Layout-Bridging Text-to-Image Synthesis

Jiadong Liang, Wenjie Pei, Member, IEEE, and Feng Lu, Senior Member, IEEE

Abstract—The crux of text-to-image synthesis stems from the difficulty of preserving the cross-modality semantic consistency between the input text and the synthesized image. Typical methods, which seek to model the text-to-image mapping directly, could only capture keywords in the text that indicates common objects or actions but fail to learn their spatial distribution patterns. An effective way to circumvent this limitation is to generate an image layout as guidance, which is attempted by a few methods. Nevertheless, these methods fail to generate practically effective layouts due to the diversity of input text and object location. In this paper we push for effective modeling in both text-to-layout generation and layout-to-image synthesis. Specifically, we formulate the text-to-layout generation as a sequence-to-sequence modeling task, and build our model upon Transformer to learn the spatial relationships between objects by modeling the sequential dependencies between them. In the stage of layout-to-image synthesis, we focus on learning the textual-visual semantic alignment per object in the layout to precisely incorporate the input text into the layout-to-image synthesizing process. To evaluate the quality of generated layout, we design a new metric specifically, dubbed Layout Quality Score, which considers both the absolute distribution errors of bounding boxes in the layout and the mutual spatial relationships between them. Extensive experiments on three datasets demonstrate the superior performance of our method over state-of-the-art methods on both predicting the layout and synthesizing the image from the given text.

Index Terms—Layout generation, text-to-image synthesis, transformer, cross-modality.

I. INTRODUCTION

TEXT-TO-IMAGE synthesis aims to synthesize a realistic image that is consistent with the textual description. It has extensive applications ranging from artistic creation to computer-aided design. Text-to-image synthesis is quite challenging in that it demands not only high quality of the synthesized image, but also cross-modality semantic consistency between the given text and the synthesized image.

Most existing methods [1], [2], [3], [4], [5], [6] for text-to-image synthesis focus on modeling direct text-to-image semantic mapping by incorporating the input textual information into the generative processing. Despite the substantial progress made by these methods, an important limitation is that such methods can only capture keywords in the text indicating typical objects or actions, but fail to learn the spatial distribution patterns of objects in the image, namely the image layout. This is basically resulted from the difficulties of modeling the text-to-image mapping directly. As a result, the synthesized images do not have a reasonable object layout with correct spatial relationships between objects. As an example, AttGAN [2] in Figure 1 could only synthesize an image containing texture of person but fail to capture a reasonable layout between ‘person’ and ‘skis’.

An intuitive and straightforward way to tackle above limitation is to generate an image layout that is consistent with the given text, and then use the layout as a transitional modality to bridge the gap between the input text and the synthesized image. A state-of-the-art method following this way is Obj-GANs [7], which employs a bi-directional LSTM [8] to learn hidden representations for the input text and then generates a layout by training another LSTM. Whilst Obj-GANs can obtain plausible layout in some cases, the generated layout tends to over-fit simple patterns, which cannot reflect the diversity in real scenarios due to its limited capability of modeling object layout. Besides, it is also arduous for Obj-GANs to learn the complex spatial relationships between objects. As shown in Figure 1, the generated layout by Obj-GANs fails to indicate the correct spatial relationship between ‘person’ and ‘skis’.

Fig. 1. Given the input text, AttGAN [2] seeks to synthesize the output image directly while our method and Obj-GANs [7] generate a layout to bridge the text and the output image. Our method is able to generate more reasonable object layout between ‘person’ and ‘skis’ than Obj-GANs [7]. Guided by the generated layout, our method can synthesize an image with more consistent semantics with the given text than both other methods.
In this paper, we decompose the text-to-image synthesis task into two easier sub-tasks: text-to-layout generation and layout-to-image synthesis, and push for the effective modeling in both two sub-tasks. To learn the text-to-layout mapping precisely, we formulate the layout generating process as a sequential prediction of objects and model the objects’ relationships as sequential dependencies. As a result, the text-to-layout generation boils down to a sequence-to-sequence modeling task. We adopt Transformer [9] as the internal encoding and decoding structure due to its powerful capability of sequence-to-sequence modeling. The pivotal point to adapt Transformer to text-to-layout generation lies in how to predict the category and the location for each object in the layout correctly. We perform a unified prediction scheme, in which we partition the layout area into $S \times S$ grids equally and predict the object category and its center position jointly. Such unified scheme is not only efficient for prediction, but also enables our model to better perceive the object state and model the object dependencies by associating the category and location together.

The generated layout is further leveraged to guide the image synthesis in the stage of layout-to-image synthesis. To be specific, we draw on the generative structure of LostGANs_V2 [10] and focus on learning the textual-visual semantic alignment per object in the layout to precisely incorporate the input textual description into the layout-to-image synthesizing process.

The presented example in Figure 1 shows that our model is able to generate much more reasonable layout than ObjGANs, and thereby synthesize more consistent image with the given textual description than other two methods. To conclude, we make following contributions.

- We design a specific model based on Transformer to generate high-quality image layout which is consistent with the input text.
- A layout-to-image synthesizer is proposed to synthesize the image by incorporating both the generated layout and the input text.
- We propose quantitative metrics for evaluating the quality of generated layout.
- Extensive experiments on three datasets demonstrate that our proposed method performs favorably against state-of-the-art methods on both predicting the image layout and synthesizing the image from the given text.

### II. RELATED WORKS

#### A. Text-to-Image Synthesis Without Layout

Generative adversarial network (GAN) [11] has been widely employed for text-to-image synthesis. Representative methods include: Reed et al. [12] that proposes an early work for text-to-image synthesis at low resolutions, StackGAN [1] which achieves high resolution image synthesis with a coarse-to-fine framework, AttnGAN [2] leveraging word-level features as input to perform fine-grained image generation by employing attention mechanism, DMGAN [3] that introduces a dynamic memory module to fix distortions of initial images that usually cause large errors in the subsequent generations, and MirrorGAN [13] which first adopts a text-to-image-to-text cycle framework to guarantee text-image consistency. Based on these fundamental works, the research on text-to-image synthesis is further advanced on different aspects, as discussed below. There are also several approaches [14], [15], [16], [17] that explore the application of attention mechanisms in the field.

1) Parsing the Semantics of Input Text: Some works [6], [18], [19], [20], [21] further optimized the details of the generated images by parsing the input text. Lao et al. [18] disentangle the input text into two types latent code of content and style by a dual adversarial inference mechanism to obtain comprehensive semantic information. SD-GAN [22] further improved the image synthesis quality by exploring the consistency between the input sentences. RiFeGAN [19] and RiFeGAN2 [23] used multiple sentences as input to provide rich semantic information to the generator. However, these two methods only achieved promising performance on CUB-200 [24] and Oxford-102 [25]. LeicaGAN [20] proposed text-visual co-embeddings to translate input text to corresponding visual features. DAE-GAN [6] adopts aspect information of text to enhance the details of synthesized images. Park et al. [21] proposed new benchmarks to perform a systematic study of the model’s generalization performance for novel word compositions.

2) Modeling the Generators and Discriminators: Many studies [22], [26], [27], [28], [29] have focused on developing delicate discriminators to generate images that are well-matched to the input text. SEGAN [27] combined conditional discriminators with the Siamese networks to achieve fine-grained text-to-image generation. Text-SeGAN [28] can generate diverse images that are semantically relevant to the input text by adding a semantic classifier to the discriminator. ControlGAN [29] introduced fine-grained word semantic as conditional information in the discriminator by an attention mechanism to push for the text-image semantic alignment. TIME [26] adopts a image caption model as discriminator to improve semantic consistency.

However, to generate high quality images, these methods generally employed stacked generative adversarial networks, which further increases the computational consumption and reduces training efficiency. DTF-GAN [30] and DF-GAN [5] only employed a pair of generator and discriminator to synthesize high-quality and semantically consistent images. CKD [31] designed a hierarchical knowledge distillation paradigm to extract image semantic which injects into the image generation process.

