# Text-to-Image Synthesis Based on Machine Generated Captions

Marco Menardi[1], Alex Falcon[1], Saida S. Mohamed[1(✉)], Lorenzo Seidenari[2], Giuseppe Serra[1], Alberto Del Bimbo[2], and Carlo Tasso[1]

[1] Artificial Intelligence Laboratory, University of Udine, Udine, Italy
{menardi.marco,falcon.alex,mahmoud.saidasaadmohamed}@spes.uniud.it,
{giuseppe.serra,carlo.tasso}@uniud.it
[2] Media Integration and Communication Center, University of Firenze, Florence, Italy
{lorenzo.seidenari,alberto.delbimbo}@unifi.it

**Abstract.** Text-to-Image Synthesis refers to the process of automatic generation of a photo-realistic image starting from a given text and is revolutionizing many real-world applications. In order to perform such process it is necessary to exploit datasets containing captioned images, meaning that each image is associated with one (or more) captions describing it. Despite the abundance of uncaptioned images datasets, the number of captioned datasets is limited. To address this issue, in this paper we propose an approach capable of generating images starting from a given text using conditional generative adversarial network (GAN) trained on uncaptioned images dataset. In particular, uncaptioned images are fed to an Image Captioning Module to generate the descriptions. Then, the GAN Module is trained on both the input image and the "machine-generated" caption. To evaluate the results, the performance of our solution is compared with the results obtained by the unconditional GAN. For the experiments, we chose to use the uncaptioned dataset LSUN-bedroom. The results obtained in our study are preliminary but still promising.

**Keywords:** Generative Adversarial Networks (GANs) · StackGAN · Self-Critical Sequence Training (SCST) · Text-to-Image Synthesis

## 1 Introduction

Text-to-Image Synthesis, also called Conditional Image Generation, is a process that consists in generating a photo-realistic image given a textual description. It is a challenging task and it is revolutionizing many real-world applications. For example, starting from a Digital Library of adventure books it could be possible to enrich the reading experience with computer-generated images of the locations explored in the story, while a Digital Library of recipe books may be enriched with images representing the steps involved in a given recipe. In addition, such


M. Ceci et al. (Eds.): IRCDL 2020, CCIS 1177, pp. 62–74, 2020.
https://doi.org/10.1007/978-3-030-39905-4_7

images may be used to exploit Information Retrieval systems based on visual similarity. Due to its great potentiality and usefulness, it raised a lot of interest in the research fields of Computer Vision, Natural Language Processing, and Digital Libraries.

One of the main approaches used for the text-to-image task involves the use of Generative Adversarial Networks (GAN) [6]: starting from a given textual description, GANs can be conditioned on text [24,25,35] in order generate high-quality images that are highly related to the text meaning.

To condition a GAN on text, captioned images datasets are needed, meaning that one (or more) captions must be associated to each image. Despite the large amount of uncaptioned images datasets, the number of captioned datasets is limited. For example, LSUN [33] dataset, which consists in more than 59 million labeled images for each of 10 scene categories and 20 object categories [33]. The LSUN-bedroom dataset contains images from LSUN dataset tagged with the "bedroom" scene category. It contains around ∼3,000,000 images [33], but it does not contain the associated captions. This may lead to a difficulty in training a conditional GAN to generate bedroom images related to a given textual description, such as "a bedroom with blue walls, white furniture and a large bed". In this paper we propose an innovative, though quite simple approach to address this issue as shown in Fig. 1.

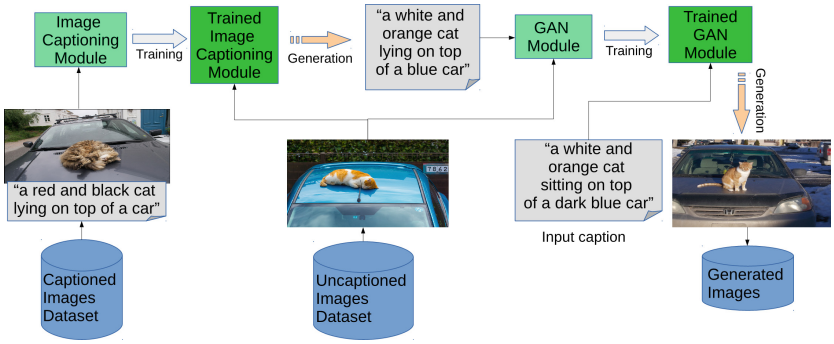


**Fig. 1.** Our pipeline: captioned images are used to train the Image Captioning Module; uncaptioned images are then captioned through the Trained Image Captioning Module and both the image and the generated captions are used to train the GAN Module; finally, the Trained GAN Module is used to generate an image based on an input caption.

