Efficient Neural Architecture for Text-to-Image Synthesis

Douglas M. Souza, Jônatas Wehrmann, Duncan D. Ruiz
School of Technology
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, Brazil
{douglas.souza90,jonatas.wehrmann}@edu.pucrs.br, duncan.ruiz@pucrs.br

***Abstract*—Text-to-image synthesis is the task of generating images from text descriptions. Image generation, by itself, is a challenging task. When we combine image generation and text, we bring complexity to a new level: we need to combine data from two different modalities. Most of recent works in text-to-image synthesis follow a similar approach when it comes to neural architectures. Due to aforementioned difficulties, plus the inherent difficulty of training GANs at high resolutions, most methods have adopted a multi-stage training strategy. In this paper we shift the architectural paradigm currently used in text-to-image methods and show that an effective neural architecture can achieve state-of-the-art performance using a single stage training with a single generator and a single discriminator. We do so by applying deep residual networks along with a novel sentence interpolation strategy that enables learning a smooth conditional space. Finally, our work points a new direction for text-to-image research, which has not experimented with novel neural architectures recently.**

***Index Terms*—text-to-image synthesis, generative models, multimodal learning.**

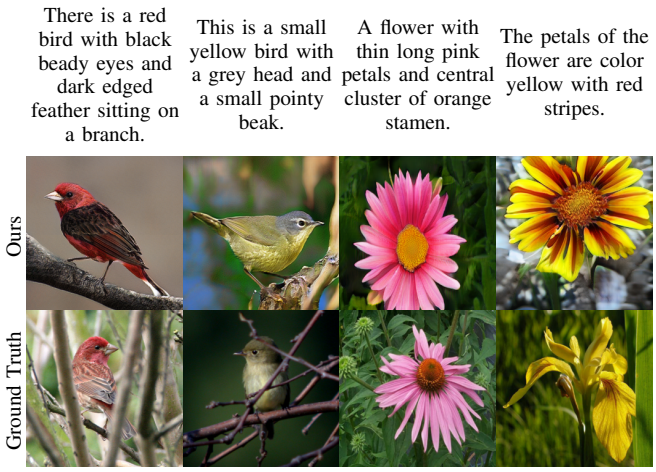


Fig. 1. Images generated by our method.

I. Introduction

Text-to-image synthesis is the task of generating images from text descriptions. Image generation, by itself, is a challenging task. When we combine image generation and text, we bring complexity to a new level: we need to combine data from two different modalities. In the most common setting, text-to-image methods are based on generative models that learn a text-conditioned distribution over images. Given a text description and some random variable, the algorithm produces a random image (controlled by the random variable) that correlates with the information present in the text. Text-to-image synthesis is a very recent research area and it has the potential to aid several real-world applications, from automated content generation to assisted drawing.

Most recent advances in text-to-image generation are driven by Generative Adversarial Networks (GANs) [1]. GANs brought a leap of improvement in learning generative models over complex data distributions such as images. GANs have been successful in several tasks, such as image-to-image translation [2]–[5], image inpainting [6], image editing [7], and image super resolution [8]. In the context of text-to-image synthesis, a conditional GAN [9] is conditioned to a vector representation of the text description. In order to encode text to a vector representation, most methods rely on another algorithm, such as a Recurrent Neural Network (RNN) [10], [11]. There are two main levels in which a text description can be encoded to be used by GANs: sentence level and word level. At the sentence level, the entire description is encoded as global sentence vector. At the word level, on the other hand, there is a vector representation for each individual word in the description.

First approaches to text-to-image synthesis [12], [12]–[15] have simply extended GANs to be conditioned to sentence vectors. Naturally, results were not optimal. Most recent methods [16]–[19] have proposed different strategies to circumvent the complex relationship between image and text. Most of those works, however, follow a similar pattern when it comes to neural architectures. Due to previously mentioned difficulties, plus the inherent difficulty of training GANs at high resolutions, most recent works have adopted a multi-stage training strategy. In a multi-stage setting, training is performed first at low resolutions (*i.e.* 32×32 and 64×64 pixels) and then refined to higher resolutions (128×128 and 256×256 pixels). Usually, multi-stage training is implemented using several generators and several discriminators, which makes training complex and slow. This architectural choice has been followed by most pre-

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior – Brasil (CAPES) – Finance Code 001.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

vious work, which have been adding small improvements, such as word-level features through Attention Mechanisms [16], Memory Networks [19], Siamese Networks [17] and a Mirror strategy [17].