Another interesting line of research has explored the generation of faces and food. CookGAN [32] completed the step-by-step generation of food in an interactive way. ChefGAN [33] performed recipe-to-food generation by a joint image-recipe embedding model. SEA-T2F [34] is proposed for text-to-face generation, where multiple captions are used as inputs to improve the semantic consistency of the generated faces. TediGAN [35] proposed a novel framework for multi-modal face generation and manipulation with textual descriptions. Zhou and Shimada [36] employed a pre-trained...
transformer-based BERT model and a StyleGAN encoder to synthesize the realistic face images.

3) Generating Diverse Objects: Although these aforementioned approaches have achieved significant progress on some easy datasets which only have one object per image (e.g., bird, flower, face, and food). The context-rich image synthesis which has complex objects is still challenging. CPGAN [4] focused on parsing the content of both the input text and the synthesized image thoroughly to model the text-to-image consistency of complex image. Huang et al. [37] designed a unified image-and-text framework to jointly study image-to-text and text-to-image generations. XMC-GAN [38] efficiently performs cross-modal translation by maximizing the mutual information between image and text. Since the semantic ambiguity of the input text and the image diversity, it is difficult for these methods to generate images with correct spatial relationships of objects.

4) Leveraging Transformer Instead of CNNs as the Backbone: In recent years, thanks to the rapid development of VIT [39], several works have attempted to solve the text-to-image generation problem by training directly on large-scale datasets in an end-to-end manner, such as DALL-E [40] and CogView [41]. The latest works of Imagen [42], DALL-E2 [43] and GLIDE [44] also demonstrate the potential of diffusion models for text-to-image synthesis. To achieve good visualization results, these methods not only have a huge number of trainable parameters but also require a gigantic scale of labeled data for training. In other words, they typically require hundreds of GPUs and weeks of training time to get the desired results, which limits further development in this field.

The difficulty of text-to-image synthesis is caused by the large differences between text and image. While significant gap between two modalities stems from the sparse and ambiguous semantic information of text compared with the corresponding image. Some text-image matching works attempt to mitigate this problem using hierarchical feature aggregation [45], teacher-student learning framework [46], and region reinforcement networks [47]. For text-to-image synthesis, Bridge-GAN [48] built an implicit feature space as a bridge to associate text and image. Some cross-modal retrieval works [49], [50] also explored how to learn the common latent space for different modalities. However, the implicit bridge does not work well in context-rich image synthesis which has complex objects. Hence, using the layout as an explicit bridge between text and image not only alleviates the difficulty of the task but also generates images with a reasonable space distribution of objects.

B. Text-to-Image Synthesis With Image Layout

Image layout captures the spatial distribution patterns of objects in the scene and thus has been taken into consideration by a few recent text-to-image works. Hong et al. [51] first developed a text-to-layout-to-image framework to generate images with reasonable spatial distributions. Following this work, Obj-GANs [7] further improved the layout synthesis quality by incorporating fine-grained word embeddings. By employing the layout synthesis from Obj-GAN [7], OP-GAN [52] proposed a new spatial location related generator to achieve text-to-image synthesis. R-GAN [53] adopts a similar method of text-to-layout generation as Obj-GAN and further incorporates scene graph information to synthesize more reasonable layouts. Besides, Text2Scene [54] focused on text-to-layout synthesis by designing a Seq-to-Seq framework. [55] attempted to synthesize layout using a plain transformer structure in an autoregressive manner, but it was unable to have semantic control over the generated layout. Among these recent works, Obj-GAN is a representative that performs well in many simple scenarios. However, modeling complex spatial relationship between objects still remains challenging.

III. Method

Given a textual description, we aim to generate a high-quality layout that indicates the spatial distribution of objects in the image to be synthesized. The generated layout, which is expected to bridge the gap between the input text and the synthesized image, is further leveraged to guide the image synthesis and thereby yielding high-quality image for the given text. Thus, our method performs text-to-image synthesis in two stages: text-to-layout generation and layout-to-image synthesis. Figure 2 illustrates the overall structure of our method.

A. Text-to-Layout Generation

The layout is generated following an encoder-decoder framework by our proposed LayoutTransformer, which consists of two modules: Text Encoder and Layout Decoder. It first encodes the input text to learn latent representations for the entire text by its Text Encoder, then employs Layout Decoder to generate the image layout. We formulate the layout generating process as a sequential prediction of objects. As a result, the layout generation boils down to a sequence-to-sequence modeling task. Hence we design the text-to-layout generation by adopting the similar internal structure as Transformer [9] due to its powerful capability of sequence-to-sequence modeling.

1) Text Encoder: The Text Encoder of LayoutTransformer performs text encoding in the similar way as the encoder of classical Transformer, which takes a sequence of word embeddings as input and learns a corresponding sequence of representations with the same length as the input sequence:

$$\{s_1, \ldots, s_{T_i}\} = F_e(\{e_1, \ldots, e_{T_i}\}), \quad (1)$$

where $T_i$ is the length (in word) of the input text. $F_e$ denotes the transformation function performed by Text Encoder, which is composed of 6 identical basic layers. Each basic layer comprises a multi-head self-attention layer and a position-wise fully connected layer. The input word embedding at $t$-th step $e_t \in \mathbb{R}^D$ corresponds to the indexed vectorial representation in the embedding matrix $E \in \mathbb{R}^{D \times K}$ for a word dictionary containing $K$ words, which is learned together with the whole model.
Fig. 2. Architecture of the proposed method, which performs text-to-image synthesis in two stages: text-to-layout Generation by the proposed LayoutTransformer and layout-to-image synthesis by the proposed Text-Aligned Layout-to-Image Synthesizer.

![Architecture Diagram](image)

Fig. 3. Visualization of synthesized multi-caption masking.

- **a) Incorporation of multi-caption input:** The captions for a same image are highly diverse, it means that naively using these captions as unique inputs will generate unstable layouts. Hence, we concatenate them together as input to form a more comprehensive description. To avoid semantic inter-pollution between different initial captions, we perform masking to ensure that each word in the concatenated caption can only attend to words in the same initial caption while other words are masked out to be invisible as shown in Figure 3. Specifically, given three initial input captions \( \{s_1, s_2, s_3\} \), we construct corresponding masks \( \{m_{mask1}, m_{mask2}, m_{mask3}\} \) for each of initial input captions. When we encode the word in \( s_i \), we perform element-wise multiplication between the concatenated input text \( s_{concat} \) and the corresponding mask \( m_{maski} \) to ensure that each word in the concatenated caption can only attend to words in the same initial caption while other words are masked out to be invisible. During the training stage, we process these sentences simultaneously.

2) **Layout Decoder:** The obtained latent representations \( S = \{s_1, \ldots, s_T\} \) from Text Encoder are then decoded by Layout Decoder of our LayoutTransformer to generate the image layout. The key difference between our method and Transformer is that the output of our Layout Decoder is image layout instead of a text sequence. Thus, how to formulate the layout generating process is crucial to the design of Layout Decoder.

- **a) Sequential object prediction:** The generated layout is required to show the object distribution in the image to be synthesized. Thus generating an image layout involves predicting the bounding box (including both location and size) and the class label for each potential object. Besides, the dependencies between objects, such as spatial or semantic associations, should be taken into account during the layout generation. Hence the prediction of current object should depend on the previously predicted objects rather than predicting objects independently. Based on these two considerations, we design Layout Decoder to predict the objects sequentially to generate the image layout progressively, i.e., predicting one object per step. Accordingly, we adopt similar sequential decoding structure as Transformer for Layout Decoder, which also employs 6 basic layers consisting of multi-head attentions and fully connected layers for decoding. Formally, Layout Decoder performs decoding to learn the hidden state \( h_t \) at \( t \)-th time step prepared for output by:

\[
h_t = F_d(S, O_{1:t-1}).
\]  

Herein, \( O_{1:t-1} \) denotes the previously predicted results and \( F_d \) refers to the transformation function by the decoding process of Layout Decoder.