First of all, a captioning system (that we call Image Captioning Module) is trained on a generic captioned dataset and used to generate a caption for the uncaptioned images. Then, the conditional GAN (that we call GAN Module) is trained on both the input image and the "machine-generated" caption. A high-level representation of the architecture is shown in Fig. 2. To evaluate the results, the performance of the GAN using "machine-generated" captions are compared

with the results obtained by the unconditional GAN. To test and evaluate our pipeline, we are using the LSUN-bedroom [33] dataset.

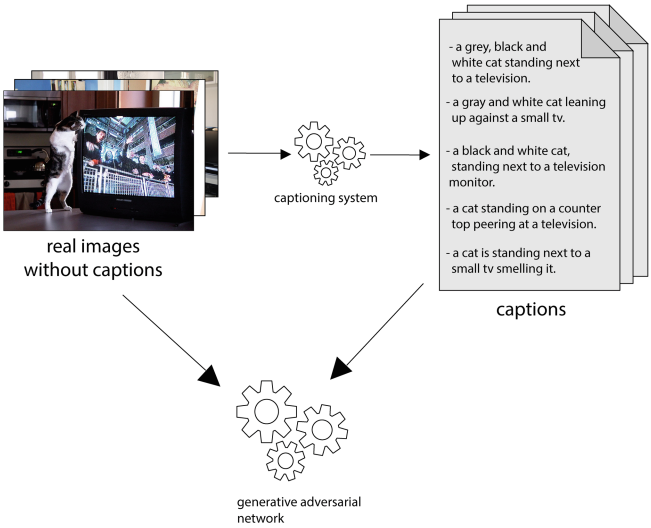


**Fig. 2.** Pipeline: images are fed to a captioning system that outputs its captions. The generated captions and the images are then given as input for training the conditional GAN.

The results obtained in the experiments are very preliminary yet very promising. According to our study, the GAN Module does not learn how to produce meaningful images, with respect to the caption meaning, and we hypothesize that this is due to the "machine-generated" captions we use to condition the GAN Module. The Image Captioning module is trained on the COCO dataset [17], which contains captioned images for many different classes of objects and intuitively this should lead the Image Captioning Module to learn how to produce captions for bedroom images as well. Despite being able to produce the desired captions, we notice that the "machine-generated" captions are often too similar and not detailed for different bedroom images. The last section of the paper proposes some approaches that can deal with these problems.

## 2 Related Work

In 2014, Goodfellow et al. introduced Generative Adversarial Networks (GAN) [6], a generative model framework that consists in training simultaneously two models: a generator network and a discriminator one. The generator network has the task of generating images as real as possible, while the discriminator network has to distinguish the generated images from the real ones. Generative

models are trained to implicitly capture the statistical distribution of training data; once trained, they can synthesize novel data samples, which can be used for example in the tasks of semantic image editing [38] and data augmentation [1].

GANs can be trained to sample from a given data distribution, in such case a random vector is provided as input to the generator. Otherwise, as in the case of text-to-image synthesis, they can be trained conditionally, meaning that an additional variable is provided as input to control the generator output. In certain formulations, the discriminator observes the conditioning variable too, during training. In the literature, several possibilities were tested for the variables used to condition a GAN: attributes or class labels (e.g. [2,20]), images (e.g. for the tasks of photo editing [38] and domain transfer [11]).