In this paper, we shift the architectural paradigm currently used in text-to-image methods and show that an effective neural architecture can achieve state-of-the-art performance using a single stage training directly at the target resolution. By doing so, we not only introduce a simpler method for text-to-image synthesis but also point a new direction in text-to-image research, which has not experimented with novel neural architecture recently.

Specifically, we introduce an adversarial training-based architecture that leverages full capacity of modern deep convolutional networks, alongside to an improved sentence embedding approach for generating photorealistic text-conditioned images. Both discriminator and generator networks draw inspiration from [20], though we provide important improvements on that architecture, allowing for the use of sentence embeddings rather than class labels as conditioning vectors. Results show that our models single-handedly outperform multi-stage state-of-the-art methods without heavy hyper-parameter optimization in two widely used benchmarks, namely CUB [21] and Oxford-102 [22] datasets, in terms of both Inception Score [23] and Fréchet Inception Distance [24]. Fig. 1 shows samples generated by our method. Moreover, we provide an extensive set of experiments, in which we explore key components and abilities of our models.

Formally, in this work we make the following contributions:

- We introduce a novel sentence interpolation strategy that allows the generator to learn a smooth conditional space, and also work as a data augmentation procedure.
- We show how the use of a modern residual neural architecture enables single-stage training at the target image size, and generates state-of-the-art text-to-image models.
- We perform an extensive analysis of the properties of text-to-image models, both in quantitative and qualitative fashion.
- We demonstrate that our models enable image editing using natural language via arithmetic operations in the conditional space, being able to modify aspects of the image while keeping its overall structure.

II. Related Work

In this section we discuss the most important topics related to our work, namely, Generative Adversarial Networks (GANs), and specific methods designed to perform text-to-image synthesis.

A. Generative Adversarial Networks

Generative Adversarial Networks (GANs) [1] is a class of generative method that learns generative models via an adversarial training procedure. In its traditional form, GANs are composed of two differentiable functions (*e.g.*, neural networks), namely a Generator G and a Discriminator D. D is trained to correctly classify real and generated images while G is trained simultaneously to make D mistakenly classify generated images as real.

Since its debut in 2014, there has been remarkable advances in GAN research. Important methods were proposed to address training stability and quality of results [20], [25]–[31] and several tasks were improved by adversarial training, such as image-to-image translation [2]–[5], image inpainting [6], image editing [7], and image super resolution [8].

B. GANs for Text-to-image Synthesis

Conditioning image generation of GANs on text descriptions was first proposed by Reed *et al.* [12] where the task was defined as two subtasks: encoding text descriptions to a vector representation and using this representation as a condition to train a Conditional GAN [9]. In [32], Reed *et al.*extends text-to-image to support location in which elements should be drawn.

Zhang *et al.* [13] proposed StackGAN, which introduced important concepts that are still used by recent works. StackGAN used stacked GANs to train text-to-image models in a two stage fashion: the first stage generates low resolution 64×64 pixel images then a second stage refines to a higher resolution of 256×256 pixels. StackGAN also introduced the Conditioning Augmentation (CA) module. CA maps text embeddings to a smooth known distribution that makes it easier to learn the text-to-image generator. Stackgan++ [14] extends StackGAN by adding multiples generators and discriminators, a pair is used at each of the following stages: 64×64, 128×128 and 256×256 pixels. HDGAN [15] follows similar multi-stage strategy but applies a patchwise adversarial loss.

Since previous works had only used the global sentence embedding as condition to train text-to-image models, AttnGAN [16] introduced Attention [33] modules to add word-level cues so that generated images are closely related to the description. MirrorGAN [17] learns text-to-image models by redescription, *i.e.* regenerating a text description for a generated image. DM-GAN [19] uses a dynamic memory module to select important aspects of first-stage images and refine to higher resolutions that are closely related to text descriptions. Finally, SD-GAN [18] proposes a siamese multi-stage networks structure that is intended to make generated images consistent across a variety of descriptions.

