AttrLostGAN: Attribute Controlled Image Synthesis from Reconfigurable Layout and Style

Stanislav Frolov[1,2](✉), Avneesh Sharma[1], Jörn Hees[2], Tushar Karayil[2], Federico Raue[2], and Andreas Dengel[1,2]

[1] Technical University of Kaiserslautern, Kaiserslautern, Germany
asharma@rhrk.uni-kl.de
[2] German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
{stanislav.frolov,jorn.hees,tushar.karayil, federico.raue,andreas.dengel}@dfki.de

Abstract. Conditional image synthesis from layout has recently attracted much interest. Previous approaches condition the generator on object locations as well as class labels but lack fine-grained control over the diverse appearance aspects of individual objects. Gaining control over the image generation process is fundamental to build practical applications with a user-friendly interface. In this paper, we propose a method for attribute controlled image synthesis from layout which allows to specify the appearance of individual objects without affecting the rest of the image. We extend a state-of-the-art approach for layout-to-image generation to additionally condition individual objects on attributes. We create and experiment on a synthetic, as well as the challenging Visual Genome dataset. Our qualitative and quantitative results show that our method can successfully control the fine-grained details of individual objects when modelling complex scenes with multiple objects. Source code, dataset and pre-trained models are publicly available (https://github.com/stanifrolov/AttrLostGAN).

Keywords: Generative Adversarial Networks · Image synthesis

1 Introduction

The advent of Generative Adversarial Networks (GANs) [8] had a huge influence on the progress of image synthesis research and applications. Starting from low-resolution, gray-scale face images, current methods can generate high-resolution face images which are very difficult to distinguish from real photographs [20]. While unconditional image synthesis is interesting, most practical applications

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-030-92659-5_23.

C. Bauckhage et al. (Eds.): DAGM GCPR 2021, LNCS 13024, pp. 361–375, 2021.
https://doi.org/10.1007/978-3-030-92659-5_23

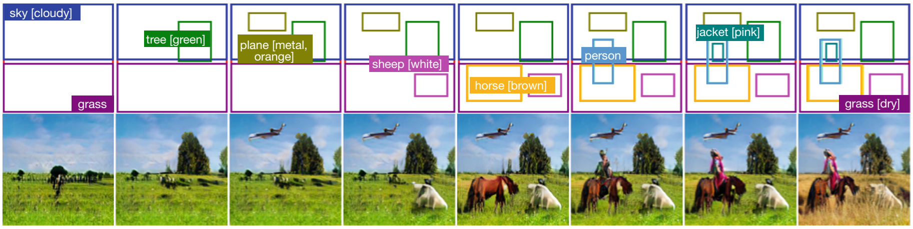

Fig. 1. Generated images using a reconfigurable layout and attributes to control the appearance of individual objects. *From left to right:* add tree [green], add plane [metal, orange], add sheep [white], add horse [brown], add person, add jacket [pink], grass → grass [dry]. (Color figure online)

require an interface which allows users to specify what the model should generate. In recent years, conditional generative approaches have used class labels [3,27], images [17,50], text [34,43,46], speech [4,42], layout [37,48], segmentation masks [6,41], or combinations of them [15], to gain control over the image generation process. However, most of these approaches are "one-shot" image generators which do not allow to reconfigure certain aspects of the generated image.

While there has been much progress on iterative image manipulation, researchers have so far not investigated how to gain better control over the image generation process of complex scenes with multiple interacting objects. To allow the user to create a scene that reflects what he/she has in mind, the system needs to be capable of iteratively and interactively updating the image. A recent approach by Sun and Wu [37] takes a major step towards this goal by enabling reconfigurable spatial layout and object styles. In their method, each object has an associated latent style code (sampled from a normal distribution) to create new images. However, this implies that users do not have true control over the specific appearance of objects. This lack of control also translates into the inability to specify a style (i.e., to change the color of a shirt from red to blue one would need to sample new latent codes and manually inspect whether the generated style conforms to the requirement). Being able to not just generate, but control individual aspects of the generated image without affecting other areas is vital to enable users to generate what they have in mind. To overcome this gap and give users control over style attributes of individual objects, we propose to extend their method to additionally incorporate attribute information. To that end, we propose Attr-ISLA as an extension of the Instance-Sensitive and Layout-Aware feature Normalization (ISLA-Norm) [37] and use an adversarial hinge loss on object-attribute features to encourage the generator to produce objects reflecting the input attributes. At inference time, a user can not only reconfigure the location and class of individual objects, but also specify a set of attributes. See Fig. 1 for an example of reconfigurable layout-to-image generation guided by attributes using our method. Since we continue to use latent codes for each object and the overall image, we can generate diverse images of objects with specific attributes. This approach not only drastically improves the flexibility but also allows the user to easily articulate the contents of his mind into the image generation process. Our contributions can be summarized as following:

- we propose a new method called AttrLostGAN, which allows attribute controlled image generation from reconfigurable layout;
- we extend ISLA to Attr-ISLA thereby gaining additional control over attributes;
- we create and experiment on a synthetic dataset to empirically demonstrate the effectiveness of our approach;
- we evaluate our model on the challenging Visual Genome dataset both qualitatively and quantitatively and achieve state-of-the-art performance.