Several methods have been developed to generate images conditioned on text. Mansimov et al. [18] built an AlignDRAW model trained to learn the correspondence between text and generated images. Reed et al. in [26] used PixelCNN to generate images using both text descriptions and object location constraints. Nguyen et al. [19] used an approximate Langevin sampling approach to generate images conditioned on text, but it required an inefficient iterative optimization process. In [25], Reed et al. successfully generated $64 \times 64$ images for birds and flowers conditioning on text descriptions. In their follow-up work [24], they were able to generate $128 \times 128$ images by using additional annotations on object part locations. Denton et al. in [5] proposed the Laplacian pyramid framework (LAPGANs), which is composed of a series of GANs. A residual image is conditioned at each level of the pyramid on the image of the previous stage to produce an image for the next stage. Also in [13], Kerras et al. use a similar approach by incrementally adding more layers in the generator and in the discriminator. [34] and [35] suggest the use of a so-called sketch-refinement process, where the images are first generated at low resolutions using a GAN conditioned over the textual description, and then refined with another GAN conditioned on both the image generated at the previous step and the input textual description. [9] and [15] infer a semantic label map by predicting bounding boxes and object shapes from the text, and then synthesize an image conditioned on the layout and the text description. A recent work by Qiao et al. [21] uses a three-step approach where it first computes word- and sentence-level embedding from the given textual description, then it uses the embeddings to generate images in a cascaded architecture, and finally starting from the image generated at the previous step it tries to regenerate the original textual description, in order to semantically align with it. Although several different state-of-the-art architectures may be chosen for the task, such as HDGAN [36] and AttGAN [32], in our pipeline we decided to use StackGAN-v2 [35] as the conditional GAN component, given the availability of its code on GitHub.

Recently, several impressive results [16,27,37] were obtained for the Image Captioning (or image-to-text) task, which deals with the generation of a caption describing the given image and the objects taking part to it. It is an important task that raises a lot of interest in the Computer Vision and Natural Language

Processing research fields. A recent and comprehensive survey about the task is provided by Hossain et al. in [10]. Some of the approaches used for this task involve the use of Encoder/Decoder networks and Reinforcement learning techniques.

The encoder/decoder paradigm for machine translation using recurrent neural networks (RNNs) [3] inspired [12,30] to use a deep convolutional neural network to encode the input image, and a Long Short-Term Memory (LSTM) [8] RNN decoder to generate the output caption. Given the unavailability of labeled data, recent approaches to the image captioning task involve the use of reinforcement learning and unsupervised learning-based techniques. [37] and [16] use actor-critic reinforcement learning methods, where a "policy network" (the actor) is trained to predict the next word based on the current state, whereas a "value network" (the critic) is trained to estimate the reward of each generated word. These techniques overcome the need to sample from the policy (actors) action space, which can be enormous, at the expense of estimating future rewards. Another approach, used by Ranzato et al. in [22], consists in applying the REINFORCE algorithm [31]. A limitation of this algorithm consists in the requirement of a context-dependent normalization to tackle the high variance encountered when using mini-batches. The approach we are following uses Self-Critical Sequence Training (SCST) [27] which is a REINFORCE algorithm that utilizes the output of its own test-time inference algorithm to normalize the rewards it experiences: doing so, it does not need neither to estimate the reward signal nor the normalization.

## 3 Our Approach

We propose a pipeline whose goal is to generate images by conditioning on "machine-generated" captions. This is fundamental when image captions are not available for a specific domain of interest. Thus, the proposed solution involves the use of a generic captioned dataset, such as the COCO dataset, to make the Image Captioning Module capable of generating captions for a specific domain.

To do so, we want to explore the possibility of using an automatic system to generate textual captions for the images and use them for the training of a Generative Adversarial Network. For achieving our goal, we built a pipeline composed by an Image Captioning Module and a GAN Module, as shown in Fig. 1. First of all, the Image Captioning Module is trained over a generic captioned dataset to generate multiple captions for the input image. Then, real images are given as input to the Trained Image Captioning Module, which outputs multiple captions for each image. The generated captions together with the images are then fed to the GAN Module, which learns to generate images conditioned on the "machine-generated" captions. By feeding the GAN with multiple captions for each image, the GAN can better learn the correspondence between images and captions.

In the following sections, we detail the two modules used in our pipeline: the Image Captioning Module and the GAN Module.