- **b) Joint classification of object category and position:** For prediction of the each object in the layout, Layout Decoder needs to predict not only the class label of the object, but the size and location of the bounding box. We find that although the distribution of objects in a layout is highly diverse, the location of objects is directly related to their category. Hence, predicting the class label of objects and the bounding boxes jointly can facilitate the predicting precision. Specifically, we partition the layout area into a \( S \times S \) grid, and then predict the class label and the grid cell containing the object center jointly each time we predict an object. Consequently, we formulate it as a classification problem with total \( S \times S \times C \) classes, where \( C \) is total category number of objects. It is equivalent to evaluating all possibilities of combination of the category and the location (in grid) for the object to be predicted and then finding the optimal option, namely the grid cell with highest confidence that contains the most probable category of object. Formally, Layout Decoder performs classification at
t-th time step based on the hidden states \( \mathbf{h}_t \) by:

\[
v_t = \text{argmax} \ (\mathbf{p}_t)_{j \in [1, S \times S \times C]}, \quad \mathbf{p}_t = \text{Softmax}(\mathbf{Mh}_t),
\]

where \( \mathbf{M} \) denotes a linear transformation. \( \mathbf{p}_t \in \mathbb{R}^{S \times S \times C} \) is calculated probabilities for total \( S \times S \times C \) classification entries. \( v_t \) is the entry with the maximal probability, from which we can infer the corresponding class label \( c_t \) of the predicted object in this step as well as the grid cell \( < G_t^x, G_t^y > \) containing the object center. Note that \( \mathbf{O}_t \) in Equation 2 is represented in one-hot encoding of \( v_t \).

Finally, we directly regress vertex coordinates of the bounding box based on the grid cell \( < G_t^x, G_t^y > \). The classification performed in Equation 3 predicts the grid cell that the object center fall in, which is a coarse object position. To predict the precise location of bounding box center within the localized grid cell as well as the size of the bounding box, we performs fine-grained regression by a regression head (denoted as \( \mathcal{F}_{\text{reg}} \)) consisting of three fully connected layers as well as activation function ReLU to predict the state of bounding box \( \mathbf{f}_t = \{f_t^x, f_t^y, f_t^w, f_t^h\} \) at t-th step:

\[
\{f_t^x, f_t^y, f_t^w, f_t^h\} = \mathcal{F}_{\text{reg}}(\mathbf{h}_t),
\]

where \( f_t^x, f_t^y \) are coordinates of bounding box center within the grid cell \( < G_t^x, G_t^y > \), and \( f_t^w, f_t^h \) jointly indicate the bounding box size of the predicted object. Such coarse-to-precise localization strategy decomposes the complex localization problem into two much easier prediction steps.

A prominent benefit of formulating the prediction of both the category and position of the object as one integrated classification task is that such mechanism enables the model to achieve more accurate estimation of the object state than predicting the category and position separately. As a result, such mechanism facilitates the modeling of dependencies between predicted objects in sequential object prediction shown in Equation 2, which is validated in experiments in Section IV-C1.

3) Parameter Learning: The whole model of LayoutTransformer for text-to-layout generation, including Text Encoder and Layout Decoder, is trained under the supervision of two types of losses: the classification loss \( \mathcal{L}_{\text{cls}} \) for prediction in Equation 3 and regression loss \( \mathcal{L}_{\text{reg}} \) for bounding boxes regression:

\[
\mathcal{L}_{\text{layout}} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{reg}}, \quad \mathcal{L}_{\text{cls}} = \sum_{n=1}^{N} \sum_{t=1}^{T_n} \mathcal{L}_{\text{CE}}(\mathbf{p}_t, \mathbf{p}_t),
\]

where \( \mathcal{L}_{\text{CE}} \) is cross entropy loss and \( \mathcal{L}_{\text{reg}} \) is the regression loss of bounding box adopted in YOLO-V3 [56]. Here \( T_n^{(n)} \) denotes the predicted sequence length, namely the number of objects, for the \( n \)-th training sample composed of a textual description and a paired image. \( \lambda \) is the hyper-parameter to balance two losses.

B. Layout-to-Image Synthesis

The generated layout is further leveraged to guide the image synthesis, thereby synthesizing an image that is consistent with both the generated layout and the textual description. As shown in Figure 2, we draw on generative structure of LostGAN_V2 [10] and build our image synthesizer upon it. Since the goal of LostGAN_V2 is to synthesize image from sole layout without textual description, we incorporate the textual description into our image synthesizer to ensure the textual-visual consistency.

1) Text-Aligned Layout-to-Image Synthesizer: Text-Aligned Layout-to-Image Synthesizer (TALIS) is designed to generate the output image, taking both the given textual description and the generated layout as input. As illustrated in Figure 2, its basic backbone is the generative network \( \mathbf{G} \), which consists of one fully connected layer and five residual building blocks (ResBlocks). The image synthesis is performed by two core operations. 1) Basis image generation: the generative network \( \mathbf{G} \) takes an initial vectorial embedding \( \mathbf{q}_{\text{img}} \) to generate the basis of the whole image. 2) Guidance of layout: a feature normalization module termed ISLA-Norm of LostGAN_V2 is employed to steer the generating process of \( \mathbf{G} \) based on the generated layout to achieve the layout-image consistency. The semantics of the input text are incorporated into both operations.

Since the Text Encoder designed for text-to-layout generation aims to learn textual embeddings that are particularly used for predicting the layout, it is not well suitable for layout-to-image synthesis in which textual-visual semantic alignment is crucial. Thus, we learn an individual text encoder to extract textual semantics for describing each object in the layout. The proposed text encoder, referred to as Object-Oriented Text Encoder, utilizes the encoder of transformer \( \mathcal{E}_T \) to learn the hidden representations for each word in the text:

\[
\{z_1, \ldots, z_T\} = \mathcal{E}_T((\mathbf{e}_1, \ldots, \mathbf{e}_T)),
\]

where word embedding at t-th step \( \mathbf{e}_t \in \mathbb{R}^D \) is the vectorial representation indexed from a learned embedding matrix \( \mathbf{E} \in \mathbb{R}^{D \times K} \).

Basis image generation. The global representation \( \bar{\mathbf{z}} \) for the whole textual description is obtained by averaging word features \( \{z_1, \ldots, z_T\} \). It is concatenated with a noise embedding \( \mathbf{q}_{\text{img}} \) sampled from a normal distribution to form the initial embeddings \( \mathbf{q}_{\text{img}} \) for \( \mathbf{G} \):

\[
\mathbf{q}_{\text{img}} = \text{Concat}(\bar{\mathbf{z}}, \mathbf{q}_{\text{img}}).
\]

Guidance of layout. The learned textual representations are also incorporated into the feature normalization module ISLA-Norm to manipulate the generating process of \( \mathbf{G} \). Specifically, we extract the related semantics for each object in the generated layout by attending to each word in the text and measuring their compatibility to the object category. For instance, the textual semantics \( \mathbf{a}_k \) for \( k \)-th object in the layout with the class label \( c_k \) are extracted by:

\[
\mathbf{a}_k = \sum_{i=1}^{T_l} w_i \mathbf{z}_i, \quad w_l = \frac{\exp(\mathbf{z}_l^\top \mathbf{U}[c_k,:])}{\sum_{i=1}^{T_l} \exp(\mathbf{z}_i^\top \mathbf{U}[c_k,:])},
\]

where \( \mathbf{U} \in \mathbb{R}^{D \times d_c} \) is a learned embedding matrix for all C categories. \( w_l \) denotes the compatibility between the object category \( c_k \) and the t-th word. In this way we can learn
the textual semantics for all $T_o$ objects in the layout $A = \{a_1, \ldots, a_{T_o}\}$. The obtained textual semantics are concatenated with the category embeddings and a noise embedding together for each object to form the joint input vector for ISLA-Norm:

$$I_k = \text{Concat}(a_k, U[c_k, :], n^{k}_{\text{obj}}), \quad k = 1, \ldots, T_o,$$

where $n^{k}_{\text{obj}}$ is a sampled noise embedding for $k$-th object. ISLA-Norm utilizes the obtained layout information $L = \{I_1, \ldots, I_{T_f}\}$ to perform feature normalization for feature maps in each ResBlock of the generative network $G$. The details of ISLA-Norm are presented in LostGAN_V2 [10]. The generative network $G$ is trained following the training procedures of LostGAN_V2 while the proposed Object-Oriented Text Encoder is pre-trained individually, which is explicated in the following subsection.