2 Related Work

Class-Conditional Image Generation. Generating images given a class label is arguably the most direct way to gain control over what image to generate. Initial approaches concatenate the noise vector with the encoded label to condition the generator [27,30]. Recent approaches [3,33] have improved the image quality, resolution and diversity of generative models drastically. However, there are two major drawbacks that limit their practical application: they are based on single-object datasets, and do not allow reconfiguration of individual aspects of the image to be generated.

Layout-to-Image. The direct layout-to-image task was first studied in Layout2Im [48] using a VAE [21] based approach that could produce diverse 64×64 pixel images by decomposing the representation of each object into a specified label and an unspecified (sampled) appearance vector. LostGAN [37,38] allows better control over individual objects using a reconfigurable layout while keeping existing objects in the generated image unchanged. This is achieved by providing individual latent style codes for each object, wherein one code for the whole image allows to generate diverse images from the same layout when the object codes are fixed. We use LostGAN as our backbone to successfully address a fundamental problem: the inability to specify the appearance of individual objects using attributes.

Scene-Graph-to-Image. Scene graphs represent a scene of multiple objects using a graph structure where objects are encoded as nodes and edges represent the relationships between individual objects. Due to their convenient and flexible structure, scene graphs have recently been employed in multiple image generation approaches [1,11,18,39,45]. Typically, a graph convolution network (GCN) [10] is used to predict a scene layout containing segmentation masks and bounding boxes for each object which is then used to generate an image. However, scene graphs can be cumbersome to edit and do not allow to specify object locations directly on the image canvas.

Text-to-Image. Textual descriptions provide an intuitive way for conditional image synthesis [7]. Current methods [14,34,43,46,51] first produce a text

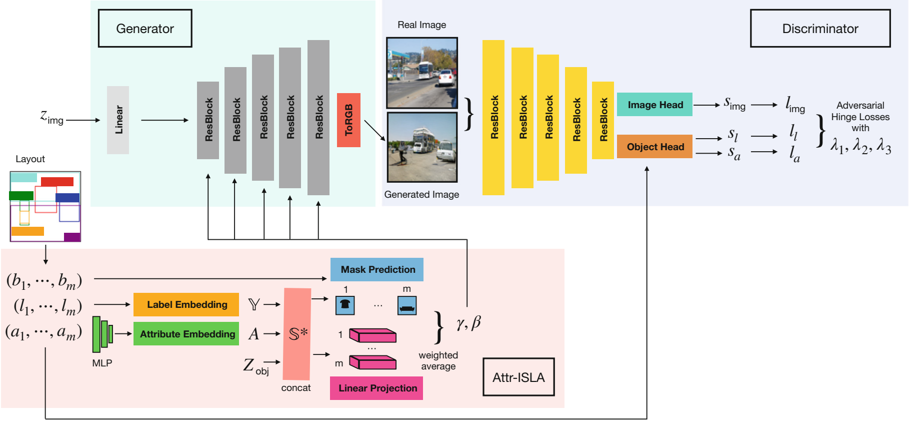

Fig. 2. Illustration of our proposed method for attribute controlled image synthesis from reconfigurable layout and style. Given a layout of object positions, class labels and attributes, we compute affine transformation parameters γ and β to condition the generator. Using separate latent codes Z_{obj} and z_{img} for individual objects and image, respectively, enables our model to produce diverse images. The discriminator minimizes three adversarial hinge losses on image features, object-label features, and object-attribute features.

embedding which is then input to a multi-stage image generator. In [13,14], additional layout information is used by adding an object pathway to learn the features of individual objects and control object locations. Decomposing the task into predicting a semantic layout from text, and then generate images conditioned on both text and semantic layout has been explored in [15,25]. Other works focus on disentangling content from style [23,49], and text-guided image manipulation [24,29]. However, natural language can be ambiguous and textual descriptions are difficult to obtain.

Usage of Attributes. Early methods used attributes to generate outdoor scene [5,19], human face and bird images [44]. In contrast, our method can generate complex scene images containing multiple objects from a reconfigurable layout. Most similar to our work are the methods proposed by Ke Ma et al. [26] as an extension of [26] using an auxiliary attribute classifier and explicit reconstruction loss for horizontally shifted objects, and [31] which requires semantic instance masks. To the best of our knowledge, [26] is currently the only other direct layout-to-image method using attributes. Our method improves upon [26] in terms of visual quality, control and image resolution using a straightforward, yet effective approach built on [37,38].

