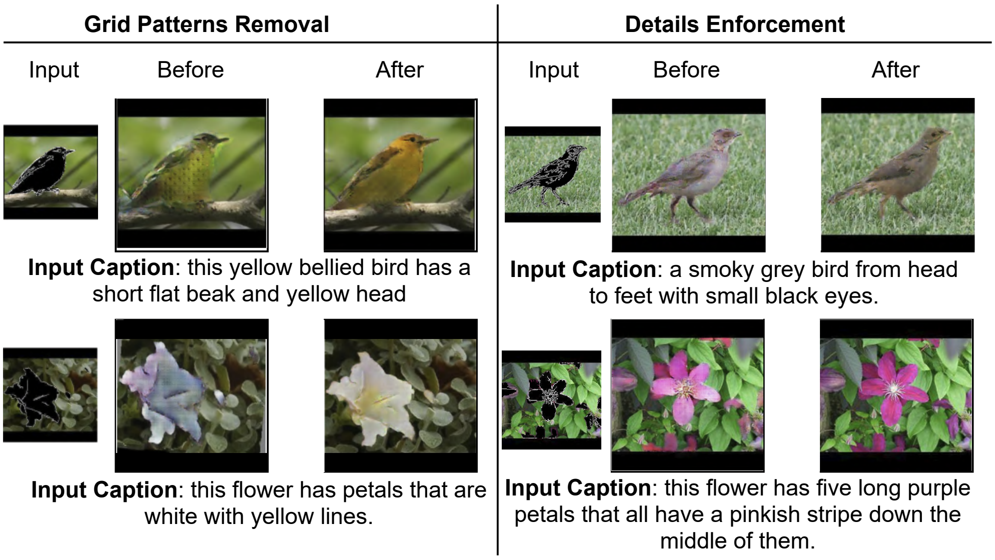
Grid Patterns Removal
Details Enforcement
Input
Before
After
Input
Before
After
Input Caption: this yellow bellied bird has a short flat beak and yellow head
Input Caption: a smoky grey bird from head to feet with small black eyes.
Input Caption: this flower has petals that are white with yellow lines.
Input Caption: this flower has five long purple petals that all have a pinkish stripe down the middle of them.

Figure 4. Single-step generation artifacts are shown along with the corresponding inputs and outputs.

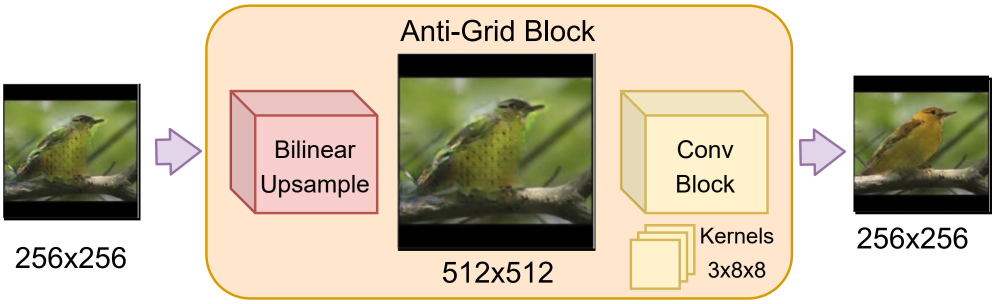

Figure 5. The Anti-Grid Block exploits a linear function to upsample the image, and 8 × 8 convolutional kernels provide the learned smoothing function.

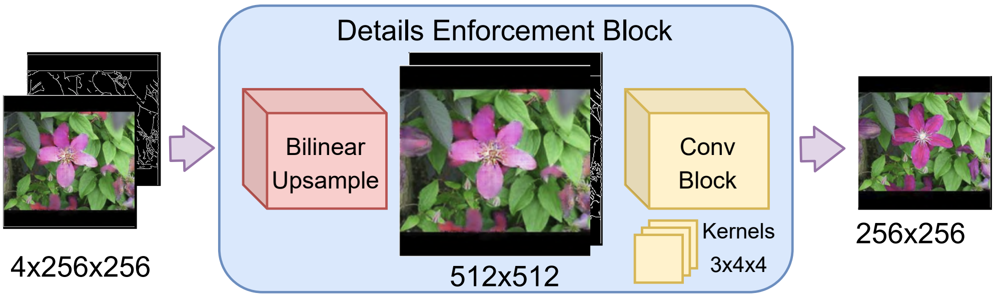

Figure 6. The Detail Enforcement Block design is similar to that of the AGB. The difference resides in the task of the convolutional layer, which retraces the input sketch for fine and sharp details.

3.3. Loss Functions

aRTIC GAN loss functions are based on several components to enforce all the constraints defined by TMG, AGB and DEB. Four types of cost functions have been implemented for both the generator and discriminator.

Binary and Patch Losses, introduced by [1,7], are widely used in image translation methods. In particular, the Binary Loss states the likelihood of an image, while the Patch Loss associates a confidence with the realism of each image region. aRTIC GAN exploits them together to enforce precise object generation with both fine details and global consistency (e.g., small-scale generated parts placed in a realistic way). Moreover, each of these losses provides a measurement on the image-caption consistency other than the Text Reconstruction Loss (described later in this section). Binary and Patch Losses are defined by applying a Mean Squared Error cost function, as proposed by Mao et al. [61]:

$$\mathcal{L}_{Binary} = \mathbb{E}[(D_{Bin}(z,y) - 1)^2] + \mathbb{E}[(D_{Bin}(z,G(z)))^2] \tag{1}$$

$$\mathcal{L}_{Patch} = \mathbb{E}[(D_{Patch}(z,y) - 1)^2] + \mathbb{E}[(D_{Patch}(z,G(z)))^2] \tag{2}$$

where D_{Bin} is the binary output of the discriminator, D_{Patch} is the patch output of the discriminator, G is the generator output, y is the ground truth image and z is the concatenation of the RGB image with the inpainted sketch and the text description embedding.

Double L1 Loss deals with the synthesized object and the recursive generation of multiple elements. aRTIC GAN computes this criterion function twice: the first one is calculated over the whole image to ensure background consistency; the second one is computed on the eroded sketch area, as introduced by [11], to ensure the visibility of the object.

Double L1 Loss is defined as follows:

$$\mathcal{L}_{L1} = \mathbb{E}[||y - G(z)||_1] + \mathbb{E}[||(y_{Masked}) - (G(z)_{Masked})||_1], \tag{3}$$

where y_{Masked} is the eroded ground truth and G_{Masked} is the eroded output of the generator.