### 3.1 Image Captioning Module

The goal of the Image Captioning Module is to generate a natural language description of an image. Good performance in this task are obtained by learning a model which is able to first understand the scene described in the image, the objects taking part to it and the relationships between them, and then to compose a natural language sentence describing the whole picture. Given the complexity of such a task, it is still an open challenge in the fields of Natural Language Processing and Computer Vision. The task of open domain captioning is a challenging task. It requires a fine-grained understanding of the whole entities, attributes and relationships in an image. In our pipeline, we are implementing our Image Captioning Module in a similar way as the one proposed in [27], meaning that we also use a captioning system based on FC models. It has been built using an optimization approach that is called Self-Critical Sequence Training (SCST).

Typical deep learning models used for the Image Captioning task are trained with the "teacher-forcing" technique, which consists in maximizing the likelihood of the next ground-truth word given the previous ground-truth word. This has been shown to generate some mismatches between the training and the inference phase, knows as "exposure bias". Moreover, the metrics used during the testing phase are non-differentiable (such as BLEU and CIDEr), meaning that the captioning model can not be trained to directly optimize them. To overcome these problems, Reinforcement Learning techniques such as the REINFORCE algorithm have been used. SCST is a variation and an improvement of the popular REINFORCE algorithm that, rather than estimating a baseline to normalize the rewards and reduce variance, utilizes the output of its own test-time inference algorithm to normalize the rewards it experiences. This means that it is forced to improve the performance of the model under the inference algorithm used at test time. Practically, SCST has much lower variance than REINFORCE and can be more effectively trained on mini-batches of samples using SGD. Moreover, it has been shown that SCST system has achieved state-of-the-art performance by optimizing their system using the test metrics of the MSCOCO task. Practically, it has been found that SCST has much lower variance, and can be more effectively trained on mini-batches of samples using SGD. Since the SCST baseline is based on the test-time estimate under the current model, SCST is forced to improve the performance of the model under the inference algorithm used at test time. In addition, this encourages training consistency like the maximum likelihood-based approaches except it optimized sequence metrics.

### 3.2 GAN Module

The GAN Module has the major role of learning to generate images by conditioning on the "machine-generated" captions. In particular, we are using StackGAN-v2 [35] as our GAN Module.

StackGAN-v2 consists of a multiple stage generation process, where high-resolution images are obtained by initially generating low-resolution images

which are then refined in multiple steps. It consists in a single end-to-end network composed by multiple generators and discriminators in a tree-like structure. Different branches of the tree generate images of different resolutions: at branch $i$, the generator $G_i$ learns the image distribution $p_{G_i}$ at that scale, while the discriminator $D_i$ estimates the probability of a sample being real. The framework of StackGAN-v2 has a tree-like structure, that takes as input the noise vector $z \sim p_{noise}$. The noise z is transformed in hidden feature layer by layer. The hidden features $h_i$ for each generator $G_i$ are calculated by a non-linear transformation

$$h_0 = F_0(z); \quad h_i = F_i(h_{i-1}, z), \tag{1}$$

where $h_i$ represents hidden features for the $i^{th}$ branch, $m$ is the total number of branches, and $F_i$ are modeled as neural networks. The noise vector $z$ is concatenated to the hidden features $h_{i-1}$ as the inputs of $F_i$ for calculating $h_i$. The generators produce samples at different scales $(s_0, s1, ..., s_{m-1})$ based on the hidden features at different layers $(h_0, h_1, ..., h_{m-1})$.

$$s_i = G_i(h_i), \quad i = 0, 1, ..., m-1, \tag{2}$$

where $G_i$ is the generator for the $i^{th}$ branch. Since we are more interested in the conditional case, we are not reporting the loss function used by the generator and the discriminator in the unconditional setting, for which more details can be found in [35].
The discriminator $D_i$ takes a real image $x_i$ or a fake sample $s_i$ as input and is trained to classify them as real or fake by minimizing the cross entropy loss:

$$\begin{aligned} \mathcal{L}_{D_i} = & \underbrace{-\mathbb{E}_{x_i \sim p_{data_i}}[logD_i(x_i)] - \mathbb{E}_{x_i \sim p_{G_i}}[log(1 - D_i(s_i))]}_{\text{unconditional loss}} \\ & \underbrace{-\mathbb{E}_{x_i \sim p_{data_i}}[logD_i(x_i, c)] - \mathbb{E}_{x_i \sim p_{G_i}}[log(1 - D_i(s_i, c))]}_{\text{conditional loss}} \end{aligned} \tag{3}$$