3 Approach

LostGAN [37,38] achieves remarkable results and control in the layout-to-image task, but it lacks the ability to specify the attributes of an object. While an object class label defines the high-level category (e.g., “car”, “person”, “dog”, “building)”, attributes refer to structural properties and appearance variations such as colors (e.g., “blue”, “yellow”), sentiment (e.g., “happy”, “angry”), and forms (e.g., “round”, “sliced”) which can be assigned to a variety of object classes [22]. Although one could randomly sample many different object latent codes to generate diverse outputs, it does not allow to provide specific descriptions of the appearance to enable users to generate “what they have in mind”. To address this fundamental problem, we build upon [37] and additionally condition individual objects on a set of attributes. To that end, we create an attribute embedding, similar to the label embedding used in [37], and propose Attr-ISLA to compute affine transformation parameters which depend on object positions, class labels and attributes. Furthermore, we utilize a separate attribute embedding to compute an additional adversarial hinge loss on object-attribute features. See Fig. 2 for an illustration of our method.

3.1 Problem Formulation

Given an image layout $L = \{(l_i, b_i, a_i)_{i=1}^{m}\}$ of m objects, where each object is defined by a class label l_i, a bounding box b_i, and attributes a_i, the goal of our method is to generate an image I with accurate positioned and recognizable objects which also correctly reflect their corresponding input attributes. We use LostGAN [37,38] as our backbone, in which the overall style of the image is controlled by the latent z_{img}, and individual object styles are controlled by the set of latents $Z_{\text{obj}} = \{z_i\}_{i=1}^{m}$. Latent codes are sampled from the standard normal distribution $\mathcal{N}(0,1)$. Note, the instance object style codes Z_{obj} are important even though attributes are provided to capture the challenging one-to-many mapping and enable the generation of diverse images (e.g., there are many possible images of a person wearing a blue shirt). In summary, we want to find a generator function G parameterized by Θ_G which captures the underlying conditional data distribution $p = (I|L, z_{\text{img}}, Z_{\text{obj}})$ such that we can use it to generate new, realistic samples. Similar to [37], the task we are addressing in this work can hence be expressed more formally as in Eq. 1

$$I = G(L, z_{\text{img}}, Z_{\text{obj}}; \Theta_G), \tag{1}$$

where all components of the layout L (i.e., class labels l_i, object positions b_i and attributes a_i) are reconfigurable to allow fine-grained control of diverse images using the randomly sampled latents z_{img} and Z_{obj}. In other words, our goals are 1) to control the appearance of individual objects using attributes, but still be able to 2) reconfigure the layout and styles to generate diverse objects corresponding to the desired specification.

3.2 Attribute ISLA (Attr-ISLA)

Inspired by [3,20,28], the authors of [37] extended the Adaptive Instance Normalization (AdaIN) [20] to object Instance-Sensitive and Layout-Aware feature Normalization (ISLA-Norm) to enable fine-grained and multi-object style control.

In order to gain control over the appearance of individual objects, we propose to additionally condition on object attributes using a simple, yet effective enhancement to the ISLA-Norm [37]. On a high-level, the channel-wise batch mean μ and variance σ are computed as in BatchNorm [16] while the affine transformation parameters γ and β are instance-sensitive (class labels and attributes) and layout-aware (object positions) per sample. Similar to [37], this is achieved in a multi-step process:

1) Label Embedding: Given one-hot encoded label vectors for m objects with d_l denoting the number of class labels, and d_e the embedding dimension, the one-hot label matrix Y of size $m \times d_l$ is transformed into the $m \times d_e$ label-to-vector matrix representation of labels $\mathbb{Y} = Y \cdot W$ using a learnable $d_l \times d_e$ size embedding matrix W.
2) Attribute Embedding: Given binary encoded attribute vectors for m objects, an intermediate MLP is used to map the attributes into an $m \times d_e$ size attribute-to-vector matrix representation A.
3) Joint Label, Attribute & Style Projection: The sampled object style noise matrix Z_{obj} of size $m \times d_{\text{noise}}$ is concatenated with the label-to-vector matrix $\mathbb{Y}$ and attribute-to-vector matrix A to obtain the $m \times (2 \cdot d_e + d_{\text{noise}})$ size embedding matrix $\mathbb{S}^* = (\mathbb{Y}, A, Z_{\text{obj}})$. The embedding matrix $\mathbb{S}^*$, which now depends on the class labels, attributes and latent style codes, is used to compute object attribute-guided instance-sensitive channel-wise γ and β via linear projection using a learnable $(2 \cdot d_e + d_{\text{noise}}) \times 2C$ projection matrix, with C denoting the number of channels.
4) Mask Prediction: A non-binary $s \times s$ mask is predicted for each object by a sub-network consisting of up-sample convolutions and a sigmoid transformation. Next, the masks are resized to the corresponding bounding box sizes.
5) ISLA γ, β Computation: The γ and β parameters are unsqueezed to their corresponding bounding boxes, weighted by the predicted masks, and finally added together with averaged sum used for overlapping regions.

Because the affine transformation parameters depend on individual objects in a sample (class labels, bounding boxes, attributes and styles), our AttrLostGAN achieves better and fine-grained control over the image generation process. This allows the user to create an image iteratively and interactively by updating the layout, specifying attributes, and sampling latent codes. We refer the reader to [37] for more details on ISLA and [38] for an extended ISLA-Norm which integrates the learned masks at different stages in the generator.