Our proposed method departs from most of the previously established strategies. We dramatically simplify the text-to-image framework. First, we shift the architectural paradigm from a multi-stage architecture to a single stage modern deep residual network, which makes training simpler and faster. Second, we introduce a sentence interpolation strategy that allows the generator to learn a smooth conditional space, which not only improve results but also allow us to perform image editing by performing arithmetic operations in conditional space. Finally, we demonstrate that our method outperforms previous works that are also conditioned only on the sentence vector.

III. METHOD

In this section we present in details our proposed approach. Text-to-image synthesis methods have followed a similar design pattern regarding neural architectures: they make use of multi-stage training using several networks. This choice, however, increases training complexity and computational costs required to train such models. Our approach departs from this design altogether. We present evidence that the use of an adequate neural architecture plus a simple sentence interpolation strategy can produce state-of-the-art results. In addition, our method peforms a single-stage training with a single generator and a single discriminator. Next, we detail every component of our proposed method: the text encoder, the sentence interpolation strategy and the neural architecture.

A. Text Encoder

We encode text descriptions into a vector representations by using a pre-trained Deep Attentional Multimodal Similarity Model (DAMSM) [16]. The DAMSM module, similarly to [34]–[38], learns image and text encoding functions, namely $\varphi(I)$ and $\phi(S)$, that map images I and textual descriptions S into the same semantic multimodal space. Such a space is trained so that correlated image-caption pairs lie close to each other, while non-correlated pairs must present larger distance than the correlated ones. By optimizing that space, the learned text representation is forced to closely resemble the content from images, and therefore can be as a condition vector $\boldsymbol{s} \in \mathbb{R}^{256}$ in our architecture.

Original image captions S are tokenized, and each token is represented by a specific vector $\mathbb{R}^{300}$. Those vectors feed a Bidirectional GRU network, which provides per-token hidden representations, as well as a final global vector. Hidden representations are used for learning fine-grained correlations with the spatial information of the images, while the global vector contains holistic high-level information of the caption. In this study, we use the global vector alone as textual condition vector $\boldsymbol{s}$, hence $\phi(S) = \boldsymbol{s}$.

B. Sentence Interpolation

In this section we detail a novel strategy for improving the smoothness of the conditional space, which we hereby call Sentence Interpolation (SI). This technique consists in using all the available captions for computing the general sentence embedding regarding an image during training. By doing so, we make the textual representation vector to be continuous in the projected space, rather than being discrete points in the manifold, as a traditional approach would generate.

Formally, let I_i be the i^{th} image from the training dataset, and $S_{ij} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_n\}$ be the set of n correlated sentence embeddings that describe that particular image. We sample an n-sized vector of weights $\mathbf{m} \sim \mathcal{U}(0, 1)$, and further normalize it with a softmax function. Those normalized values are used to weight each one of the sentence vectors, so their sum consists in an interpolated representation of the original sentences. Therefore, the vector $\dot{\boldsymbol{s}}$ that represents the interpolated textual embedding of a given image is calculated as follows:

$$\dot{\boldsymbol{s}} = \sum_{j=1}^{n} \left[S_j \times \left(\frac{e^{\mathbf{m}}}{\sum_{k=1}^{n} e^{\mathbf{m}_k}} \right)_j \right] \tag{1}$$

Such an approach makes a limited set of sentences to be represented by countless continuous points during the training process. The main implications of this technique are two-fold: (i) it makes the sentence embedding space to be more smooth; (ii) and also works as a data augmentation strategy, given that the same textual descriptions can assume different forms depending on the sampling of $\mathbf{m}$. In comparison to the Conditioning Augmentation (CA) module introduced by StackGAN [13], the sentence interpolation has the advantage of being deterministic. This is due to the fact that it is not used during the test phase. CA, on the other hand, introduces randomness when encoding sentence vectors during training and testing.

C. Architecture

We follow the steps of Brock *et al.* [20], which introduced the state-of-the-art architecture for GANs, namely BigGAN-Deep. This architecture is based on residual blocks with bottleneck structure of He *et al.* [39], which makes deeper networks more computationally efficient and easier to train. Also, like SAGAN [30], BigGAN-Deep applies Spectral Normalization [29] and Non-local Blocks [40] in both generator and discriminator. Finally, BigGAN-Deep introduces conditioning information in the generator using Conditional Batch Normalization [41] and in the discriminator using the projection approach of Miyato *et al.* [42].