Text Reconstruction Loss (TRL) is fundamental for preserving semantic information represented by the text embedding in the discriminator architecture. Without this specific loss, semantic information would be ignored leading to *Mode Collapsing*. Since Binary and Patch Losses focus mainly on spatial consistency, TRL ensures coherence between color patterns and referring captions, as shown in Section 5.4. Text Reconstruction Loss is defined using the Cosine Similarity (CS) as follows:

$$\mathcal{L}_{TRL} = \mathbb{E}[\mathrm{CS}(z_{text}, D_{TRL}(z,y))] + \mathbb{E}[\mathrm{CS}(z_{text}, D_{TRL}(z,G(z)))] \tag{4}$$

where D_{TRL} is the text reconstruction output of the discriminator and z_{text} is the input text embedding.

The final **Generator and Discriminator Losses** take the following forms:

$$\mathcal{L}_G = \alpha\mathcal{L}_{Binary} + \beta\mathcal{L}_{Patch} + \lambda_1\mathcal{L}_{L1_{image}} + \lambda_2\mathcal{L}_{L1_{target}} \tag{5}$$

$$\mathcal{L}_D = \alpha\mathcal{L}_{Binary} + \beta\mathcal{L}_{Patch} + \gamma\mathcal{L}_{TRL}, \tag{6}$$

where $\mathcal{L}_G$ is the Generator Loss, $\mathcal{L}_D$ is the Discriminator Loss and the weight parameters $\alpha, \beta, \lambda_1, \lambda_2, \gamma$ are defined in Section 4.3.

3.4. aRTIC GAN Architecture

One of the main advantages of aRTIC GAN is the tight model composed of a single generator–discriminator pair. The discriminator's multi-head structure balances between the network compactness and the number of constraints given to the model in the previously described loss.

The overall architecture is shown in Figure 2. The generator G is a mixture of four components: the TMG, the encoder-decoder (E-D) backbone and the two refinement blocks. The output of the Text Mask Generator is fed to the E-D backbone. This core unit has a U-net [62] structure and is in charge of analyzing features at several scales while feeding the upsampling blocks with the corresponding encoder information. The final output image is given by the two refinement blocks working in pipeline. The discriminator D has been designed to take advantage of two feature-extraction backbones and is composed of 10 convolutional layers. It is able to perform three different tasks in parallel: binary classification, patch classification and text reconstruction.

4. Implementation Details

A detailed overview of the technical aspects is provided here, starting from the datasets employed in our experiments, the input text and sketches preprocessing, the network setup and its training procedure.

4.1. Datasets

The **Caltech-UCSD Birds 200 (CUB-200)** [13] dataset is composed of more than 11,000 images depicting 200 bird species. Images are associated with objects' bounding boxes, segmentation masks and attributes.

The **Oxford Flowers 102** [14] dataset is made up of more than 8000 images belonging to 102 flower categories. For each class, we can find a total number of n images, with $40 \leq n \leq 258$, resulting in an unbalanced dataset. One of the main advantages of Oxford Flower 102 is a high variation with respect to flowers pose and light conditions in images, even within the same class.

The **Birds and Flowers Captions** [62] dataset describing the images contained in CUB-200 and in Oxford Flowers 102, is provided by Reed et al. [12]. It consists of 10 short text captions for each image, together with their 1024 vector embeddings, retrieved via *Amazon Mechanical Turk*. The text descriptions provide attributes such as appearances and colors of birds and flowers.

The choice to employ the aforementioned datasets is due to the visual chromatic variance which can be found in their images, a consequence of brightness, saturation and high contrast provided by colors in nature. Other types of datasets, even among the most common in this field such as COCO [63], ADE20K [64] or Cityscapes [65], characterized by urban environments, display less variance. Moreover, the amount and high quality of the human-made captions provided by [12] makes a strong contribution for the dataset choice.

4.2. Input Preparation

We discuss here the RGB input image preparation, the sketch generation process and the text embedding. Unless specifically reported, all the image data are resized to 256 × 256 pixels, which is the standard aRTIC GAN working resolution.

4.2.1. Background Generation

Background images are obtained directly from the CUB-200 dataset, upscaling the bird images and isolating the four 256 × 256 external edge squares, as suggested in [11].

4.2.2. Sketch Mask and Input Generation

One of the two aRTIC GAN inputs is a merging of a sketch and an RGB background, which is either a random background image (in case of the first generated object) or the output of the previous generation step. The sketch is inpainted into the background image by replacing the corresponding pixels and the area within the sketch. At training time, the objects' binary masks are taken from CUB-200 and Oxford Flowers 102 datasets. On the other hand, if a sketch is provided at test time, the binary mask is generated through *closure morphological operator* and *Fill* algorithms. The binary masks are resized to 32 × 32, 64 × 64 and 128 × 128 to feed the corresponding upsampling block in the TMG (see Figure 3).

4.2.3. Sketch Generation

In order to automatically reproduce plausible human-made sketches for the images in CUB-200 and Oxford Flowers 102 datasets, we apply an edge detection strategy on the ground truth images. The Canny algorithm [66] is chosen as the baseline method for this purpose taking into account two separate thresholds to regulate the sensitiveness of the resulting sketch with respect to the amount of details. The choice of the sensitivity threshold level is particularly tricky in the case of the CUB-200 dataset because low values lead to noisy details while high values can remove contours and structural details (e.g., birds' eyes). Two different Canny edge detectors are then employed, one working on the elements inside the object and one working on the segmentation mask to obtain the contours. Finally, the outcomes are summed up and normalized in $[-1, 1]$. Examples of the sketch generation procedure are displayed in Figure 7.

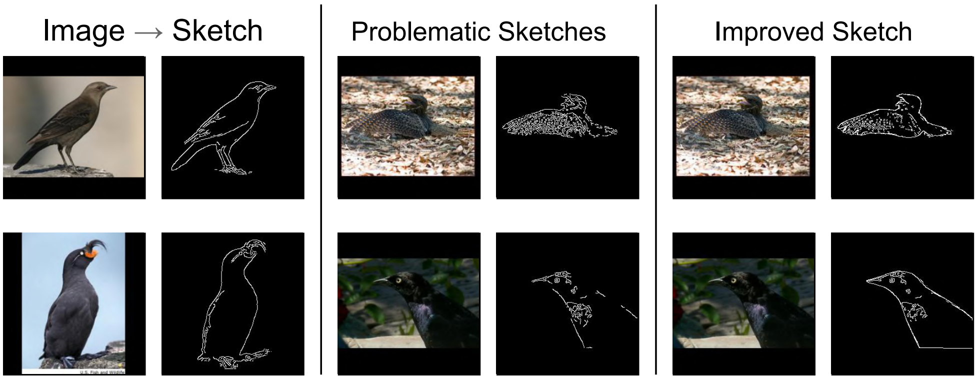

Figure 7. Sketch generation outcomes and the related issues are reported. Our algorithm efficiently deals with complex cases.