2) Learning the Textual-Visual Semantic Alignment: One key point that affects the performance of the proposed Text-Aligned Layout-to-Image Synthesizer (TALIS) is that the learned textual semantics for each object by Object-Oriented Text Encoder should be aligned well with the corresponding visual semantics corresponding to the synthesized image for the same object. To this end, we pre-train Object-Oriented Text Encoder individually to push for the textual-visual semantic alignment. Given a text-image pair $(X, I)$ for training, we extract the visual features for each object in the image by a pre-trained VGG net [57].

$$B, \tilde{b} = \text{ROI}(\text{VGG}(I)),$$

where $B = \{b_1, \ldots, b_{T_f}\}$ is the visual features for total $T_o$ object and $\tilde{b}$ is the visual feature for the whole image. Herein ROI is the ROI-align layer appended after the feature maps of VGG net for cropping features for objects. We aim to maximize the consistency between the visual features $B$ and the textual semantics $A$ obtained in Equation 8 for all objects in the image. Thus we define the consistency score between them based on Cosine distance:

$$S_{\text{obj}}(X, I) = \log(\sum_{k=1}^{T_o} \exp(\frac{b_k a_k}{\|b_k\| \|a_k\|})).$$

Similarly, we also measure the global consistency score between the global textual semantics $\tilde{e}$ and the global image features $\tilde{b}$:

$$S_{\text{img}}(X, I) = \frac{\tilde{b} \tilde{e}}{\|\tilde{b}\| \|\tilde{e}\|}.$$  

Finally, we train the Object-Oriented Text Encoder by a contrastive loss,

$$L_{\text{con}} = L_{\text{obj}} + L_{\text{img}},$$

$$L_{\text{obj}} = -\log \frac{\exp(S_{\text{obj}}(X_k, I_k))}{\sum_{i=1}^{J} \exp(S_{\text{obj}}(X_i, I_k))},$$

$$L_{\text{img}} = -\log \frac{\exp(S_{\text{img}}(X_k, I_k))}{\sum_{i=1}^{J} \exp(S_{\text{img}}(X_i, I_k))},$$

where $J$ is the batch size in the training stage.

IV. EXPERIMENT

A. Experimental Setup

1) Dataset: We use three datasets to evaluate our method. 1) COCO [58], 2) COCO-stuff [59] and 3) LN-COCO. COCO [58] dataset is commonly used for text-to-image synthesis. Each image has 5 corresponding textual captions. Compared with the 80 instance categories (bicycle, bus, etc.) in COCO dataset, COCO-stuff [59] is a more challenging dataset containing additional 91 stuff categories (hill, grass, etc.). LN-COCO has the same set of images as COCO-stuff whilst each image is described by a narrative obtained by Localized Narratives [60]. Narratives are four times longer than COCO and COCO-stuff captions on average and thus are more informative, which makes text-to-layout synthesis on LN-COCO much more challenging than on COCO and COCO-stuff.

2) Implementation Details: Since the gradient cannot be back-propagated through the generated layout, we train the LayoutTransformer for text-to-layout generation and the TALIS for layout-to-image synthesis separately. Both networks are trained from scratch. In our experiment, the grid size $S$ of the Layout Transformer is set to be 7. The hyper-parameter $\lambda$ in Equation 5 is set to be 2.0. All networks are trained using Adam [61].

B. Evaluation Metrics

We employ existing standard metrics for the evaluation of synthesized images, and propose a new metric for the evaluation of text-to-layout generation due to the lack of effective metrics.

1) Metrics for Evaluating Synthesized Images: We adopt the five standard metrics for evaluating the quality of synthesized images: Inception Score [62] (IS), Frechet Inception Distance [63] (FID), R-precision [2]. ClipScore [37] and SOA [52]. IS focuses on evaluating the authenticity and diversity of synthesized images while FID measures the distribution distance between synthesized images and the corresponding groundtruth images. R-precision usually uses a vision-and-language CLIP [64] model as the retrieval model to measure the semantic consistency between the textual description and the synthesized image. However, the retrieval models used in previous works are inconsistent, which can lead to biased behavior. In this paper, we uniformly use a powerful vision-and-language CLIP [64] model as the retrieval model to accomplish the quantitative evaluation of semantic consistency. Compared to R-precision, which evaluates semantic similarity by ranking retrieval results, ClipScore adopts the CLIP model to directly calculate the cosine similarity of text-image pairs. In this paper, we take Clipscore as a fine-grained semantic consistency evaluation metric to complement R-precision. SOA adopts a pre-trained object detection network to measure whether the objects specifically mentioned in the caption are recognizable in the generated images. Specifically, it includes two sub-metrics: SOA-C (average recall w.r.t. object category) and SOA-I (average recall w.r.t. image sample).
2) Metrics for Evaluating Generated Layouts: Docsim [65] was proposed to evaluate the quality of generated layouts. However, Docsim only measures the absolute errors of bounding box positions between the generated layout and the ground truth while neglecting the mutual spatial relationships between bounding boxes in a layout. Thus, we propose a new metric specifically for evaluating the generated layouts called Layout Quality Score (LQS), which considers both the absolute distribution errors of bounding boxes and the mutual spatial relationships between them. Concretely, we consider four measurements as follows.

**Label Recall (LR),** which measures the recall rate of the object categories in the generated layout referring to the groundtruth layout.

**Label Precision (LP),** measuring the precision of the object categories in the generated layout compared to the groundtruth layout. Formally, We denote the label set of the objects in a generated layout containing \( T_o \) objects as \( \mathbf{d} = \{d_1, d_2, \ldots, d_{T_o}\} \). Similarly, the label set in the corresponding groundtruth layout is denoted as \( \mathbf{d} = \{d_1, d_2, \ldots, d_{T_o}\} \). LR and LP are defined as:

\[
\text{LR} = \frac{|\mathbf{d} \cap \hat{\mathbf{d}}|}{|\mathbf{d}|}, \quad \text{LP} = \frac{|\mathbf{d} \cap \hat{\mathbf{d}}|}{|\hat{\mathbf{d}}|}.
\]