BigGAN-Deep presented a new state-of-the-art result in the ImageNet [43] dataset in the supervised setting. Therefore, it was designed to be conditioned on class labels. Since in this architecture class labels are represented by dense embeddings, we extended it to handle the sentence vector. Specifically, we replaced the trainable class embeddings by the fixed sentence vectors $\boldsymbol{s}$. In the discriminator, sentence vectors are linearly projected to be used in the projection conditioning. In the generator, sentence vectors are concatenated with the noise vector $\boldsymbol{z}$ and then linearly projected to form BatchNorm gains and biases, gains are one-centered while biases are zero-centered. By using the fixed sentence vectors, the generator and discriminator are forced to adapt to the conditional space learned by the DAMSM encoder, which yields interesting properties, such as the generator's ability to handle arithmetic operations in conditional space, which is presented in Section VI.

The BigGAN-Deep architecture was originally designed to be used in large scale training. Large scale training is done by using a big batch size (*e.g.* 2048) and training the models across several devices. In order to apply this architecture in a small scale, we need to make additional adaptations. First, we switch relu activation to leaky relu. This helps avoiding sparse gradients, which is helpful due to the second adaptation. Second, we reduce the number of parameters of both networks. We reduce the number of parameters in the generator and

discriminator by reducing the channel multiplier ch to 96 instead of 128 in default BigGAN-Deep architecture. This reduction represents 43% less parameters in the discriminator and 36% less parameters in the generator. Finally, training is performed directly at the target resolution of 256×256 pixels. As far as we know, no previous text-to-image method was able to train directly at this resolution without relying on multiples generators and discriminators.

D. Objective Function

We adopt the so-called hinge adversarial loss. The hinge loss works similar to WGAN loss [25] but is more stable thanks to the margins introduced in the discriminator loss function. For the discriminator, the hinge loss is given by:

$$V_D(\hat{G}, D) = \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{s}\sim q_{\text{data}}} \left[\min\left(0, -1 + D(\boldsymbol{x}, \boldsymbol{s})\right)\right] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\boldsymbol{z}},\boldsymbol{s}\sim q_{\text{data}}} \left[\min\left(0, -1 - D\left(\hat{G}(\boldsymbol{z}, \boldsymbol{s}), \boldsymbol{s}\right)\right)\right], \tag{2}$$

where $\boldsymbol{x}$ and $\boldsymbol{s}$ are real images and their corresponding sentence vectors, respectively. $\hat{G}(\boldsymbol{z}, \boldsymbol{s})$ is a fake image from the generator for a given random vector $\boldsymbol{z}$ and a sentence vector $\boldsymbol{s}$, respectively. Note that the hat in G means that, in this case, the generator's weights are not being updated.

Similarly, the loss function for the generator is given by:

$$V_G(G, \hat{D}) = - \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\boldsymbol{z}},\boldsymbol{s}\sim q_{\text{data}}} \left[\hat{D}\left(G(\boldsymbol{z}, \boldsymbol{s}), \boldsymbol{s}\right)\right], \tag{3}$$

in this case, the hat in D means the discriminator's weights are not being updated.

IV. Experiments

In this section we present our experimental setup. We conduct extensive experiments in the most used datasets for text-to-image generation. We also present an extensive quantitative and qualitative analysis of our findings.

A. Datasets

We have used two widely used datasets for training and evaluating our models, as follows.

Caltech-UCSD Birds (CUB) [21]: The CUB dataset is composed of 11,788 images of birds distributed among 200 class categories. The dataset is split in 8,855 images of 150 categories for training and 2,933 images of 50 categories for testing. Each image contains 10 text descriptions.

Oxford-102 [22]: The Oxford-102 dataset is composed of 8189 images of flowers of 102 categories. The dataset is split in 7034 images for training and 1154 images for testing. Each image contains 10 text descriptions.

Fig. 2. Image generation based on condition space arithmetic of embedded textual descriptions.

B. Evaluation

In order to evaluate our method, we employ the two most widely used metrics to evaluate generative models: the Inception Score (IS) and the Fréchet Inception Distance (FID). The IS uses a pretrained Inception Network [44] to compute class probabilities over generated samples. IS is both a measure of *objectness* and variety, therefore, the higher the score the better. In order to compute IS, and also be able to compare results, we use the same Inception Networks used to evaluate previous work. The networks are provided by StackGAN [13] and are finetuned for the CUB and Oxford-102 datasets.