4.2.4. Char CNN-RNN Text Embedding

Char CNN-RNN is employed to extract the visually discriminative text embedding of a given description, the second input of aRTIC GAN. This method was proposed by Reed et al. [12] to pretrain a text encoder and it is largely used in Text-to-Image Synthesis tasks. It maps text descriptions to the common images feature space by learning a correspondence function between text and images. The choice to employ the aforementioned model in the evaluation phase is to pursue a fair comparison of aRTIC GAN with other generative models.

4.3. Network Setup

aRTIC GAN inputs have dimensions of 256 × 256 × 3 for the RGB image and 1 × 1 × 1024 for the text embedding. The text embedding feature vector is fed to the TMG, resulting in a 256 × 256 × 1 features map, which is concatenated to the RGB tensor as an additional channel.

The encoder-decoder structure is composed of eight convolutional blocks for the downsampling stage and seven transposed convolutional blocks for the upsampling. The encoder is equipped with batch normalization (BN) [67] and the tanh activation function, with an exception for the first layer. The decoder is equipped with BN and ReLU activation functions. A dropout strategy is used in the first three blocks to improve the robustness of the model. We employed two AGBs and two DEBs with filters of dimension 8 × 8 and 4 × 4, respectively (see Figure 2).

The discriminator input has size 256 × 256 × 7, obtained by the concatenation of the inquired image, the Text Mask and the ground truth (GT) image. The two shared architectures are composed of three downsampling blocks (with a factor of 2). Every convolutional layer is followed by BN and ReLU activation functions. The final discriminator outputs representing the Binary, Patch and Text Reconstruction tensors, have dimensions of 1 × 1 × 1, 16 × 16 × 1 and 1 × 1 × 1024, respectively. The Binary and Patch Losses exploit a Mean Square Error (MSE) cost function, while Text Reconstruction utilizes the Cosine Similarity (CS). The weights chosen for the two final loss functions (Equations (5) and (6)) are: $\alpha = 0.6$, $\beta = 0.1$, $\gamma = 1.0$, $\lambda_{tot} = 100$ and $\lambda_{mask} = 100$. Our model is trained with a batch size of 3 for 200 epochs with image augmentation and for an additional 100 epochs with background augmentation (experiments described in Section 5.2). Finally, aRTIC GAN is trained with single-object generation, as each generation is independent from the others.

4.4. Training Procedures

aRTIC GAN can generate an object in a scene while preserving the given background up to a plausible inpainting using only a single generator–discriminator pair. When generating multiple objects, this structure allows the model to handle each object generation as almost completely independent from the others. Indeed, occlusions and artifacts are processed separately at each step.

Accordingly, two training strategies have been developed for aRTIC GAN: independent steps learning and random consecutive steps training.

The **Independent steps learning** strategy is based on the idea of training aRTIC GAN on a single object at a time. The whole procedure is summarized in Algorithm 1. The input background is either composed by random environment surroundings or a GT image belonging to one of the two datasets used, mimicking the context of multi-object images. The gradients for both the generator G and discriminator D are computed at each step as well as their weights update. This strategy allows us to provide the final image starting from any type of sketch, description and background, avoiding any dependency with respect to the position, patterns and relative occlusions from other inputs. This modality has been exploited for all the results shown in our work.

Algorithm 1: Independent steps learning

```
Input: Inpainted sketch, text embedding and GT image
Output: Generated image and weights update step
Generator Call over the inpainted sketch and text embedding
Discriminator Call over the generator output and the GT
Generator Losses: Binary, Patch and Double L1
Discriminator Losses: Binary, Patch and Text Reconstruction
Gradient Computation
Weights Update;
```

Random consecutive steps training is an alternative procedure based on a sequence of generation steps and a single-discriminator evaluation. The procedure is summarized in Algorithm 2, where the input is defined as a set of m, with $1 \leq m \leq 4$, sketch-text embedding pairs. The final image is generated via consecutive calls of G, where the current sketch is inpainted in the output of the previous step (except for the first object, which is inpainted in the initial background input), while the Double L1 Loss is computed and stored at each step. After all the generations are completed, the discriminator is called and its loss is computed along with the Binary and Patch components of the Generator Loss in order to perform a single gradient descent step.

Algorithm 2: Random consecutive steps training

```
Input: List of inpainted sketches and text embeddings
Output: Generated image and weights update step
for Input sketch–text pair do
    Generator Call over the inpainted sketch and text embedding
    Generator Double L1 Losses: $G_{D_{L1}}$
Discriminator Call over the generator output and the GT
Discriminator Losses: Binary, Patch and Text Reconstruction
Generator Losses Aggregation: $G_{tot} = G_{Binary} + G_{Patch} + \sum_i G^i_{D_{L1}}$
Gradient Computation
Weights Update;
```

5. Experiments and Results

Extensive experiments have been carried out to prove the validity of the proposed method. In Section 5.1, we report the metrics used to measure aRTIC GAN performance.

In Section 5.2, we evaluate the generation of multiple elements, highlighting the behavior of aRTIC GAN with respect to different backgrounds. A comparative analysis is then provided in Section 5.3. Finally, an ablation study of the main components is described in Section 5.4.

5.1. Metrics

A quantitative evaluation has been performed using several metrics: the Inception Score (IS) [17] (the higher the better) for the realism of the generated image, the Structure Similarity (SSIM) [68] to quantify the similarity between the GT and the generated images ($\text{SSIM}_{\text{GTgen}}$), the Cosine Similarity (CS) to measure text descriptions' variations and the Frèchet Inception Distance (FID) [69] (the lower the better), which compares the distribution of the generated images with the distribution of the training images. The definitions of the cited metrics are shown below.

IS is a measure of the characteristics of a generative model:

$$\text{IS} = \exp[\mathbb{E}_{z\sim p(z)}[\mathbb{D}(p(y|g(z)))\|p(y)]] \tag{7}$$

where $g(z)$ is an image generator to be evaluated, y is the label, $p(y|x)$ is the posterior probability of a label computed for an image x, $p(y) = \int p(y|g(z))dz$ is the marginal class distribution and $\mathbb{D}(p\|q)$ is the KL-divergence between two probability distributions p, q.