**Location Consistency (LC),** which measures the location consistency between the objects in the generated layout and those in the groundtruth layout with the same category. We consider two kinds of consistency: 1) Absolute Location Consistency (ALC) which is defined as the distance between the predicted object centers to the corresponding groundtruth object centers and 2) Relative Location Consistency (RLC) which measures the consistency of pairwise distance between two objects, comparing the generated layout with the groundtruth layout. Denoting \( \mathbf{m} = \mathbf{d} \cap \hat{\mathbf{d}} \), they are defined as:

\[
\text{ALC} = \sum_{\mathbf{m} \in \mathbf{d}} \| \mathbf{g}_o - \hat{\mathbf{g}}_o \|_2 / |\mathbf{m}|,
\]

\[
\text{RLC} = \sum_{\mathbf{m} \in \mathbf{d}} \sum_{i \neq j} \frac{\| \mathbf{r}_{i,j} - \hat{\mathbf{r}}_{i,j} \|_2}{|\mathbf{m}| \times (|\mathbf{m}| - 1)},
\]

where \( \mathbf{g}_o \) and \( \hat{\mathbf{g}}_o \) are the coordinates of the \( o \)-th object in intersection set \( \mathbf{m} \) (correctly predicted objects). \( \mathbf{r}_{i,j} = \mathbf{g}_i - \mathbf{g}_j \) is the relative distance between the \( i \)-th object and the \( j \)-th object. However, the values of ALC and RLC are directly related to the whole image area \( u_s \) (the larger \( u_s \), the larger the values of ALC and RLC). To remove the effect of \( u_s \) on the value of ALC and RLC, we adopt the Gaussian kernel function (parameterized by \( \sigma_l \)) to calculate smoothing values. Finally, we define Location Consistency (LC) by combining these two measurements:

\[
\text{LC} = \gamma_l \text{exp}(-\text{ALC}/2\sigma_l^2) + (1 - \gamma_l)\text{exp}(-\text{RLC}/2\sigma_l^2).
\]

(16)

When calculating LC, the generated coordinates and the annotated ones of the \( o \)-th object should be matched. However, the objects are generated in random order at the first stage. Hence, we first match the bounding boxes of the same category. When the same category has several objects, we select the group with the lowest ALC from all combinations.

**Area Consistency (AC),** which measures the consistency between the area of the predicted object bounding boxes and the area of groundtruth object bounding boxes. Similar to the definition of LC, AC is also defined considering two types of measurements: 1) Absolute Area Consistency (AAC) measuring the difference between area of predicted objects and the corresponding groundtruth objects; 2) Relative Area Consistency (RAC) which measures the consistency of pairwise size relationship. Suppose the bounding box area for the \( i \)-th predicted object and the corresponding groundtruth object in \( \mathbf{m} = \mathbf{d} \cap \hat{\mathbf{d}} \) are \( \hat{u}_i \) and \( u_i \), then AAC and RAC are defined as:

\[
\text{AAC} = 1 - \frac{1}{|\mathbf{m}|} \sum_{\mathbf{m} \in \mathbf{d}} \frac{|\hat{u}_i - u_i|}{u_s},
\]

\[
\text{RAC} = \frac{\sum_{\mathbf{m} \in \mathbf{d}} \sum_{i \neq j} (1 - |1_{u_i > u_j} - 1_{\hat{u}_i > \hat{u}_j}|)}{|\mathbf{m}| \times (|\mathbf{m}| - 1)}.
\]

(17)
where $\mathbb{1}$ is the indicator function. Finally, we define Area Consistency (AC) by combining both AAC and RAC based on a Gaussian kernel function (with parameter $\sigma_3$):

$$\text{AC} = \gamma_{ac} \text{AAC} + (1 - \gamma_{ac}) \text{RAC}. \quad (18)$$

LR and LP are defined to measure the correctness of predicted object categories while LC and AC measures the quality of the bounding boxes including location and shape. Taking into account all four measurements, we calculate the Layout Quality Score ($\text{LQS} \in [0, 4]$) by:

$$\text{LQS} = \text{LR} + \text{LP} + \text{LC} + \text{AC}. \quad (19)$$

Although both LC and AC consist of absolute and relative parts. Since a text input can be consistent with multiple different layouts, relative consistency is more important than absolute consistency. Therefore, in this paper, we set both $\gamma_c$ and $\gamma_{ac}$ to 0.25. Subsequently, with $256 \times 256$ image area resolution, $\alpha_3$ is set to 80 to obtain smoothing values of LC. By adjusting these hyperparameters, LQS can perform...
TABLE I
PERFORMANCE OF DIFFERENT METHODS FOR TEXT-TO- LAYOUT GENERATION ON THREE DIFFERENT DATASETS

| Dataset   | Model         | LR ↑ | LP ↑ | ALC ↓ | RLC ↓ | LC ↑ | AAC ↑ | RAC ↑ | AC ↑ | LQS ↑ | IS ↑ | FID ↓ | DS ↓ | CAS ↑ |
|-----------|---------------|------|------|-------|-------|------|-------|-------|------|-------|------|-------|------|-------|
| COCO      | Obj-GANs      | 46.58% | 68.37% | 63.449 | 97.962 | 0.5370 | 0.8268 | 0.6859 | 0.7211 | 2.4076 | -   | -     | -   | -     |
|           | Text2Scene    | 49.16% | 61.24% | 91.304 | 102.408 | 0.4609 | 0.8634 | 0.6809 | 0.7265 | 2.2914 | -   | -     | -   | -     |
|           | Ours          | 47.44% | 75.75% | 57.2088 | 88.5594 | 0.6000 | 0.8481 | 0.6964 | 0.7343 | 2.5662 | -   | -     | -   | -     |
| COCO-stuff| Obj-GANs      | 33.36% | 48.09% | 54.4306 | 77.3151 | 0.6685 | 0.7915 | 0.6144 | 0.6587 | 2.1417 | 10.75 | 88.28 | 0.7288 | 0.3765 |
|           | Text2Scene    | 44.30% | 48.33% | 64.0896 | 65.1006 | 0.7200 | 0.7719 | 0.6811 | 0.7038 | 2.3501 | 5.79 | 126.08 | 0.7149 | 0.2410 |
|           | Ours          | 47.06% | 62.03% | 43.6186 | 63.2825 | 0.7640 | 0.7849 | 0.6905 | 0.7141 | 2.5690 | 19.03 | 39.40 | 0.6944 | 0.6228 |
| LN-COCO   | Obj-GANs      | 27.65% | 41.11% | 54.5414 | 81.7542 | 0.6431 | 0.7899 | 0.5796 | 0.6322 | 1.9629 | 9.65 | 128.06 | 0.7165 | 0.4018 |
|           | Text2Scene    | 49.24% | 53.14% | 66.6123 | 68.2388 | 0.4981 | 0.7786 | 0.6649 | 0.7098 | 2.4319 | 5.69 | 100.63 | 0.7110 | 0.2664 |
|           | Ours          | 46.61% | 62.70% | 41.6081 | 69.5552 | 0.7862 | 0.7795 | 0.6893 | 0.7118 | 2.5912 | 19.27 | 40.04 | 0.6894 | 0.6719 |

C. Evaluation of Generated Layouts

In this section we evaluate the proposed LayoutTransformer for text-to-layout Generation both quantitatively and qualitatively. We first conduct ablation study to evaluate the effectiveness of each module in LayoutTransformer. Next, we compare our LayoutTransformer with other methods for text-to-layout generation. Finally, we visualize the synthesized layouts for several examples to gain more insight into text-to-layout generation. Finally, we visualize the synthesized layouts for several examples to gain more insight into text-to-layout generation.