A downside of the IS is that it does not consider the statistics present in the real data. A generative model that generates a few high quality examples for each class would have a very high IS score, despite its variety being low. To circumvent this issue, Heusel *et al.* [24] introduced the Fréchet Inception Distance (FID). FID considers the statistics present in the training data, so it possibile to evaluate if the generative model learned a distribution that have similar statistics. Specifically, FID uses an Inception Network to compute activation features of both training set images and generated images. The Fréchet Distance is then computed over the features of real and fake images. FID gives a measure of how close the statistics of generated images are from those in the training set, hence, the lower the score the better.

C. Implementation Details

We use Adam optimizer [46] with a learning rate of 4×10^{-4} for D and 10^{-4} for G. We set $\beta_1 = 0$ and $\beta_2 = 0.999$ for both G and D. We train one D step per G step. We use synchronized implementation of BatchNorm, where statistics

TABLE I
QUANTITATIVE COMPARISON OF TEXT-TO-IMAGE METHODS.

Method	# Networks		Multi-Stage	IS↑		FID↓	
	Discriminators	Generators		CUB	Oxford-102	CUB	Oxford-102
GAN-INT-CLS [12]	1	1	No	2.88 ± 0.04	2.66 ± 0.03	-	-
GAWWN [32]	1	1	No	3.60 ± 0.07	-	-	-
StackGAN [13]	2	2	Yes	3.70 ± 0.04	3.20 ± 0.01	55.28	51.89
StackGAN++ [14]	3	3	Yes	4.04 ± 0.05	3.26 ± 0.01	15.30	48.68
TAC-GAN [45]	1	1	No	-	3.45 ± 0.05	-	-
HDGAN [15]	3	3	Yes	4.15 ± 0.05	3.45 ± 0.07	-	-
Ours	1	1	No	**4.23 ± 0.05**	**3.71± 0.06**	**11.17**	**16.47**

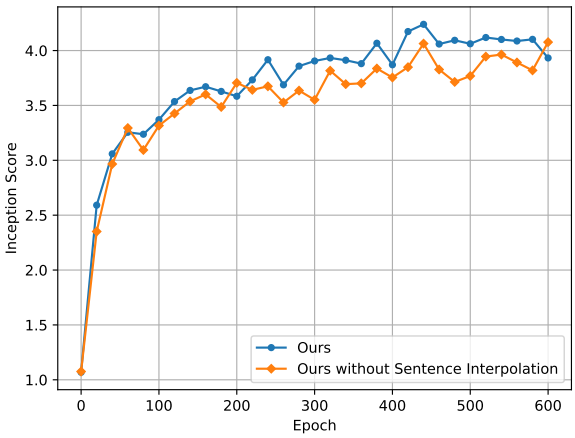


Fig. 3. Inception Score during training epochs for our model with and without Sentence Interpolation in the CUB dataset.

are aggregated across all devices. We keep an exponential moving average of the generator weights with a decay of 0.999 for sampling. Since BatchNorm statistics are not computed for averaged weights of the generator, we employ the "standing statistics" strategy of Brock *et al.* [20]. In other words, we first run 100 forward passes through G to update its BatchNorm statistics, making the generator invariant to batch sizes. Finally, we perform training using 3 GPUs with a batch size of 8 per GPU, making up for a batch of size 24. Most models take up to 3 days to train.

V. COMPARISON TO STATE-OF-THE-ART METHODS

In order to provide reassurance on the generative performance of our models, we compare their quantitative and qualitative results against current state-of-the-art methods [12]–[15], [45]. Note that some of them have not reported FID results. Hence, we compare to the results publicly available.

A. Quantitative Analysis

Table I depicts quantitative results, alongside to the number of discriminator and generator networks used in each work. It arguably shows that our approach is the preferred method, once it achieves top performance in all metrics while employing just a single discriminator and a single generator in the entire architecture. Notably, it outperforms all the baseline approaches by a margin across all datasets and metrics.

The largest improvement provided by our approach is on Oxford-102 dataset. It provides a relative improvement of $\approx 7\%$ IS and $\approx 300\%$ FID when compared to the strongest baseline with public results available. Clearly our approach also leads to a significantly better results on CUB dataset, allowing for a $\approx 24\%$ FID reduction.