The Structure Similarity (SSIM) estimates the visual impact of shifts in image luminance, changes in image contrast and structural changes. The SSIM between two image windows $\mathbf{x}$ and $\mathbf{y}$ with the same dimension is defined as:

$$\text{SSIM}(\mathbf{x},\mathbf{y}) = [l(\mathbf{x},\mathbf{y})]^{\alpha}[c(\mathbf{x},\mathbf{y})]^{\beta}[s(\mathbf{x},\mathbf{y})]^{\gamma} \tag{8}$$

where $\alpha, \beta, \gamma > 0$ control the significance of each of the three terms. The luminance, contrast and structural components are defined as:

$$\begin{aligned} l(\mathbf{x},\mathbf{y}) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \\ c(\mathbf{x},\mathbf{y}) &= \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ s(\mathbf{x},\mathbf{y}) &= \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \end{aligned} \tag{9}$$

where μ_x and μ_y represent the means of the two images, σ_x and σ_y represent the standard deviations, σ_{xy} is the covariance of the two images and $C_1, C_2, C_3 \in \mathbb{R}$.

The Frèchet Inception Distance (FID) is the Wasserstein distance between two multivariate normal distributions $X_r \sim \mathcal{N}(\mu_r, \Sigma_r)$ and $X_g \sim \mathcal{N}(\mu_g, \Sigma_g)$:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}) \tag{10}$$

Furthermore, an additional metric has been defined to evaluate the capability of the generator to reproduce the target domain distribution for the given conditional input. In particular, we compute ΔSSIM as the absolute difference between the similarity obtained on the ideal transformation (inputGT) and the one performed by aRTIC GAN (inputGen) as follows:

$$\Delta\text{SSIM} = |\text{SSIM}_{\text{inputGT}} - \text{SSIM}_{\text{inputGen}}| \tag{11}$$

where $\text{SSIM}_{\text{inputGT}}$ and $\text{SSIM}_{\text{inputGen}}$ are the structural similarities computed over the input image with respect to the ground truth and the generated output.

Finally, we exploit the Cosine Similarity (CS) metric to understand text embeddings' variations and similarities.

The CS between two vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as:

$$\mathrm{CS}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} \tag{12}$$

The CS measures in fact the similarity between two vectors of an inner product space on the basis of their angle cosine. A CS value equal to zero means that the two vectors are orthogonal, while two vectors pointing up to the same direction correspond to a CS value equal to one. We refer to the CS value of a dataset as the mean value of all CSs calculated on 10^4 different vector combinations among the dataset text embeddings. Furthermore, we refer to the CS value of a dataset pair as the mean value of all CSs calculated on 10^4 pairs of vectors taken from the respective datasets. This value can be empirically used as the distance between the two datasets' text embeddings.

5.2. aRTIC GAN Evaluation

Table 1 reports the performance of aRTIC GAN in terms of IS, SSIM and ΔSSIM over the two datasets and their combined usage. An interesting result is represented by the increasing performance in terms of IS when training aRTIC GAN over both domains (birds and flowers), keeping the same hyperparameters setting. This highlights the importance of using a multi-domain dataset to improve the quality of image generation.

Table 1. The Inception Score (IS), the Structural Similarity Score (SSIM) and the ΔSSIM (Equation (11)) are reported, respectively, for the generation of birds, flowers and their combined use.

Distance	**CUB-200**	**Flowers 102**	**Multi-Domain**
IS	5.54 ± 0.33	4.28 ± 0.31	7.15 ± 0.34
SSIM	0.86	0.71	0.80
ΔSSIM	0.04	0.045	0.042

The combined use of text embeddings and sketches is analyzed in Table 2 in relation to the respective datasets (i.e., CUB-200 [13] and Oxford Flower 102 [14]).

Table 2. The distance between the datasets is calculated using the SSIM and CS.

Distance	**CUB-200**	**Flowers 102**	**CUB-Flowers**
SSIM	0.072 ± 0.192	0.057 ± 0.121	0.062 ± 0.053
CS	0.27 ± 0.178	0.64 ± 0.092	−0.04 ± 0.034

The reported CS values in the case of single dataset show the coherence of text embeddings in the Oxford Flower and CUB-200 datasets, with the latter showing a smaller value because of the greater variety of text descriptions of birds with respect to flowers. The almost-zero CS value calculated between the two datasets suggests that the respective embeddings are almost orthogonal; thus, a multi-domain training will increase the model generalization.

A qualitative analysis is provided in Figure 8, showing a large set of resulting images with respect to several conditions (e.g., single-object, multiple-object generation). The left block shows examples of aRTIC GAN outputs when trained on both the domains, while the middle and right blocks display results provided by single domain training procedure. By the combined use of the two domains (left block), aRTIC GAN is able to improve the realism of color-pattern textures, boosting the IS value of about ∼2 points (see Table 1).

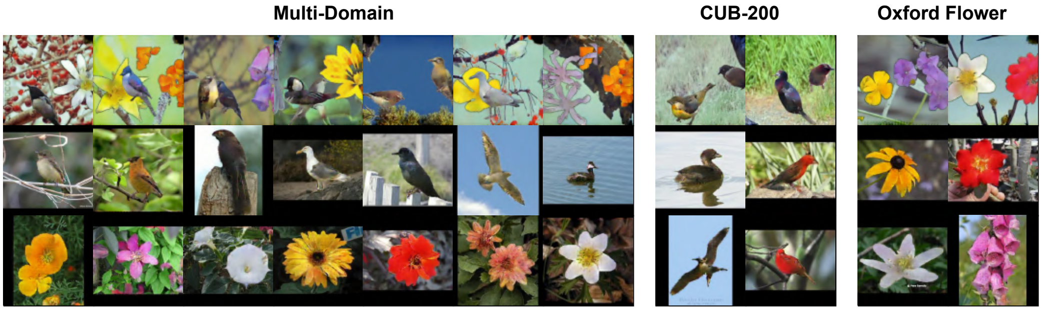

Figure 8. Examples of images generated by aRTIC GAN, trained over single and multiple domains. Examples of multiple-element generation are shown as well as of single birds and single flowers.

Finally, the possibility of generating objects out of their original environments has to be considered since random backgrounds can be provided as inputs to our model. In order to deal with this issue, we investigated a fine-tuning training strategy over an augmented dataset using the procedure introduced in [11]. The GT object and relative sketch pair is inpainted both into the original background and into a randomly chosen one. Table 3 reports the obtained IS, which slightly decreases in the more challenging setting of complete random backgrounds. In Figure 9, some background images and original-augmented GT image pairs are presented in the first and second rows, respectively. The last two rows provide examples of the generation process employing or not the background augmentation technique. The not-augmented model, while trying to preserve the background, propagates black regions from the input image, resulting in unrealistic outputs.