where $x_i$ is an image from the true image distribution $p_{data_i}$ at the $i^{th}$ scale, $s_i$ is from the model distribution $p_{G_i}$ at the same scale. While StackGAN-v2 [35] follows the approach of Reed et al. [23] to pre-train a text encoder to extract visually-discriminative text embeddings of the given description, in our case we use Skip-Thought [14], that works at the sentence level, to generate the text embeddings ($c$ in the Eqs. 3 and 4). Sentences that share semantic and syntactic properties are mapped to corresponding vector representations [14].

The multiple discriminators are trained in parallel each one for a different scale, while the generator is instead optimized to jointly approximate multi-scale image distributions $p_{data_0}, p_{data_1}, ..., p_{data_{m-1}}$ by minimizing the following loss function:

$$\mathcal{L}_G = \sum_{i=1}^{m} \mathcal{L}_{G_i}, \quad \mathcal{L}_{G_i} = \underbrace{-\mathbb{E}_{s_i \sim p_{G_i}}[logD_i(s_i)]}_{\text{unconditional loss}} \underbrace{-\mathbb{E}_{s_i \sim p_{G_i}}[logD_i(s_i, c)]}_{\text{conditional loss}} \tag{4}$$

where $L_{G_i}$ is the loss function for approximating the image distribution at the $i^{th}$ scale. The unconditional loss is used to determine whether the image is real or fake, while the conditional loss is used to determine if the image and the condition match.

## 4 Experimental Results

In this section, we present the preliminary results of the experiments involving the proposed pipeline. The Image Captioning Module was trained on the COCO dataset [17], which contains 120,000 generic images tagged with categories and captioned by five different sentences each.

The uncaptioned dataset that we considered is the LSUN [33] dataset, which consists in more than 59 million labeled images. From the LSUN dataset, we first select the ∼3,000,000 images tagged with the "bedroom" scene category and from that set a subset of the first 120,000 images is selected: 80,000 are then used to train the GAN and 40,000 as test set. Later on in this paper, the selection of the ∼3,000,000 images tagged with the "bedroom" scene category is called "LSUN-bedroom".

A typical metric used to evaluate both the quality and the diversity of generated images is the Inception Score [28]. Unfortunately, the type of image of the LSUN dataset is very different from those used by ImageNet [4,35], therefore it has been shown that the Inception Score is not a good indicator for the quality of generated images [35]. So we decided not to report the obtained scores.

We performed three experiments over the considered dataset.

The first experiment consists in training the GAN Module on the whole LSUN-bedroom dataset (∼3,000,000 images). This is done because of two reasons: first, it serves as a baseline for the next experiment; second, we compare the results obtained by our computing facilities with the results obtained in [35], since with our graphics card we are limited to a lower batch size of 16. Figure 4 shows some examples of generated images, and it is possible to see that the quality of the generated images is similar to those shown in Fig. 3 [35].

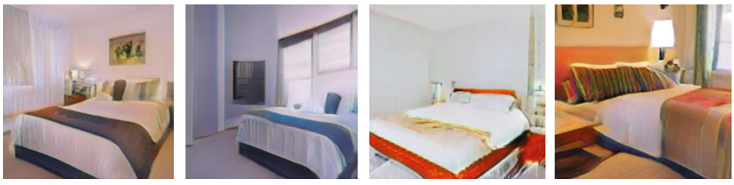

**Fig. 3.** Examples of images generated by the StackGAN Module trained on the whole LSUN-bedroom dataset.

To reproduce the results reported in the paper, we used an NVIDIA GTX 1080 8 GB machine. It took us around one month to train the GAN Module on

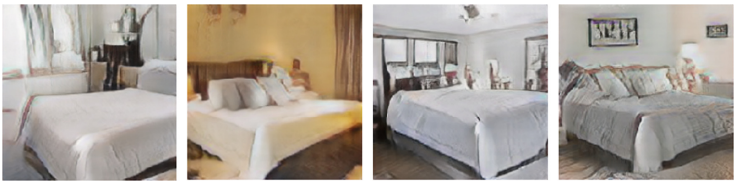

**Fig. 4.** Examples of images generated by the GAN Module trained on the whole LSUN-bedroom dataset.

the whole LSUN dataset. Because of this, we decided to explore and understand how the GAN Module performs with less training images. In the second experiment, the training of the GAN Module without conditioning is done on a subset of LSUN-bedroom, consisting of 120,000 images. Some of the results obtained in this experiment are showed in Fig. 5. Although the quality of the generated images is slightly reduced, it is possible to see that the semantic content is still clear and defined.