3.3 Architecture and Objective

We use LostGAN [37,38] as our backbone without changing the general architecture of the ResNet [9] based generator and discriminator. The discriminator $D(\cdot; \Theta_D)$ consists of three components: a shared ResNet backbone to extract features, an image head classifier, and an object head classifier. Following the design of a separate label embedding to compute the object-label loss, we create a separate attribute embedding to compute the object-attribute loss to encourage the generator G to produce objects with specified attributes. Similar to [37,38], the objective can be formulated as follows. Given an image I, the discriminator predicts scores for the image (s_{img}), and average scores for the object-label (s_l) and object-attribute (s_a) features, respectively:

$$(s_{\text{img}}, s_l, s_a) = D(I, L; \Theta_D) \tag{2}$$

We use the adversarial hinge losses

$$\mathcal{L}_t(I, L) = \begin{cases} \max(0, 1 - s_t); & \text{if } I \text{ is real} \\ \max(0, 1 + s_t); & \text{if } I \text{ is fake} \end{cases} \tag{3}$$

where $t \in \{\text{img}, l, a\}$. The objective can hence be written as

$$\mathcal{L}(I, L) = \lambda_1 \mathcal{L}_{\text{img}}(I, L) + \lambda_2 \mathcal{L}_l(I, L) + \lambda_3 \mathcal{L}_a(I, L), \tag{4}$$

where $\lambda_1, \lambda_2, \lambda_3$ are trade-off parameters between image, object-label, and object-attribute quality. The losses for the discriminator and generator can be written as

$$\begin{aligned} \mathcal{L}_D &= \mathbb{E}\left[\mathcal{L}\left(I^{\text{real}}, L\right) + \mathcal{L}\left(I^{\text{fake}}, L\right)\right] \\ \mathcal{L}_G &= -\mathbb{E}\left[\mathcal{L}\left(I^{\text{fake}}, L\right)\right] \end{aligned} \tag{5}$$

We set $\lambda_1 = 0.1$, $\lambda_2 = 1.0$, and $\lambda_3 = 1.0$ to obtain our main results in Table 1, and train our models for 200 epochs using a batch size of 128 on three NVIDIA V100 GPUs. Both λ_1, and λ_2 are as in [37]. We use the Adam optimizer, with $\beta_1 = 0$, $\beta_2 = 0.999$, and learning rates 10^{-4} for both generator and discriminator.

4 Experiments

Since we aim to gain fine-grained control of individual objects using attributes, we first create and experiment with a synthetic dataset to demonstrate the effectiveness of our approach before moving to the challenging Visual Genome [22] dataset.

4.1 MNIST Dialog

Dataset. We use the MNIST Dialog [36] dataset and create an annotated layout-to-image dataset with attributes. In MNIST Dialog each image is a 28×28 pixel MNIST image with three additional attributes, i.e., digit color (red, blue, green, purple, or brown), background color (cyan, yellow, white, silver, or salmon), and style (thick or thin). Starting from an empty 128×128 image canvas, we randomly select, resize and place 3–8 images on it thereby creating an annotated layout-to-image with attributes dataset, where each "object" in the image is an image from MNIST Dialog. While randomly placing the images on the canvas, we ensure that each image is sufficiently visible by allowing max. 40% overlap between any two images.

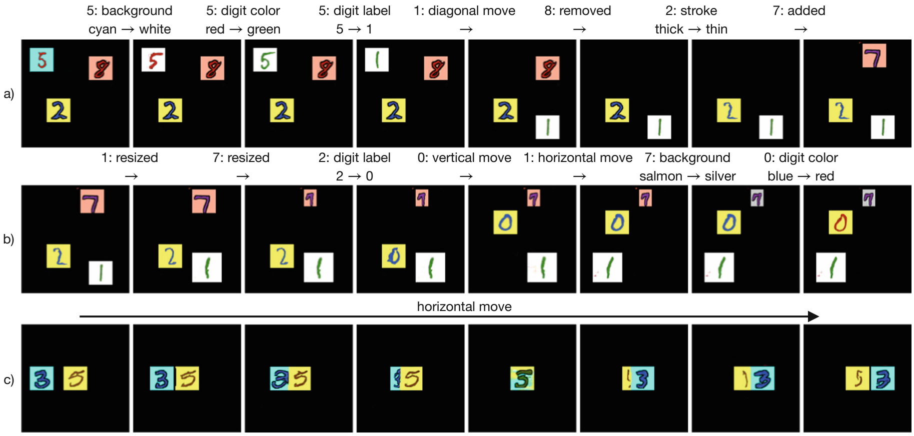

Fig. 3. Iterative reconfiguration example on the MNIST Dialog based dataset (all images are generated by our model). In a) and b), we reconfigure various aspects of the (not shown) input layout to demonstrate controlled image generation. Our approach allows fine-grained control over individual objects with no or minimal changes to other parts of the image. During reconfiguration, we can sometimes observe small style changes, indicating partial entanglement of latent codes and specified attributes. In c), we horizontally shift one object showing that nearby and overlapping objects influence each other which might be a desirable feature to model interactions in more complex settings.