1) Ablation Study:

a) Effect of sequential object prediction: To investigate the effect of sequential object prediction mechanism, we evaluate the variant of LayoutTransformer (Ours), termed as Ours w/o SP, which predicts the objects in parallel. Thus the dependencies between objects are not modeled. As a result, layouts generated by Ours w/o SP have redundant bounding boxes (‘tree’, ‘grass’ and ‘snow’) as shown in Figure 5. The quantitative results in Figure 4 also show that Ours w/o SP performs much worse than LayoutTransformer (especially on LP). The sequential object prediction enables the LayoutTransformer to perceive the previously generated objects and effectively reduces the repeated generation of objects of the same category based on the semantics of the input text.

b) Effect of joint classification of object category and position: Inspired by the strong correlation between object category and position, joint classification can perform better location estimation of object. To investigate the effect of such design, we evaluate the performance of the variant of our model that predicts the object category and position (in grid) separately, denoted as Ours w/o JC. Figure 4 shows that the performance of Ours w/o JC is distinctly lower than original LayoutTransformer (especially on LP and RLC). Compared to the full LayoutTransformer, interactions between objects (‘person/kite’ and ‘person/skis’) are incorrect as shown in Figure 5, which validates that joint prediction can achieve more accurate estimation of spatial relationships between objects.

c) Incorporating multi-caption input vs single-caption input: To investigate the effect of incorporating multi-caption input, we evaluate the variant of LayoutTransformer, termed as Ours w/o M, which only uses single caption as input in the training stage. Due to the diversity of textual expressions, the text encoder of Ours w/o M cannot precisely extract the semantic information in the input text. Figure 4 shows that Ours w/o M also performs worse than the original version on LP, RLC and AAC. Besides, Figure 5 also manifests that the generated layouts by Ours w/o M cannot learn reasonable layout between objects.

d) Effect of hyper-parameter λ in equation (5): To investigate the effect of hyper-parameter, we adopted different values of λ in the training process and the experiment results are shown in Table II. We found that both too high and too low λ can degrade the performance of the model. This is due to the fact that λcls guides the model for coarse-grained location prediction and λreg steers the model for fine-grained position prediction when the two losses are in balance to achieve reasonable prediction results. Based on the results of the ablation study, we set λ to 2.0 in the subsequent experiments.

2) Comparison With State-of-the-Art Methods:

a) Quantitative evaluation: Table I presents the experimental results of our LayoutTransformer and other state-of-the-art methods for text-to-layout generation: Text2Scene [54]
TABLE III
QUANTITATIVE RESULTS OF DIFFERENT TEXT-TO-IMAGE SYNTHESIS MODELS ON COCO DATASET

| Model        | IS ↑ | IS w/o DAMSM ↑ | FID ↓ | R-precision ↑ | ClipScore ↑ | SOA-C ↑ | SOA-A ↑ | Rank-1 ratio ↑ |
|--------------|------|-----------------|-------|---------------|-------------|---------|---------|---------------|
| AttnGAN [2]  | 23.61| 12.22           | 33.10 | 53.36         | 68.40       | 25.88   | 39.01   | 3.36%         |
| DM-GAN [3]   | 32.32| 11.65           | 27.34 | 69.10         | 72.30       | 33.44   | 48.03   | 3.38%         |
| DF-GAN [5]   | -    | 17.25           | 19.32 | 39.06         | 65.34       | 17.77   | 30.42   | 10.88%        |
| DAE-GAN [6]  | 33.11| 10.57           | 56.85 | 66.92         | 71.31       | 30.45   | 45.08   | 7.12%         |
| CSM-GAN [67] | 26.77| -               | 33.48 | -             | -           | -       | -       | -             |
| CGL-GAN [68] | 13.62| -               | 37.12 | -             | -           | -       | -       | -             |
| KD-GAN [69]  | 34.01| -               | 23.92 | -             | -           | -       | -       | -             |
| DM-GAN-MDD [70]| 34.46| -               | 24.30 | -             | -           | -       | -       | -             |
| Bridge-GAN [48]| 16.40| -               | -     | -             | -           | -       | -       | -             |
| TIME [26]    | 30.85| -               | 31.14 | -             | -           | 32.78   | -       | -             |
| Huang [37]   | -    | 25.31           | 29.90 | 69.90         | 77.20       | 38.12   | 53.31   | -             |
| VQ Diffusion-BA [71]| -    | -               | 19.75 | -             | -           | -       | -       | -             |
| Obj-GANs [7] | 24.09| 11.34           | 36.52 | 60.05         | 70.26       | 27.14   | 41.24   | 5.64%         |
| OP-GAN [52]  | 27.88| 10.32           | 24.07 | 63.81         | 71.14       | 35.85   | 50.47   | 6.76%         |
| R-GAN [53]   | -    | -               | 24.60 | -             | -           | -       | -       | -             |
| Ours         | 38.43| 22.95           | 15.85 | 78.40         | 78.82       | 38.59   | 57.05   | 54.44%        |

Fig. 7. Visualization of synthesized images by different models with (bottom row) and without DAMSM (top row) respectively.

and Obj-GANs [7] on three datasets. In addition to LR and AAC, our LayoutTransformer achieves the best performance in other metrics and outperforms other methods significantly in terms of LP, LC, IS, and FID, which is owing to joint contributions from all three modules we proposed. Particularly, the enhancement of LC shows that our method can generate layouts with more correct spatial relationships. The performance of IS, FID, DS, and CAS also prove that the layout generated by our method can be used to synthesize high quality images containing more recognizable objects.

b) Qualitative evaluation: The qualitative comparisons in Figure 6 reveal that our method is able to generate more reasonable layouts between objects than other methods on three different datasets. Other methods either fail to predict correct object categories, or cannot correctly establish spatial relationships between different objects.

In the COCO dataset, we observe that Obj-GANs and Text2Scene either fail to predict correct object categories, or cannot correctly establish spatial relationships between different objects. The results in the first and second rows of Figure 6 show that our model can synthesize layouts that reflect subtle semantic differences of input captions (riding the same motorcycle vs two different motorcycles). The results in the third row of Figure 6 also demonstrate that our method is not simply overfitting the dataset but can comprehensively understand the input caption and generate a reasonable layout.

The COCO-stuff dataset contains more categories which increases the challenge of text-to-layout synthesis but our model still can generate reasonable layouts. In the LN-COCO dataset, the longer captions further increase the challenge of semantic parsing of text encoder. Obj-GANs and Text2Scene lack the ability to parse semantics of long input captions to correctly predict object categories as illustrated in Figure 6. Since the outstanding long-range semantic parsing capability of our text encoder, our model is still able to generate more reasonable layouts.

D. Evaluation of Synthesized Images

Next we compare our method with other state-of-the-art methods for text-to-image synthesis.

1) Quantitative Evaluation: Following previous work, we report validation results by generating images for 30,000 random captions. Table III reports the experimental results in terms of IS, FID, R-precision, ClipScore, SOA-C, and

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 9. Qualitative comparison between our model with other state-of-the-art models for text-to-image synthesis.

SOA-I of different methods for text-to-image synthesis on the COCO dataset. Here we present two versions of performance in IS: 1) using DAMSM [2] (IS), which is a loss function for pushing for the text-to-image alignment and 2) performance without using DAMSM (IS w/o DAMSM). We report these two versions of performance since DAMSM has a substantial impact and bias towards IS score due to the same pre-trained Inception-V3 [72] module for image encoding employed by both DAMSM and IS calculation. To investigate whether DAMSM can improve the quality of the synthesized images for a same model, we visualize the synthesized images by different models, with and without DAMSM respectively in Figure 7. We observe that DAMSM degenerates the performance for almost all models, probably because DAMSM leads to the overfitting on IS score. The results in Table III show that our model achieves the second best performance on IS w/o DAMSM and best performance on other metrics and outperforms other methods significantly in terms of FID and R-precision which is owing to the guidance of layout in text-to-image synthesis.

2) Human Evaluation: As a complement to the standard evaluation metrics, we also perform human evaluation to compare our method with other state-of-the-art methods. Specifically, we randomly select 50 samples from the test set and ask human subjects to compare the quality of the synthesized images by different methods and vote for the
best one, considering both the text-image consistency and the quality of object layout. We calculate the rank-1 ratio for each method as the metric as shown in Table III. Our method achieves 54.44% votes from collected 50 human subjects, which outperforms other methods significantly.

3) Qualitative Evaluation: To have a qualitative comparison, we visualize the synthesized images by our models and other state-of-the-art models in Figure 9. Note that Obj-GANs and our method synthesize images via generating intermediate layout while other methods perform text-to-image synthesis directly. The visualized results clearly show that the synthesized images by our method are much better than the results of other methods, in terms of both image quality and the reasonability of object layouts. Specifically, the object locations in the second (lake, mountain, and cattle) and fourth row (stove and sink) of the Figure 9 show that the images synthesized by our method have a more reasonable layout. Besides, the objects in the images generated by our method also have a higher authenticity than other models (toilet, giraffe, and train).