B. Qualitative Analysis

Fig. 4 depicts qualitative results of models trained on CUB dataset. In that Fig., we compare our model to the baseline ones. One can observe that our model brings improvement on several aspects regarding the generated images. For instance, our images look more photorealistic, present better semantic correspondence of the generated images to the provided description, and also generate more fine-grained details in both foreground and background elements.

Results shown in Fig. 5 were generated using a model trained on Oxford-102 dataset. Once again, our model generates images with much richer detail level and photorealistic aesthetic. Such experiment supports our claims that our proposed single-stage architecture can be used for generating concepts across distinct datasets. It is worth noticing that despite Oxford-102 being a somewhat small dataset, our models were able to learn a proper distribution without suffering from mode collapse or additional training instabilities.

VI. CONDITION SPACE ARITHMETIC

In this section we explore the inherent capability of our approach to handle condition space arithmetic. This is a very interesting property and finds applications in many real world tasks, such as image manipulation via natural descriptions. This capability emerges from the fact that the employed sentence embedding vector $\boldsymbol{s}$ concatenated to the z vector lie in a smooth embedding space that present structural regularity. In that particular kind of space we can find semantic regularities regarding concepts learned by the model, i.e., they respect a semantic organization of concepts. We observed that the use of our novel sentence interpolation strategy during training is quite helpful to improve the learned condition space. It increases the model capacity of learning a smooth condition space, in which embedding regularities are more easily found.

Fig. 4. Qualitative results in the CUB Dataset.

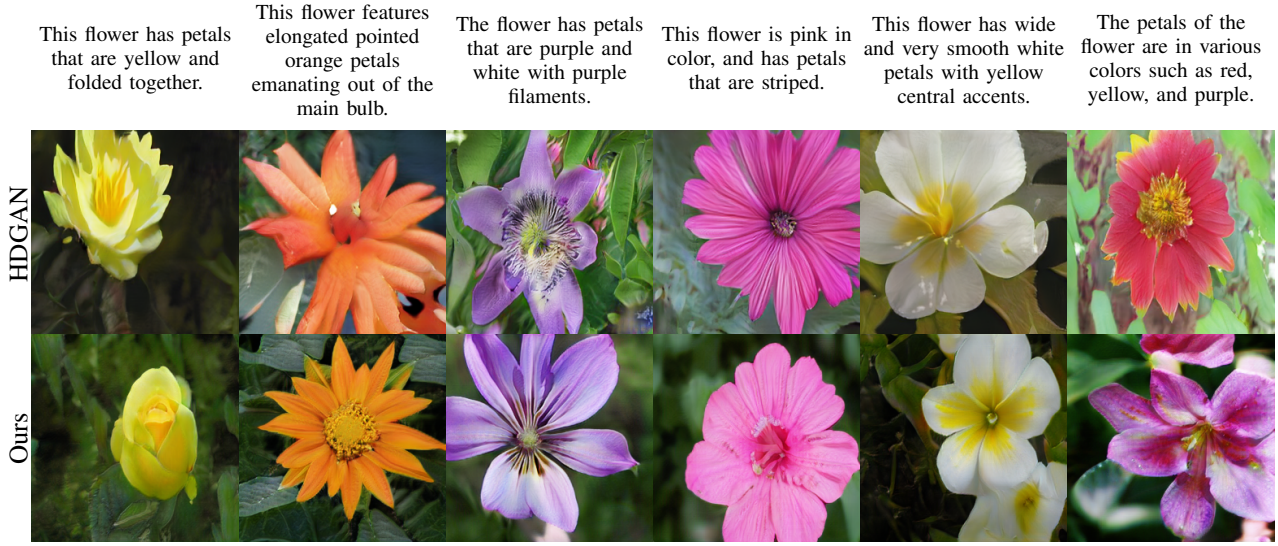
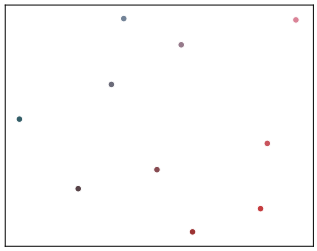


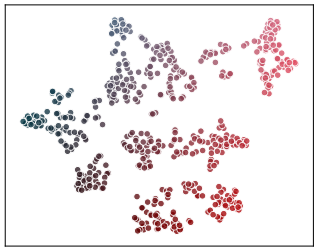
Fig. 5. Qualitative results in the Oxford-102 Dataset.