Results. In Fig. 3, we depict generated images using a corresponding layout. Our model learned to generate sharp objects at the correct positions with corresponding labels and attributes, and we can successfully control individual object attributes without affecting other objects. When reconfiguring one object we can sometimes observe slight changes in the style of how a digit is drawn, indicating that the variation provided by the object latent codes is not fully disentangled from the attribute specification. However, we also observe that other objects remain unchanged, hence providing fine-grained control over the individual appearance. We further show how two nearby or even overlapping objects can influence each other which might be necessary to model interacting objects in more complex settings. We hypothesize this is due to the weighted average pool in ISLA which computes an average style for that position.

4.2 Visual Genome

Dataset. Finally, we apply our proposed method to the challenging Visual Genome [22] dataset. Following the setting in [18,37], we pre-process and split the dataset by removing small and infrequent objects, resulting in 62,565 training, 5,062 validation, and 5,096 testing images, with 3 to 30 objects from 178 class labels per image. We filter all available attributes to include only such that appear at least 2,000 times, and allow up to 30 attributes per image.

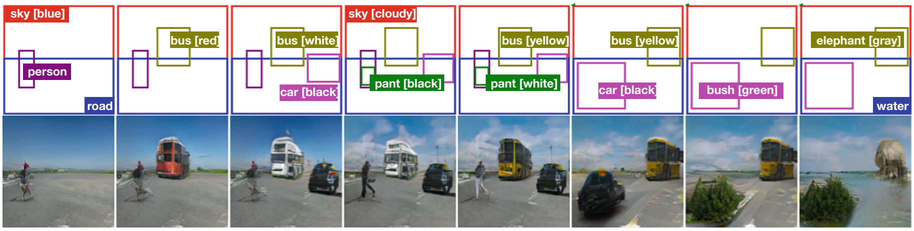

Fig. 4. Our model can control the appearance of generated objects via attributes. *From left to right:* add bus [red]; bus [red → white], add car [black]; sky [blue → cloudy], add pant [black]; bus [white → yellow], pant [black → white]; remove person, remove pant [white], reposition bus [yellow], reposition and resize car [black]; car [black] → bush [green]; road → water, bus [yellow] → elephant [gray]. (Color figure online)

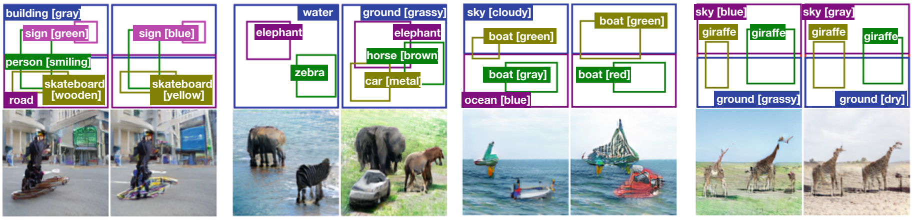

Fig. 5. More reconfiguration examples. *First pair:* sign [green → blue], skateboard [wooden → yellow]. *Second pair:* water → ground [grassy], zebra → horse [brown], repositioned horse [brown], resized elephant, added car [metal]. *Third pair:* boat [gray → red], resized boat [green]. *Fourth pair:* sky [blue → gray], ground [grassy → dry], resized both giraffes. (Color figure online)

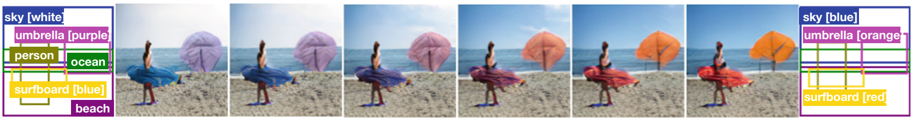

Fig. 6. Generating images from a linear interpolation between attribute embeddings produces smooth transitions. From left to right, we interpolate between the following attribute specification: sky [white → blue], umbrella [purple → orange], surfboard [blue → red]. (Color figure online)

Metrics. Evaluating generative models is challenging because there are many aspects that would resemble a good model such as visual realism, diversity, and sharp objects [2,40]. Additionally, a good layout-to-image model should generate objects of the specified class labels and attributes at their corresponding locations. Hence, we choose multiple metrics to evaluate our model and compare with baselines.

To evaluate the image quality and diversity, we use the IS [35] and FID [12]. To assess the visual quality of individual objects we choose the SceneFID

[39] which corresponds to the FID applied on cropped objects as defined by the bounding boxes. Similarly, we propose to apply the IS on generated object crops, denoted as SceneIS. As in [37], we use the CAS [32], which measures how well an object classifier trained on generated can perform on real image crops. Note, this is different to the classification accuracy as used in [26,48], which is trained on real and tested on generated data, and hence might overlook the diversity of generated images [38]. Additionally, and in the same spirit as CAS, we report the micro F1 (Attr-F1) by training multi-label classification networks to evaluate the attribute quality by training on generated and test on real object crops. As in [26,37,38,48], we adopt the LPIPS metric as the Diversity Score (DS) [47] to compute the perceptual similarity between two sets of images generated from the same layout in the testing set.