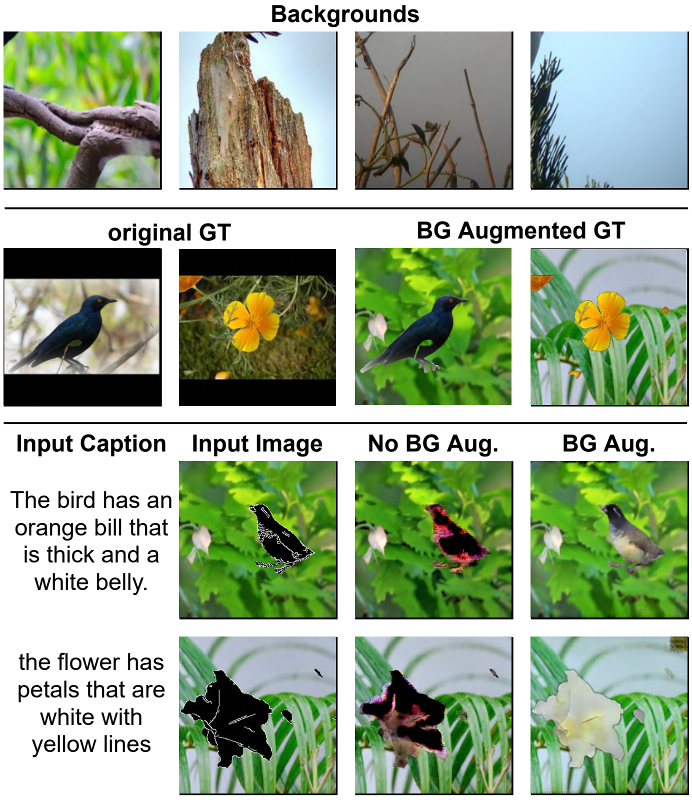
Backgrounds
original GT
BG Augmented GT
Input Caption
Input Image
No BG Aug.
BG Aug.
The bird has an orange bill that is thick and a white belly.
the flower has petals that are white with yellow lines

Figure 9. Backgrounds, GT augmentations and examples of generation failures and successes are reported here, showing the importance of the generalization step.

5.3. Comparison with the State of the Art on Single Domain

A straightforward comparison between aRTIC GAN and other methods is not trivial. As already mentioned, T2IS and I2IT lack, respectively, spatial and color-pattern constraints, possibly leading to *Mode Collapse* events. Even if the Inception Score (IS) outputs a lower score in the case of Mode Collapse [70], the actual supervision is hard to be measured: neither text-based methods, e.g., DF-GAN [34], DM-GAN [32] and StackGAN [5], nor sketch-based methods, e.g., Pix2PixHD [8], CUT [24] and CycleGAN [10], have the expressive power required to enforce both the constraints (details and textures). On the other hand, multi-conditioned methods [11] actually exploit the combined use of background images and text descriptions, but no actual control over the object's shape or details is provided, resulting in a slight variation in the Text-to-Image Synthesis task.

Moreover, the comparison with another multi-conditioned model, SPADE by Park et al. [23], is not an easy task due to the different inputs used. Pairing style images to text descriptions is far from being a trivial task, and no assurance on providing the same amount of information can be given. If no style image is fed to the network, SPADE falls into the same category as Pix2Pix and Pix2PixHD.

Table 3. The IS of the standard aRTIC GAN is compared with the BG-Augmentation setting.

Model	Multi-Domain IS
aRTIC GAN	7.20 ± 0.29
BG-Augmentation	6.28 ± 0.18

Nonetheless, we compared aRTIC GAN with several baseline models in single-domain setting in order to provide a general indicator for the realism quality of the generated images; these include the aforementioned stacked approaches, more modern architectures and even a single-stage generation competitor model, due to their high reputations among the scientific community and the availability of common metric scores on CUB-200 and Oxford Flowers 102 datasets.

Table 4 reports the achieved results in terms of Inception Score and FID score, respectively: the competitors' values have been taken directly from the original papers with the exception of Pix2Pix, which had to be trained specifically for the sake of comparison purposes. Even if our aRTIC GAN aims to tackle a more challenging objective with respect to I2IT and T2IS, due to the used inputs, the IS achieved on CUB is 5.54 ± 0.33 and the FID score is 14.17, while on Oxford Flowers the IS achieved is 4.28 ± 0.31, resulting in extremely competitive performances in both datasets: LeicaGAN* (which is LeicaGAN trained with a custom training-testing split) is the only one performing slightly better on CUB, even if it lacks the quality of controls of our proposed method, as it displays a mono-conditional I2T approach; aRTIC GAN performs better than all the presented competitors' models on Oxford Flowers dataset.

For a qualitative analysis, Figure 10 demonstrates how aRTIC GAN can control the output image, given the same background and text caption, through a change in the input sketch. In the provided examples, our method is able to generate the correct poses and species, as opposed to the methodologies that exploit random noise to generate different output images. The example in Figure 11 shows how our method reacts to a fixed-input sketch and different captions: aRTIC GAN is able to produce different coloring on request, proving the avoidance of *Mode Collapses* as opposed to other competitors' I2IT methods.

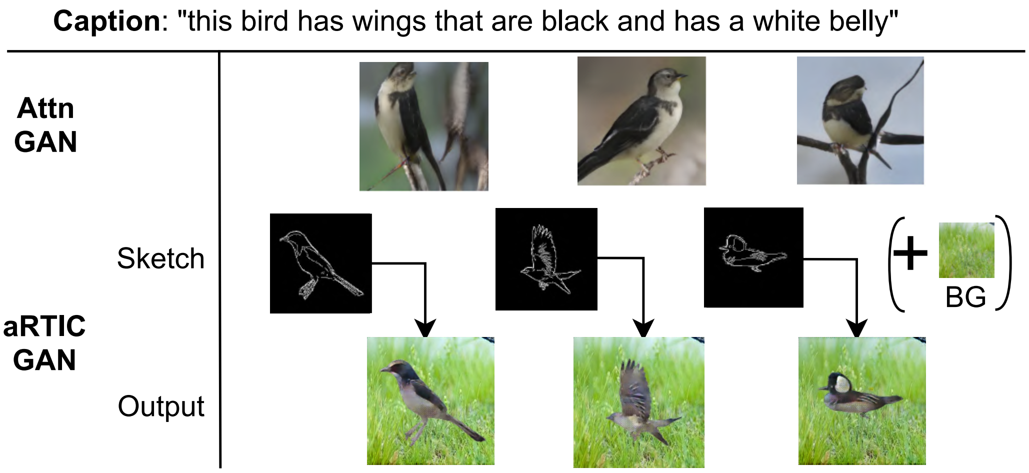
Caption: "this bird has wings that are black and has a white belly"
Attn GAN
Sketch
aRTIC GAN
Output
+
BG

Figure 10. The supervision of the generation process is committed via the sketch, generating different poses and the species with the same caption.