Fig. 2 showcases examples regarding regularities found in our trained models. For generating those images we hold z fixed, and embed captions into the multimodal space, which are used in simple vector operations, as follows. The uppermost example depicts an image generated by $G(z, \phi(\text{“}\textit{This is a red bird}\text{”}))$. We then subtract $\phi(\text{“}\textit{It is red}\text{”})$ from $\phi(\text{“}\textit{This is a red bird}\text{”})$, and generate a novel image (in the center). One can see that such operation completely removed the red color from the generated bird. Finally, we add $\phi(\text{“}\textit{It is blue}\text{”})$ to the resulting embedding, and use it to generate the rightmost image. That image shows the same bird, though with its color changed from red to blue, using only simple vector-level operations.

Note that our models are able to edit images while preserving the main image structural content without even being explicitly trained to learn disentangled representations. Fig. 2



(a) Sentence embeddings sampled without Sentence Interpolation.



(b) Sentence embeddings sampled with Sentence Interpolation.

Fig. 6. Manifold visualization of the sampled sentence embeddings during training. We visualize sentence embeddings by applying t-SNE [47] to project sentence embeddings from the original $\mathbb{R}^{256}$ space to a $\mathbb{R}^2$ space. We show 10 sentence embeddings of a randomly chosen image during the entire training (*i.e.*, resulting in 600 embeddings). In (a) is shown the regular sampling of a random sentence. In (b) is shown the sampling using the Sentence Interpolation.

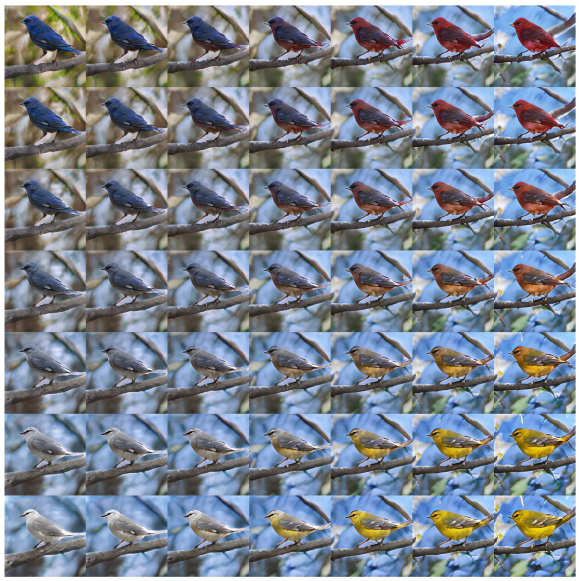


Fig. 7. Image generation with sentence embeddings linearly interpolated across all directions. There are four original embeddings, each one used to generate an image (those from the four corners), while all the remaining ones were generated using interpolated description embeddings. The upper-left position depicts an image generated with the description "It is a blue bird", the bottom-left image was generated with "It is a white bird", the upper-right image with "It is a red bird", and the bottom-right image with "It is a yellow bird".

also demonstrates that one can edit several aspects of the generated images, such as shape of the beak, and presence of colored crown.

VII. Impact of Sentence Interpolation

One of the contributions of this paper regards the introduction of a novel Sentence Interpolation procedure. In order to understand its effects, we have trained two models: (i) a default complete model that performs Sentence Interpolation; and (ii) a model with the same overall architecture, though without applying any interpolation between sentences. Fig. 3 shows per-epoch Inception Score values computed during the entirety of the training process. It arguably proves the importance of the proposed technique. During the early stages of training, results are indeed quite similar, the difference being more relevant after the 100^{th} epoch. Notably, after the 400^{th} epoch, IS results with Sentence Interpolation were consistently higher than 4.00, while the model without it surpassed that mark only twice throughout the training.