We also compare our results with XMC-GAN [38] in Figure 8. XMC-GAN effectively improves the quality of the synthesized image by using cross-modal contrastive Learning strategy on the task of text-to-image generation. Since it does not release the pre-trained model and is difficult to reproduce results in their paper. Therefore, we only perform a qualitative evaluation of XMC-GAN. Compared to XMC-GAN which requires training 1000 epochs on a 32-core Pod slice of Google Cloud TPU v3 devices, our method only requires training 150 epochs on 4 Tesla V100 GPUs. However, our approach also has achieved competitive performance as shown in Figure 8.

E. Investigation on Effectiveness of TALIS

Compared to LostGAN_V2, our TALIS learns the textual-visual semantic alignment between the input text and the synthesized image by the proposed Object-Oriented Text Encoder. We perform T-sne visualization of word features extracted by the Object-Oriented Text Encoder. As shown in the Fig 11, the distribution of the semantics of the words belonging to the same category in different sentences in the feature space is shown by the dots of the same color. Obviously, the encoder can map words of the same category to similar positions in the feature space.

Table IV also reports the results of the quantitative comparison. Our method outperforms LostGAN_V2 in all four metrics and advances the FID significantly (from 39.40 to 35.05). Besides, we also visualize an example in Figure 10, in which the semantics of the four textual descriptions are similar but not exactly the same. Our model is able to synthesize images that reflect these subtle semantic differences.

F. Discussion of Limitations

While our method can generate plausible layouts in realistic scenarios given textual descriptions, the layout-to-image synthesis for our method remains a challenging problem as shown in Figure 12. This particularly results from quite diverse appearances for some categories of objects in COCO dataset. We will continue to investigate this problem in the future work.

| Model         | IS ↑ | PDD ↓ | DS ↓ | CAS ↑ |
|---------------|------|-------|------|-------|
| LostGAN_V2*   | 19.03| 39.40 | 0.6944| 0.6228|
| Ours          | 20.35| 35.05 | 0.6920| 0.6314|

Fig. 10. Synthesized images by our model given four close-semantic descriptions.

Fig. 11. T-sne visualization of word features of different categories.
G. Future Works

In the future, we aim to extend the application of the proposed method to image generation and video generation. To achieve this, we intend to leverage a large dataset of text-image pairs and a state-of-the-art object detection method to complete the optimization of text-to-layout synthesis and layout-to-image generation. Specifically, we plan to use the object detection method to obtain layout annotations of the images in the dataset. The rich layout distribution in the training dataset, combined with the powerful modeling capabilities of LayoutTransformer, can generate even more complex layouts. Furthermore, the richer image distribution will enhance the appearance of objects in the generated images. For video generation, we will limit the speed at which objects move between frames to increase the coherence of the synthesized video.

V. Conclusion

In this paper we perform text-to-image synthesis by generating an intermediate layout to bridge the input text and the synthesized image. We have presented LayoutTransformer for text-to-layout generation, which formulates the layout generation as a task of sequential object prediction. Thus the relationships between objects can be learned by modeling the sequential dependencies between object predictions. The generated layout is further leveraged to guide image synthesis performed by the proposed layout-to-image synthesizer, which focuses on learning textual-visual semantic alignment per object in the layout. Extensive experiments validate the effectiveness of the proposed method.

REFERENCES

[1] H. Zhang et al., “StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 5907–5915.
[2] T. Xu et al., “AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1316–1324.
[3] M. Zhu, P. Pan, W. Chen, and Y. Yang, “DM-GAN: Dynamic memory generative adversarial networks for text-to-image synthesis,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 5802–5810.
[4] J. Liang, W. Pei, and F. Lu, “CPGAN: Content-parsing generative adversarial networks for text-to-image synthesis,” in Proc. ECCV, Cham, Switzerland: Springer, 2020, pp. 491–508.
[5] M. Tao, H. Tang, S. Wu, N. Sebe, F. Wu, and X.-Y. Jing, “DF-GAN: Deep fusion generative adversarial networks for text-to-image synthesis,” in Proc. CVPR, Aug. 2022, pp. 1–11.
[6] S. Ruan et al., “DAE-GAN: Dynamic aspect-aware GAN for text-to-image synthesis,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 13960–13969.
[7] W. Li et al., “Object-driven text-to-image synthesis via adversarial training,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 12174–12182.
[8] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” 2015, arXiv:1508.01991.
[9] A. Vaswani et al., “Attention is all you need,” 2017, arXiv:1706.03762.
[10] W. Sun and T. Wu, “Learning layout and style reconfigurable GANs for controllable image synthesis,” 2020, arXiv:2003.11571.
[11] I. J. Goodfellow et al., “Generative adversarial networks,” 2014, arXiv:1406.2661.
[12] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, “Generative adversarial text to image synthesis,” in Proc. ICML, 2016, pp. 1060–1069.
[13] T. Qiao, J. Zhang, D. Xu, and D. Tao, “MirrorGAN: Learning text-to-image generation by redescription,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 1505–1514.
[14] X. Shu, B. Xu, L. Zhang, and J. Tang, “Multi-granularity anchor-contrastive representation learning for semi-supervised skeleton-based action recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 6, pp. 7559–7576, Jun. 2023.
[15] X. Shu, J. Yang, R. Yu, and Y. Song, “Expansion-squeeze-excitation fusion network for elderly activity recognition,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 8, pp. 5281–5292, Aug. 2022.
[16] B. Xu, X. Shu, and Y. Song, “X-invariant contrastive augmentation and representation learning for semi-supervised skeleton-based action recognition,” IEEE Trans. Image Process., vol. 31, pp. 3852–3867, 2022.
[17] J. Tang, X. Shu, R. Yan, and L. Zhang, “Coherence constrained graph LSTM for group activity recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 2, pp. 636–647, Feb. 2022.
[18] Q. Lao, M. Havaei, A. Pesaranghader, F. Dutil, L. D. Jorio, and T. Favors, “Dual adversarial inference for text-to-image synthesis,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 7567–7576.
[19] J. Cheng, F. Wu, Y. Tian, L. Wang, and D. Tao, “RiFeGAN: Rich feature generation for text-to-image synthesis from prior knowledge,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10911–10920.
[20] T. Qiao, J. Zhang, D. Xu, and D. Tao, “Learn, imagine and create: Text-to-image generation from prior knowledge,” in Proc. NIPS, vol. 32, 2019, pp. 887–897.
[21] D. H. Park, S. Azadi, X. Liu, T. Darrell, and A. Rohrbach, “Benchmark for compositional text-to-image synthesis,” in Proc. NeurIPS Datasets Benchmarks Track, 2021.
[22] G. Yin, B. Liu, L. Sheng, N. Yu, X. Wang, and J. Shao, “Semantics disentangling for text-to-image generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2327–2336.
[23] J. Cheng, F. Wu, Y. Tian, L. Wang, and D. Tao, “RiFeGAN2: Rich feature generation for text-to-image synthesis from constrained prior knowledge,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 8, pp. 5187–5200, Aug. 2022.
[24] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The Caltech-UCSD birds-200-2011 dataset,” California Inst. Technol., CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.
[25] M.-E. Nilsback and A. Zisserman, “Automated flower classification over a large number of classes,” in Proc. 6th Indian Conf. Comput. Vis., Graph. Image Process., Dec. 2008, pp. 722–729.
[26] B. Liu, K. Song, Y. Zou, G. de Melo, and A. Elgammal, “TIME: Text and image mutual-translation adversarial networks,” in Proc. AAAI, 2021, pp. 2082–2090.
[27] H. Tan, X. Liu, X. Li, Y. Zhang, and B. Yin, “Semantics-enhanced adversarial nets for text-to-image synthesis,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 10501–10510.
[28] M. Cha, Y. L. Gwon, and H. Kung, “Adversarial learning of semantic relevance in text to image synthesis,” in Proc. AAAI Conf. Artif. Intell., vol. 33, 2019, pp. 3272–3279.
[29] B. Li, X. Qi, T. Lukasiewicz, and P. H. S. Torr, “Controllable text-to-image generation,” 2019, arXiv:1909.07083.
[30] Z. Zhang and L. Schomaker, “DTGAN: Dual attention generative adversarial networks for text-to-image generation,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2021, pp. 1–8.
[31] M. Yuan and Y. Peng, “CKD: Cross-task knowledge distillation for text-to-image generation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 5281–5290.
[32] B. Zhu and C.-W. Ngo, “CookGAN: Causality based text-to-image face synthesis and manipulation with pretrained BERT model,” in Proc. 29th ACM Int. Conf. Multimedia, Oct. 2020, pp. 4244–4252.
[33] S. Pan, L. Dai, X. Hou, H. Li, and B. Sheng, “ChefGAN: Food image generation from recipes,” in Proc. 28th ACM Int. Conf. Multimedia, Oct. 2020, pp. 4244–4252.
[34] J. Sun, Q. Li, W. Wang, J. Zhao, and Z. Sun, “Multi-caption text-to-face synthesis: Dataset and algorithm,” in Proc. 29th ACM Int. Conf. Multimedia, Oct. 2021, pp. 2290–2298.
[35] W. Xia, Y. Yang, J.-H. Xue, and B. Wu, “Text2GAN: Text-guided diverse face image generation and manipulation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 2290–2298.
[36] Y. Zhou and N. Shimada, “Generative adversarial networks for face synthesis and manipulation with pretrained BERT model,” in Proc. 16th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG), Dec. 2021, pp. 1–8.
[37] Y. Huang, H. Xue, B. Liu, and Y. Lu. “Unifying multimodal transformer for bi-directional image and text generation,” in Proc. 29th ACM Int. Conf. Multimedia, Oct. 2021, pp. 1138–1147.