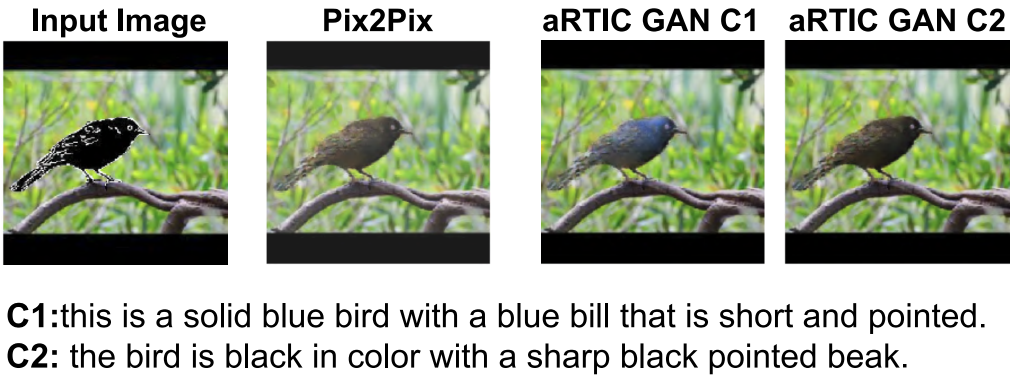
Input Image
Pix2Pix
aRTIC GAN C1
aRTIC GAN C2
C1:this is a solid blue bird with a blue bill that is short and pointed.
C2: the bird is black in color with a sharp black pointed beak.

Figure 11. The use of a single sketch and multiple captions (e.g., C1 and C2) allows aRTIC GAN to generate different color patterns as opposed to I2IT methods (e.g., Pix2Pix).

Table 4. Acomparison between our method and other models using IS and FID scores as metrics computed over both CUB-200 and Flowers 102.

Architecture	CUB-200 (IS)	Flowers 102 (IS)	CUB-200 (FID)
Pix2Pix [7]	2.76 ± 0.13	2.62 ± 0.023	-
GAWWN [12]	3.62 ± 0.07	-	67.22
AttnGAN+CL [9]	4.42 ± 0.05	-	16.34
StackGAN [5]	3.70 ± 0.04	3.20 ± 0.01	51.89
StackGAN V2 [6]	4.04 ± 0.05	3.26 ± 0.01	15.30
MirrorGAN [33]	4.56	-	34.71
LeicaGAN [71]	4.62 ± 0.06	3.92 ± 0.02	-
LeicaGAN* [71]	5.69 ± 0.06	3.80 ± 0.01	-
DM-GAN [32]	4.75	-	16.09
SEGAN [72]	4.67 ± 0.04	-	18.167
DF-GAN [34]	5.10	-	14.81
aRTIC GAN	**5.54 ± 0.33**	**4.28 ± 0.31**	**14.17**

5.4. Ablation Study

We perform an ablation study to highlight the effects of the Patch Loss, the Text Reconstruction Loss, the Anti-Grid Block and the Detail Enforcement Block. These elements are compared using a conditional discriminator baseline with the Binary Loss, where the *Gaussian Smoothing* and *Sharpening* techniques replace the proposed refinement blocks.

The Binary Loss alone produces globally consistent yet inaccurate images, failing to correct heavy image artifacts produced by the generator. We found that setting β to 0.6 in Equation (6) improves detail generation while keeping a global coherence, as shown in Figure 12. The combined use of the Binary and Patch Losses makes the discriminator focus on spatial constraints, while the generator tends to choose the median pixel color to minimize the two L1 losses [7].

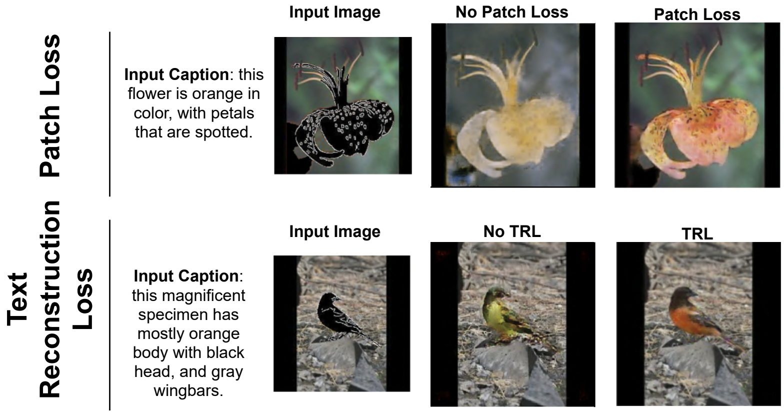
Patch Loss
Input Caption: this flower is orange in color, with petals that are spotted.
Input Image
No Patch Loss
Patch Loss
Text Reconstruction Loss
Input Caption: this magnificent specimen has mostly orange body with black head, and gray wingbars.
Input Image
No TRL
TRL

Figure 12. Region consistency and color-pattern generation are compared when using the Patch Loss and TRL, respectively.

The TRL enforces the text description, as shown in Figure 12, where the generation process is sensible to the input caption resulting in the correct color pattern. Moreover, Table 5 reports several implementations of the TMG, as in Section 3.1, and the corresponding number of parameters of the generator and the Inception Score. We empirically found that the best performance is obtained through the 32 × 32 implementation, which demonstrates the capability of this block to generalize the information provided by text.

Table 5. The TMG implementations are here presented to discuss performance with respect to the IS and the number of parameters.

Mode	Parameters	Inception Score
Fully connected	116.086.915	6.56 ± 0.24
FC 16 × 16 + upsample	**49.174.983**	6.96 ± 0.23
FC 32 × 32 + upsample	49.962.166	**7.20 ± 0.29**
FC 64 × 64 + upsample	53.110.949	6.96 ± 0.22

Here we investigate the use of the *Gaussian Smoothing* and *Sharpening* techniques when replacing the Anti-Grid Block and the Detail Enforcement Block. As shown in Figure 13, these filters affect the whole image neither preserving the background nor adapting to specific cases. Instead, the AGB removes the grid-pattern artifacts without influencing the contours and the surroundings. On the other hand, DEB contrasts the lack of details and sharp lines.