Table 1. Results on Visual Genome. Our models (in italics) achieve the best scores on most metrics, and AttrLostGANv2 is considerably better than AttrLostGANv1. When trained on higher resolution, our model performs better in terms of object and attribute quality, while achieving similar scores on image quality metrics. Note, a lower diversity (DS) is expected due to the specified attributes. Models marked with $\diamond$, $\dagger$, $\star$ are trained with an image resolution of 64×64, 128×128, and 256×256, respectively.

Method	IS ↑	SceneIS ↑	FID ↓	SceneFID ↓	DS (↑)	CAS ↑	Attr-F1 ↑
Real images†	23.50	13.43	11.93	2.46	-	46.22	15.77
Layout2Im [48]$^\diamond$	8.10	-	40.07	-	0.17	-	-
LostGANv1 [37]†	10.30	9.07	35.20	11.06	0.47	31.04	-
LostGANv2 [38]†	10.25	9.15	34.77	15.25	0.42	30.97	-
Ke Ma et al. [26]†	9.57	8.17	43.26	16.16	0.30	**33.09**	12.62
AttrLostGANv1†	10.68	9.24	32.93	8.71	0.40	32.11	13.64
AttrLostGANv2†	**10.81**	**9.46**	**31.57**	**7.78**	0.28	32.90	**14.61**
Real images*	31.41	19.58	12.41	2.78	-	50.94	17.80
LostGANv2 [38]*	**14.88**	11.87	**35.03**	18.87	0.53	**35.80**	-
AttrLostGANv2*	14.25	**11.96**	35.73	**14.76**	0.45	35.36	**14.49**

Qualitative Results. Figure 1 and Fig. 4 depict examples of attribute controlled image generation from reconfigurable layout. Our model provides a novel way to iteratively reconfigure the properties of individual objects to generate images of complex scenes without affecting other parts of the image. Figure 5 shows more examples in which we reconfigure individual objects by changing attributes, class labels, object position and size. In Fig. 6, we linearly interpolate between two sets of attributes for the same layout. Our model learns a smooth transition between attributes. In Fig. 7 we compare generated images between our AttrLostGANv1 and [26] using layouts from the testing set. As can be seen, generating realistic images of complex scenes with multiple objects is still very difficult. Although the images generated by our model look more realistic, individual objects and details such as human faces are hard to recognize. In terms of attribute control, our images better depict the input specifications in general.

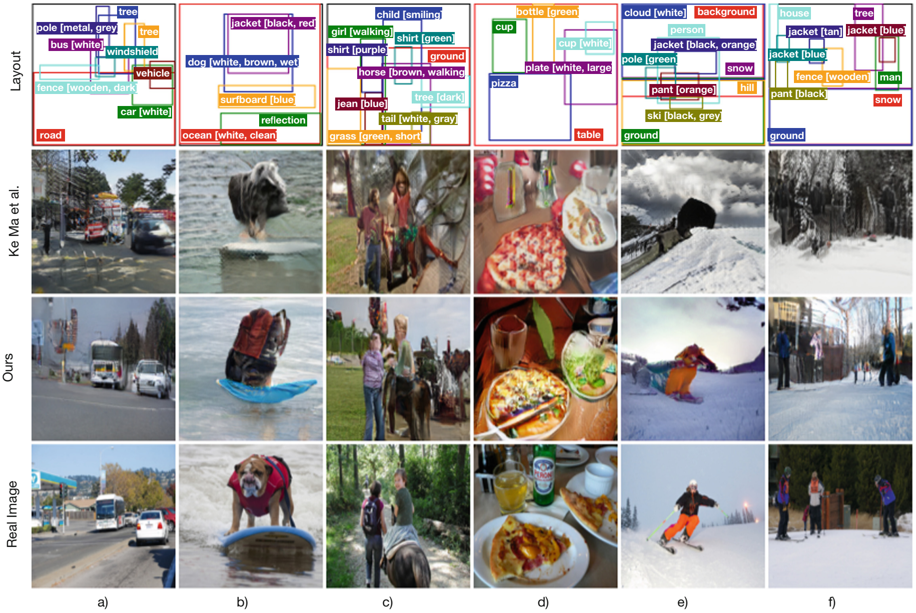

Fig. 7. Visual comparison between images generated by our AttrLostGANv1 and Ke Ma et al. [26] using the layouts shown in the first row. Our images are consistently better at reflecting the input attributes and individual objects have more details and better texture. For example, a) bus [white], b) jacket [black, red], c) shirt [purple], d) plate [white, large], e) pant [orange], f) jacket [blue]. (Color figure online)

Quantitative Results. Table 1 shows quantitative results. We train two variants of our approach: AttrLostGANv1 which is based on [37], and AttrLostGANv2 which is based on [38]. We compare against the recent and only other direct layout-to-image baseline proposed by Ke Ma et al. [26], which is an extension of Layout2Im [48] that can be conditioned on optional attributes. Since no pre-trained model was available at the official codebase of [26], we used the open-sourced code to train a model. For fair comparison, we evaluate all models trained by us. Our models achieve the best scores across most metrics, and AttrLostGANv2 is considerably better than AttrLostGANv1. [26] reaches a competing performance on attribute control, but inferior image and object quality. For example, our method increases the SceneIS from 8.17 to 9.46, and lowers the FID from 43.26 to 31.57. Furthermore, our method is better at generating the appearance specified by the attributes as indicated by the improvement of Attr-F1 from 12.62 to 14.61. In terms of CAS, our model performs slightly worse than [26], which might be due to the explicit attribute classifier used in [26] during training. Building upon [38] our method can also generate higher resolution images (256×256 compared to 128×128). By specifying attributes, a decreased DS is expected but we include it for completeness.