**Fig. 5.** Examples of images generated by the GAN Module trained on a part of the LSUN-bedroom dataset.

Finally, to test our pipeline, we used the Image Captioning Module to generate captions for the images contained in the considered subset of the LSUN-bedroom dataset. Then, the GAN Module was trained on these same images and conditioned by the "machine-generated" captions. About the preliminary results that we obtained, some examples are shown in Fig. 6. We suspect the problem is due to the similarity of the "machine-generated" captions: the LSUN-bedroom dataset does not come with captions and thus the Image Captioning Module is trained on a generic dataset (COCO) and not for that specific dataset. Because of this, the Image Captioning Module is unable to produce detailed and varied captions for different bedroom images. In addition, Usually GANs used noise vector to generate images which always different from each other [6]. In our experiment, the noise vector is taken as input by the model and used to generate an image. Then, the captions are used to yield the embeddings, which are also used as noise by the generator. The fact that the noise is almost always the same could be the cause of the observed problem.

We found that the scores for the LSUN-bedroom dataset seem to not fully correlate with the quality of the generated images. As explained in [35], this may


**Fig. 6.** Examples of images generated by the GAN Module trained on a part of the LSUN-bedroom dataset and conditioned on “machine-generated” captions.

be due to the inception score being trained on the inception dataset, and thus it does not work well on datasets with specific types of images. Also, it has to be considered that different datasets get inception scores in different ranges. For this reason, inception scores must not be compared across different datasets.

## 5 Conclusion

We explored the problem of conditional image generation using Generative Adversarial Networks with machine-generated captions. For this task, we built a pipeline to first generate captions for uncaptioned datasets and then to use the “machine-generated” captions to condition a GAN. To test our pipeline, we run experiments on the LSUN-bedroom dataset, which is a subset of the LSUN dataset containing uncaptioned images of bedrooms, and then compare the generated images in the unconditional setting and in the conditional setting where “machine-generated” captions are used. The results observed in the experiments do not achieve success in the task of conditioning with “machine-generated” captions. So we identify, analyze, and propose possible solutions to the obstacles that need to be overcome.

The Image Captioning Module that we trained on the COCO dataset seems to generate captions too similar to each other. Moreover, The captions we generated lack details and contain some errors. This is probably related to the fact that more diverse and detailed captions are needed during training in order to achieve significant improvements. During a subsequent review of works on captioning, we found a work from Shetty et al. [29], that promises to generate

more different captions, instead of variations of the same caption. This result is achieved by using GANs for image captioning instead of other traditional methods. An open question is whether with a bigger dataset the GAN could learn the image-captions correspondence, even when captions are very similar for each image. We believe improving the quality of the generated caption is the main challenge for our method. An hybrid approach could make our proposed method work by making humans write captions on a subset of the dataset, then use the obtained captions to train a captioning system. For generating human captions, crowdsourcing platforms like Amazon Mechanical Turk (AMT) could be used. We are currently working on this idea because it's likely that it will lead to improvements in the quality of generated bedroom images, given that AMT could make it possible to have high-quality and more diverse captions. Moreover, we are also considering the use of the Fréchet Inception distance [7] to evaluate the generated captions and images.