Effects of the SI approach also can be seen in Fig. 6. In this analysis, we plot ten sentence embeddings of a randomly chosen image during the entire training (*i.e.*, resulting in 600 embeddings). We plot the very same embeddings for the model trained with and without SI. We apply the t-SNE [47] technique on those embeddings so as to project $\mathbb{R}^{256}$ vectors onto a $\mathbb{R}^2$ space. Such a plot clearly shows that the proposed interpolation provides a much larger exploration of the sentence embedding manifold, allowing for sampling continuous points from that space. That sampling region is obviously constrained by the ten points regarding the image descriptions chosen. We intend to further extend this technique for future work, so as to allow sampling points from outside of those boundaries, without loosing semantic context. When training without it, one can only sample fixed discrete points, which poses a considerable constraint on the information carried on the condition vector. This analysis corroborates with our hypothesis that SI works also as a data-augmentation scheme, providing better generation results for points present in a larger region of the manifold.

VIII. Conclusion and Future Work

In this work, we propose a novel approach that shifts the architectural paradigm currently used in text-to-image methods. We show that an effective neural architecture can achieve state-of-the-art performance using a single stage training directly at the target resolution. By doing so, we not only introduce a simpler method for text-to-image synthesis but also point a new direction in text-to-image research.

In a future work we intend to explore different ways of introducing the sentence vector as condition in the discriminator. Since the projection conditioning introduced by Miyato *et al.* [42] was designed to work in conjunction with trainable class embeddings, we believe that there is space to be explored when the condition is a fixed sentence vector. Additionally, we would like to investigate the impact of adding components of

recent works, such as Attention Modules [16] and Memory networks [19].

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint ArXiv:1611.07004*, 2016.

[3] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," *arXiv preprint arXiv:1711.09020*, 2017.

[4] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," *arXiv preprint arXiv:1711.11585*, 2017.

[5] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint arXiv:1703.10593*, 2017.

[6] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.

[7] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *European Conference on Computer Vision*. Springer, 2016, pp. 597–613.

[8] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint arXiv:1609.04802*, 2016.

[9] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[12] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *International Conference on Machine Learning*, 2016, pp. 1060–1069.

[13] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5907–5915.

[14] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "Stackgan++: Realistic image synthesis with stacked generative adversarial networks," *arXiv preprint arXiv:1710.10916*, 2017.

[15] Z. Zhang, Y. Xie, and L. Yang, "Photographic text-to-image synthesis with a hierarchically-nested adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6199–6208.

[16] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1316–1324.

[17] T. Qiao, J. Zhang, D. Xu, and D. Tao, "Mirrorgan: Learning text-to-image generation by redescription," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1505–1514.

[18] G. Yin, B. Liu, L. Sheng, N. Yu, X. Wang, and J. Shao, "Semantics disentangling for text-to-image generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2327–2336.

[19] M. Zhu, P. Pan, W. Chen, and Y. Yang, "Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5802–5810.

[20] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[21] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[22] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 2008, pp. 722–729.

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, 2016, pp. 2234–2242.

[24] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, 2017, pp. 6626–6637.

[25] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[26] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[27] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[28] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[29] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[30] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018.

[31] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for gans do actually converge?" *arXiv preprint arXiv:1801.04406*, 2018.

[32] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Advances in Neural Information Processing Systems*, 2016, pp. 217–225.

[33] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[34] J. Wehrmann, C. Kolling, and R. C. Barros, "Adaptive cross-modal embeddings for image-text alignment," in *AAAI*, 2020, pp. 7718–7726.

[35] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "Vse++: Improving visual-semantic embeddings with hard negatives," 2018. [Online]. Available: https://github.com/fartashf/vsepp

[36] J. Wehrmann, M. A. Lopes, D. Souza, and R. Barros, "Language-agnostic visual-semantic embeddings," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 5803–5812.

[37] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," *arXiv preprint arXiv:1803.08024*, 2018.

[38] J. Wehrmann and R. C. Barros, "Bidirectional retrieval made simple," in *CVPR*, 2018, pp. 7718–7726.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[40] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[41] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *arXiv preprint arXiv:1610.07629*, 2016.

[42] T. Miyato and M. Koyama, "cgans with projection discriminator," *arXiv preprint arXiv:1802.05637*, 2018.

[43] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[45] A. Dash, J. C. B. Gamboa, S. Ahmed, M. Liwicki, and M. Z. Afzal, "Tac-gan-text conditioned auxiliary classifier generative adversarial network," *arXiv preprint arXiv:1703.06412*, 2017.

[46] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[47] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.