Ablation Study. We also perform ablations of our main changes to the LostGAN [37] backbone, see Appendix C. Starting from LostGAN we add attribute information to the generator and already gain an improvement over the baseline in terms of image quality, object discriminability, as well as attribute information. We ablate the additional adversarial hinge loss on object-attribute features λ_3. A higher λ_3 leads to better Attr-F1, but decreased image and object quality. Interestingly, a high $\lambda_3 = 2.0$ achieves the best SceneFID on object crops, while the image quality in terms of FID is worst. Although we only have to balance three weights, our results show that there exists a trade-off between image and object quality. We choose $\lambda_3 = 1.0$ for all remaining experiments and ablate the depth of the intermediate attribute MLP which is used to compute the attribute-to-vector matrix representation for all objects. While a shallow MLP leads to a decreased performance, a medium deep MLP with three hidden layers achieves the best overall performance.

4.3 Discussion

Our approach takes an effective step towards reconfigurable, and controlled image generation from layout of complex scenes. Our model provides unprecedented control over the appearance of individual objects without affecting the overall image. Although the quantitative as well as visual results are promising current approaches require attribute annotations which are time-consuming to obtain. While the attribute control is strong when fixing the object locations, as demonstrated in our results, the object styles can change when target objects are nearby or overlap. We hypothesize that this might be due to the average pool in ISLA when combining label and attribute features of individual objects and might hence lead to entangled representations. At the same time, such influence might be desirable to model object interactions in complex settings. Despite clearly improving upon previous methods both quantitatively and qualitatively, current models are still far from generating high-resolution, realistic images of complex scenes with multiple interacting objects which limits their practical application.

5 Conclusion

In this paper, we proposed AttrLostGAN, an approach for attribute controlled image generation from reconfigurable layout and style. Our method successfully addresses a fundamental problem by allowing users to intuitively change the appearance of individual object details without changing the overall image or affecting other objects. We created and experimented on a synthetic dataset based on MNIST Dialog to analyze and demonstrate the effectiveness of our approach. Further, we evaluated our method against the recent, and only other baseline on the challenging Visual Genome dataset both qualitatively and quantitatively. We find that our approach not only outperforms the existing method in most common measures while generating higher resolution images, but also that it provides users with intuitive control to update the generated image to their needs. In terms of future work, our first steps are directed towards enhancing the image quality and resolution. We would also like to investigate unsupervised methods to address the need of attribute annotations and whether we can turn attribute labels into textual descriptions.

Acknowledgments. This work was supported by the BMBF projects ExplAINN (Grant 01IS19074), XAINES (Grant 01IW20005), the NVIDIA AI Lab (NVAIL) and the TU Kaiserslautern PhD program.

References

1. Ashual, O., Wolf, L.: Specifying object attributes and relations in interactive scene generation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4561–4569 (2019)
2. Borji, A.: Pros and cons of GAN evaluation measures. Comput. Vis. Image Underst. **179**, 41–65 (2018)
3. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: International Conference on Learning Representations (2018)
4. Choi, H.S., Park, C.D., Lee, K.: From inference to generation: end-to-end fully self-supervised generation of human face from speech. In: International Conference on Learning Representations (2020)
5. 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 Computer Vision and Pattern Recognition, pp. 8789–8797 (2018)
6. Dong, H., Yu, S., Wu, C., Guo, Y.: Semantic image synthesis via adversarial learning. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
7. Frolov, S., Hinz, T., Raue, F., Hees, J., Dengel, A.: Adversarial text-to-image synthesis: a review. arXiv:2101.09983 (2021)
8. Goodfellow, I.J., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Vision and Pattern Recognition, pp. 770–778 (2016)
10. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv:1506.05163 (2015)
11. Herzig, R., Bar, A., Xu, H., Chechik, G., Darrell, T., Globerson, A.: Learning canonical representations for scene graph to image generation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12371, pp. 210–227. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58574-7_13
12. 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: Advances in Neural Information Processing Systems, pp. 6626–6637 (2017)
13. Hinz, T., Heinrich, S., Wermter, S.: Generating multiple objects at spatially distinct locations. In: International Conference on Learning Representations (2019)
14. Hinz, T., Heinrich, S., Wermter, S.: Semantic object accuracy for generative text-to-image synthesis. IEEE Trans. Pattern Anal. Mach. Intell. **14**, 1–14 (2020)