## References

1. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 95–104 (2016)
2. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. Adv. Neural Inf. Process. Syst. **29**(2016), 2172–2180 (2016)
3. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, pp. 1724–1734 (2014)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
5. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a Laplacian pyramid of adversarial networks. In: Advances in Neural Information Processing Systems, pp. 1486–1494 (2015)
6. Goodfellow, I., et al.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 27, pp. 2672–2680. Curran Associates, Inc. (2014)
7. 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-supplementary material
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Hong, S., Yang, D., Choi, J., Lee, H.: Inferring semantic layout for hierarchical text-to-image synthesis. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7986–7994 (2018)
10. Hossain, M., Sohel, F., Shiratuddin, M.F., Laga, H.: A comprehensive survey of deep learning for image captioning. ACM Comput. Surv. (CSUR) **51**(6), 118 (2019)
11. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp. 5967–5976 (2017)

12. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions, pp. 3128–3137 (2015)
13. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: 6th International Conference on Learning Representations, ICLR 2018 (2018)
14. Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R.S., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. In: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, Quebec, Canada, 7–12 December 2015, pp. 3294–3302 (2015). http://papers.nips.cc/paper/5950-skip-thought-vectors
15. Li, W., et al.: Object-driven text-to-image synthesis via adversarial training. CoRR abs/1902.10740 (2019), http://arxiv.org/abs/1902.10740
16. Zhang, L., Sung, F., Liu, F., Xiang, T., Gong, S., Yang, Y., Hospedales, T.M.: Actor-critic sequence training for image captioning (2017)
17. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
18. Mansimov, E., Parisotto, E., Ba, L.J., Salakhutdinov, R.: Generating images from captions with attention. In: 4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings, San Juan, Puerto Rico, 2–4 May 2016 (2016). http://arxiv.org/abs/1511.02793
19. Nguyen, A., Clune, J., Bengio, Y., Dosovitskiy, A., Yosinski, J.: Plug & play generative networks: conditional iterative generation of images in latent space. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp. 3510–3520 (2017)
20. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier GANs. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 2642–2651. JMLR. org (2017)
21. Qiao, T., Zhang, J., Xu, D., Tao, D.: MirrorGAN: learning text-to-image generation by redescription. CoRR abs/1903.05854 (2019). http://arxiv.org/abs/1903.05854
22. Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks. In: 4th International Conference on Learning Representations, ICLR 2016 (2016)
23. Reed, S.E., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016, pp. 49–58 (2016). https://doi.org/10.1109/CVPR.2016.13
24. Reed, S.E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H.: Learning what and where to draw. Adv. Neural Inf. Process. Syst. **29**(2016), 217–225 (2016)
25. Reed, S.E., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, pp. 1060–1069 (2016)
26. Reed, S.E., Oord, A.v.d., Kalchbrenner, N., Bapst, V., Botvinick, M.M., Freitas, N.d.: Generating Interpretable Images with Controllable Structure (2017). https://www.semanticscholar.org/paper/Generating-Interpretable-Images-with-Controllable-Reed-Oord/bd5c69fd9b34f481e03363f9d913c31af83547a5
27. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp. 1179–1195 (2017)

28. Salimans, T., Goodfellow, I.J., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. Adv. Neural Inf. Process. Syst. **29**(2016), 2226–2234 (2016)
29. Shetty, R., Rohrbach, M., Hendricks, L.A., Fritz, M., Schiele, B.: Speaking the same language: matching machine to human captions by adversarial training. In: IEEE International Conference on Computer Vision, ICCV 2017, pp. 4155–4164 (2017)
30. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, pp. 3156–3164 (2015)
31. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**(3–4), 229–256 (1992). http://link.springer.com/10.1007/BF00992696
32. Xu, T., et al.: AttnGAN: fine-grained text to image generation with attentional generative adversarial networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, pp. 1316–1324 (2018)
33. Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J.: LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. CoRR abs/1506.0 (2015). http://arxiv.org/abs/1506.03365
34. Zhang, H., et al.: StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
35. Zhang, H., et al.: StackGAN++: realistic image synthesis with stacked generative adversarial networks. IEEE Trans. Pattern Anal. Mach. Intell. **41**, 1947–1962 (2018)
36. Zhang, Z., Xie, Y., Yang, L.: Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, pp. 6199–6208 (2018)
37. Zhou, R., Xiaoyu, W., Ning, Z., Xutao, L., Li-Jia, L.: Deep reinforcement learning-based image captioning with embedding reward. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp. 1151–1159 (2017)
38. Zhu, J.-Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 597–613. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_36