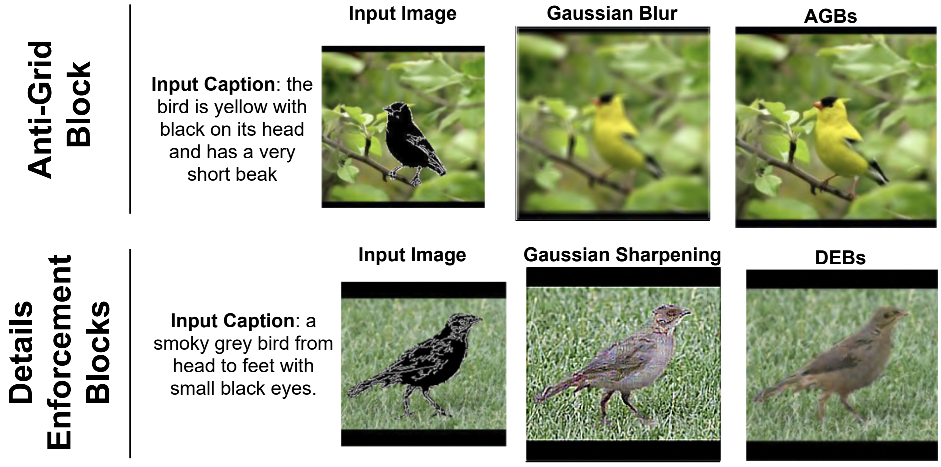
Anti-Grid Block
Input Caption: the bird is yellow with black on its head and has a very short beak
Input Image
Gaussian Blur
AGBs
Details Enforcement Blocks
Input Caption: a smoky grey bird from head to feet with small black eyes.
Input Image
Gaussian Sharpening
DEBs

Figure 13. Gaussian Blur and Sharpening global action are compared to the AGB and DEB local action.

For a more holistic view of the ablation study, Table 6 shows the effects, in terms of the FID metric, derived from the suppression of various components in the complete model, highlighting and quantifying their importance in the presented architecture.

Table 6. Here are presented the FID scores retrieved when the indicated component is removed from the complete model.

Removed Comp.	FID
aRTIC GAN	14.17
TMG (Section 3.1)	22.83
AGB (Section 3.2)	48.42
DEB (Section 3.2)	39.71
AGB + DEB	83.47

6. Conclusions

In this paper, we proposed aRTIC GAN, a method to recursively generate images conditioned on multiple text descriptions and object sketch pairs. aRTIC GAN improves the supervision on the image generation process by exploiting each constraint to tackle *Mode Collapsing*. Moreover, our approach aims at generating the inquired object as well as preserving both the semantics and the details of the given background image via a single-step generation. The proposed model uses foreground and background information to produce photo-realistic images. The experimental results suggest that our chosen input modalities significantly improve image diversity, enhancing the robustness of the model. Finally, the three novel network blocks, namely the Text Mask Generator, Anti-Grid and Detail Enforcement Blocks, boost the model's performances, offering a text embedding channel and the possibility of dealing with image artifacts.

Author Contributions: Conceptualization, E.A., C.A.C. and M.C.; methodology, E.A., C.A.C. and M.C.; software, E.A., C.A.C. and M.C.; validation, E.A., C.A.C., M.S. and P.R.; data curation, E.A., C.A.C. and M.C.; writing, E.A., C.A.C. and P.R.; editing, E.A., M.S., P.R. and I.A.; supervision, P.R. and I.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by European Union's Horizon 2020 Research and Innovation programme under grant agreement No 643950, project SecondHands.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. Advances in Neural Information Processing Systems. Available online: https://papers.nips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html (accessed on 22 May 2012).
2. LeCun, Y.; Cortes, C.; Burges, C. MNIST Handwritten Digit Database. 2010; Volume 2. Available online: http://yann.lecun.com/exdb/mnist (accessed on January 2021) .
3. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
4. Kim, J.H.; Kitaev, N.; Chen, X.; Rohrbach, M.; Zhang, B.T.; Tian, Y.; Batra, D.; Parikh, D. CoDraw: Collaborative drawing as a testbed for grounded goal-driven communication. *arXiv* **2017**, arXiv:1712.05558.
5. 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.
6. 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.* **2019**, *41*, 1947–1962. [CrossRef] [PubMed]
7. 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.
8. 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.
9. 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.
10. 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.
11. Park, H.; Yoo, Y.; Kwak, N. Mc-gan: Multi-conditional generative adversarial network for image synthesis. *arXiv* **2018**, arXiv:1805.01123.
12. 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.
13. Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; Perona, P. *Caltech-UCSD Birds 200*; Technical Report CNS-TR-2010-001; California Institute of Technology: Pasadena, CA, USA, 2010.
14. Nilsback, M.E.; Zisserman, A. Automated Flower Classification over a Large Number of Classes. In Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing, Bhubaneswar, India, 16–19 December 2008.
15. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
16. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier gans. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2642–2651.
17. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. *arXiv* **2016**, arXiv:1606.03498.
18. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv* **2017**, arXiv:1710.10196.
19. Brock, A.; Donahue, J.; Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv* **2018**, arXiv:1809.11096.
20. Bejiga, M.B.; Melgani, F. Gan-based domain adaptation for object classification. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 1264–1267.
21. Jin, X.; Chen, Z.; Lin, J.; Zhou, W.; Chen, J.; Shan, C. Ai-gan: Signal de-interference via asynchronous interactive generative adversarial network. In Proceedings of the 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Shanghai, China, 8–12 July 2019; pp. 228–233.
22. Chen, Q.; Koltun, V. Photographic image synthesis with cascaded refinement networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1511–1520.
23. Park, T.; Liu, M.Y.; Wang, T.C.; Zhu, J.Y. Semantic image synthesis with spatially-adaptive normalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2337–2346.