15. Hong, S., Yang, D., Choi, J., Lee, H.: Inferring semantic layout for hierarchical text-to-image synthesis. In: Proceedings of the IEEE Computer Vision and Pattern Recognition, pp. 7986–7994 (2018)
16. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)
17. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Computer Vision and Pattern Recognition, pp. 1125–1134 (2016)
18. Johnson, J., Gupta, A., Fei-Fei, L.: Image generation from scene graphs. In: Proceedings of the IEEE Computer Vision and Pattern Recognition, pp. 1219–1228 (2018)
19. Karacan, L., Akata, Z., Erdem, A., Erdem, E.: Learning to generate images of outdoor scenes from attributes and semantic layouts. arXiv:1612.00215 (2016)
20. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE Computer Vision and Pattern Recognition, pp. 4401–4410 (2018)
21. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. CoRR arXiv:1312.6114 (2013)
22. Krishna, R., et al.: Visual genome: connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vision **123**(1), 32–73 (2017)
23. Li, B., Qi, X., Lukasiewicz, T., Torr, P.H.S.: Controllable text-to-image generation. In: Advances in Neural Information Processing Systems (2019)
24. Li, B., Qi, X., Lukasiewicz, T., Torr, P.H.: ManiGAN: text-guided image manipulation. In: Proceedings of the IEEE Computer Vision and Pattern Recognition, pp. 7880–7889 (2020)
25. Li, W., et al.: Object-driven text-to-image synthesis via adversarial training. In: Proceedings of the IEEE Computer Vision and Pattern Recognition, pp. 12166–12174 (2019)
26. Ma, K., Zhao, B., Sigal, L.: Attribute-guided image generation from layout. In: British Machine Vision Virtual Conference (2020). arXiv:2008.11932
27. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv:1411.1784 (2014)
28. Miyato, T., Koyama, M.: cGANs with projection discriminator. arXiv:1802.05637 (2018)
29. Nam, S., Kim, Y., Kim, S.J.: Text-adaptive generative adversarial networks: manipulating images with natural language. In: Advances in Neural Information Processing Systems, pp. 42–51 (2018)
30. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier GANs. In: International Conference on Machine Learning, pp. 2642–2651 (2016)
31. Pavllo, D., Lucchi, A., Hofmann, T.: Controlling style and semantics in weakly-supervised image generation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12351, pp. 482–499. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58539-6_29
32. Ravuri, S., Vinyals, O.: Classification accuracy score for conditional generative models. In: Advances in Neural Information Processing Systems, pp. 12268–12279 (2019)
33. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with VQ-VAE-2. In: Advances in Neural Information Processing Systems, pp. 14866–14876 (2019)

34. Reed, S.E., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: International Conference on Machine Learning, pp. 1060–1069 (2016)
35. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: Advances in Neural Information Processing Systems, pp. 2234–2242 (2016)
36. Seo, P.H., Lehrmann, A., Han, B., Sigal, L.: Visual reference resolution using attention memory for visual dialog. In: Advances in Neural Information Processing Systems, pp. 3719–3729 (2017)
37. Sun, W., Wu, T.: Image synthesis from reconfigurable layout and style. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
38. Sun, W., Wu, T.: Learning layout and style reconfigurable GANs for controllable image synthesis. arXiv:2003.11571 (2020)
39. Sylvain, T., Zhang, P., Bengio, Y., Hjelm, R.D., Sharma, S.: Object-centric image generation from layouts. arXiv:2003.07449 (2020)
40. Theis, L., van den Oord, A., Bethge, M.: A note on the evaluation of generative models. CoRR arXiv:1511.01844 (2015)
41. 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 Computer Vision and Pattern Recognition, pp. 8798–8807 (2017)
42. Wang, X., Qiao, T., Zhu, J., Hanjalic, A., Scharenborg, O.: S2IGAN: speech-to-image generation via adversarial learning. In: INTERSPEECH (2020)
43. Xu, T., et al.: AttnGAN: fine-grained text to image generation with attentional generative adversarial networks. In: Proceedings of the IEEE Computer Vision and Pattern Recognition, pp. 1316–1324 (2017)
44. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2Image: conditional image generation from visual attributes. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 776–791. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_47
45. Yikang, L., Ma, T., Bai, Y., Duan, N., Wei, S., Wang, X.: PasteGAN: a semi-parametric method to generate image from scene graph. In: Advances in Neural Information Processing Systems, pp. 3948–3958 (2019)
46. Zhang, H., et al.: StackGAN++: realistic image synthesis with stacked generative adversarial networks. IEEE Trans. Pattern Anal. Mach. Intell. **41**, 1947–1962 (2017)
47. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Computer Vision and Pattern Recognition (2018)
48. Zhao, B., Meng, L., Yin, W., Sigal, L.: Image generation from layout. In: Proceedings of the IEEE Computer Vision and Pattern Recognition (2019)
49. Zhou, X., Huang, S., Li, B., Li, Y., Li, J., Zhang, Z.: Text guided person image synthesis. In: Proceedings of the IEEE Computer Vision and Pattern Recognition, pp. 3663–3672 (2019)
50. 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, pp. 2223–2232 (2017)
51. 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 Computer Vision and Pattern Recognition, pp. 5802–5810 (2019)