24. Park, T.; Efros, A.A.; Zhang, R.; Zhu, J.Y. Contrastive Learning for Unpaired Image-to-Image Translation. In Proceedings of the Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; Springer International Publishing: Cham, Switzerland, 2020; pp. 319–345.
25. Li, X.; Zhang, S.; Hu, J.; Cao, L.; Hong, X.; Mao, X.; Huang, F.; Wu, Y.; Ji, R. Image-to-image translation via hierarchical style disentanglement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8639–8648.
26. Gokay, D.; Simsar, E.; Atici, E.; Ahmetoglu, A.; Yuksel, A.E.; Yanardag, P. Graph2Pix: A Graph-Based Image to Image Translation Framework. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2001–2010.
27. Artbreeder. Available online: https://www.artbreeder.com/ (accessed on 1 April 2022).
28. Dai, L.; Tang, J. iFlowGAN: An invertible flow-based generative adversarial network for unsupervised image-to-image translation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, 3062849. [CrossRef]
29. 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, NY, USA, 20–22 June 2016; pp. 1060–1069.
30. 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 (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
31. Wang, M.; Lang, C.; Liang, L.; Feng, S.; Wang, T.; Gao, Y. End-to-End Text-to-Image Synthesis with Spatial Constrains. *ACM Trans. Intell. Syst. Technol.* **2020**, *11*, 1–19. [CrossRef]
32. 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5802–5810.
33. Qiao, T.; Zhang, J.; Xu, D.; Tao, D. MirrorGAN: Learning Text-To-Image Generation by Redescription. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
34. Tao, M.; Tang, H.; Wu, S.; Sebe, N.; Jing, X.Y.; Wu, F.; Bao, B. DF-GAN: Deep Fusion Generative Adversarial Networks for Text-to-Image Synthesis. *arXiv* **2020**, arXiv:2008.05865.
35. Li, B.; Qi, X.; Torr, P.; Lukasiewicz, T. Lightweight generative adversarial networks for text-guided image manipulation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 22020–22031.
36. Gatys, L.; Ecker, A.S.; Bethge, M. Image style transfer using convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA, 26 June–1 July 2016; pp. 10501–10510.
37. 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, 18–22 June 2018; pp. 8789–8797.
38. 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.
39. Madaan, A.; Setlur, A.; Parekh, T.; Poczos, B.; Neubig, G.; Yang, Y.; Salakhutdinov, R.; Black, A.W.; Prabhumoye, S. Politeness transfer: A tag and generate approach. *arXiv* **2020**, arXiv:2004.14257.
40. Gatys, L.A.; Ecker, A.S.; Bethge, M. A neural algorithm of artistic style. *arXiv* **2015**, arXiv:1508.06576.
41. 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.
42. Zeng, J.; Ma, X.; Zhou, K. CAAE++: Improved CAAE for age progression/regression. *IEEE Access* **2018**, *6*, 66715–66722. [CrossRef]
43. Liu, S.; Sun, Y.; Zhu, D.; Bao, R.; Wang, W.; Shu, X.; Yan, S. Face aging with contextual generative adversarial nets. In Proceedings of the 25th ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 82–90.
44. Zhai, Z.; Zhai, J. Identity-preserving conditional generative adversarial network. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–5.
45. Sun, Y.; Tang, J.; Shu, X.; Sun, Z.; Tistarelli, M. Facial age synthesis with label distribution-guided generative adversarial network. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2679–2691. [CrossRef]
46. Shi, C.; Zhang, J.; Yao, Y.; Sun, Y.; Rao, H.; Shu, X. CAN-GAN: Conditioned-attention normalized GAN for face age synthesis. *Pattern Recognit. Lett.* **2020**, *138*, 520–526. [CrossRef]
47. An, J.; Huang, S.; Song, Y.; Dou, D.; Liu, W.; Luo, J. ArtFlow: Unbiased image style transfer via reversible neural flows. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 862–871.
48. Liu, S.; Lin, T.; He, D.; Li, F.; Wang, M.; Li, X.; Sun, Z.; Li, Q.; Ding, E. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 6649–6658.
49. Lin, T.; Ma, Z.; Li, F.; He, D.; Li, X.; Ding, E.; Wang, N.; Li, J.; Gao, X. Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 5141–5150.

50. Duan, B.; Wang, W.; Tang, H.; Latapie, H.; Yan, Y. Cascade attention guided residue learning gan for cross-modal translation. *arXiv* **2019**, arXiv:1907.01826.
51. Sun, W.; Wu, T. Learning layout and style reconfigurable gans for controllable image synthesis. *arXiv* **2020**, arXiv:2003.11571.
52. Li, B.; Qi, X.; Lukasiewicz, T.; Torr, P.H. Manigan: Text-guided image manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7880–7889.
53. Kenan, E.; Sun, Y.; Lim, J.H. Learning Cross-Modal Representations for Language-Based Image Manipulation. In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Online, 25–28 October 2020; pp. 1601–1605. [CrossRef]
54. Sylvain, T.; Zhang, P.; Bengio, Y.; Hjelm, R.D.; Sharma, S. Object-centric image generation from layouts. *arXiv* **2020**, arXiv:2003.07449.
55. Turkoglu, M.O.; Thong, W.; Spreeuwers, L.; Kicanaoglu, B. A layer-based sequential framework for scene generation with gans. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8901–8908.
56. El-Nouby, A.; Sharma, S.; Schulz, H.; Hjelm, D.; Asri, L.E.; Kahou, S.E.; Bengio, Y.; Taylor, G.W. Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 10304–10312.
57. Odena, A.; Dumoulin, V.; Olah, C. Deconvolution and Checkerboard Artifacts. *Distill* **2016**, *1*, e3. [CrossRef]
58. Sugawara, Y.; Shiota, S.; Kiya, H. Checkerboard artifacts free convolutional neural networks. *APSIPA Trans. Signal Inf. Process.* **2019**, *8*, e9. [CrossRef]
59. Simonoff, J.S. *Smoothing Methods in Statistics*; Springer Science & Business Media: Berlin, Heidelberg, Germany, 2012.
60. Butterworth, S. On the theory of filter amplifiers. *Wirel. Eng.* **1930**, *7*, 536–541.
61. Mao, X.; Li, Q.; Xie, H.; Lau, R.; Wang, Z.; Smolley, S.P. Least Squares Generative Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2017; pp. 2794–2802.
62. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
63. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
64. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene Parsing Through ADE20K Dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 633–641.
65. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
66. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [CrossRef]
67. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
68. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]
69. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. Gans trained by a two time-scale update rule converge to a local Nash equilibrium. In Proceedings of theAdvances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017.
70. Borji, A. Pros and cons of gan evaluation measures. *Comput. Vis. Image Underst.* **2019**, *179*, 41–65. [CrossRef]
71. Qiao, T.; Zhang, J.; Xu, D.; Tao, D. Learn, imagine and create: Text-to-image generation from prior knowledge. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 887–897.
72. Tan, H.; Liu, X.; Li, X.; Zhang, Y.; Yin, B. Semantics-enhanced adversarial nets for text-to-image synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 10501–10510.