[38] H. Zhang, Y. J. Koh, J. Baldridge, H. Lee, and Y. Yang, “Cross-modal contrastive learning for text-to-image generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 833–842.

[39] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020, arXiv:2010.11929.

[40] A. Ramesh et al., “Zero-shot text-to-image generation,” 2021, arXiv:2102.12092.

[41] M. Ding et al., “CogView: Mastering text-to-image generation via transformers,” 2021, arXiv:2105.13290.

[42] C. Saharia et al., “Photorealistic text-to-image diffusion models with deep language understanding,” 2022, arXiv:2205.11487.

[43] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with CLIP latents,” 2022, arXiv:2204.06125.

[44] A. Nichol et al., “GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models,” 2021, arXiv:2112.10741.

[45] X. Dong, H. Zhang, L. Zhu, L. Nie, and L. Liu, “Hierarchical feature aggregation based on transformer for image-text matching,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 9, pp. 6437–6447, Sep. 2022.

[46] J. Liu, M. Yang, C. Li, and R. Xu, “Improving cross-modal image-text retrieval with teacher-student learning,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 8, pp. 3242–3253, Aug. 2021.

[47] J. Wu, C. Wu, J. Lu, L. Wang, and X. Cui, “Region reinforcement network with topic constraint for image-text matching,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 1, pp. 385–397, Jan. 2022.

[48] M. Yuan and Y. Peng, “Bridge-GAN: Interpretable representation learning for text-to-image synthesis,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 11, pp. 4258–4268, Nov. 2020.

[49] E. Yu, J. Sun, J. Li, X. Chang, X. Han, and A. G. Hauptmann, “Adaptive semi-supervised feature selection for cross-modal retrieval,” IEEE Trans. Multimedia, vol. 21, no. 5, pp. 1276–1288, May 2019.

[50] E. Yu, J. Ma, J. Sun, X. Chang, H. Zhang, and A. G. Hauptmann, “Dynamic discrete cross-modal hashing with multiple supervision,” Neurocomputing, vol. 486, pp. 215–224, May 2022.

[51] S. Hong, D. Yang, J. Choi, and H. Lee, “Inferring semantic layout for hierarchical text-to-image synthesis,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7986–7994.

[52] T. Hinz, S. Heinrich, and S. Wermter, “Semantic object accuracy for generative text-to-image synthesis,” 2019, arXiv:1910.13321.

[53] Y. Qiao et al., “R-GAN: Exploring human-like way for reasonable text-to-image synthesis via generative adversarial networks,” in Proc. 29th ACM Int. Conf. Multimedia, Oct. 2021, pp. 2085–2093.

[54] F. Tan, S. Feng, and V. Ordonez, “Text2Scene: Generating compositional scenes from textual descriptions,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 6710–6719.

[55] K. Gupta, J. Lazarow, A. Achille, L. S. Davis, V. Mahadevan, and A. Shrivastava, “Layouttransformer: Layout generation and completion with self-attention,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., Oct. 2021, pp. 1004–1014.

[56] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, arXiv:1804.02707.

[57] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[58] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. ECCV. Cham, Switzerland: Springer, 2014, pp. 740–755.

[59] H. Caesar, J. Uijlings, and V. Ferrari, “COCO-stuff: Thing and stuff classes in context,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 1209–1218.

[60] J. Pont-Tuset, J. Uijlings, S. Changpinoyo, R. Soricut, and V. Ferrari, “Connecting vision and language with localized narratives,” in Proc. ECCV. Cham, Switzerland: Springer, 2020, pp. 647–664.

[61] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[62] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training GANs,” 2016, arXiv:1506.02424.

[63] M. Hetzel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” 2017, arXiv:1706.08500.

[64] A. Radford et al., “Learning transferable visual models from natural language supervision,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 8748–8763.

[65] A. G. Patil, O. Ben-Eliezer, O. Perel, and H. Averbuch-Elor, “READ: Recursive autoencoders for document layout generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 544–545.

[66] R. Zhang, P. Isola, A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 586–595.

[67] H. Tan, X. Liu, B. Yin, and X. Li, “Cross-modal semantic matching generative adversarial networks for text-to-image synthesis,” IEEE Trans. Multimedia, vol. 24, pp. 832–845, 2022.

[68] R. Li, N. Wang, J. Peng, G. Zhang, and X. Wang, “Exploring global and local linguistic representations for text-to-image synthesis,” IEEE Trans. Multimedia, vol. 22, no. 12, pp. 3075–3087, Dec. 2020.

[69] J. Peng et al., “Knowledge-driven generative adversarial network for text-to-image synthesis,” IEEE Trans. Multimedia, vol. 24, pp. 4356–4366, 2022.

[70] F. Feng, T. Niu, R. Li, and X. Wang, “Modality disentangled discriminator for text-to-image synthesis,” IEEE Trans. Multimedia, vol. 24, pp. 2112–2124, 2022.

[71] S. Gu et al., “Vector quantized diffusion model for text-to-image synthesis,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 10969–10976.

[72] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2818–2826.

Jiadong Liang is currently pursuing the Ph.D. degree with the State Key Laboratory of Virtual Reality Technology and System, School of Computer Science and Engineering, Beihang University. His research interests include computer vision and image generation.

Wenjie Pei (Member, IEEE) received the Ph.D. degree from the Delft University of Technology, Delft, The Netherlands, in 2018, under the supervision of Dr. Laurens van der Maaten and Dr. David Tax. In 2016, he was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA. He is currently an Associate Professor with the Harbin Institute of Technology, Shenzhen, China. Before joining the Harbin Institute of Technology, he was a Senior Researcher of computer vision with the Tencent YouTu X-Laboratory, Shenzhen. His research interests include computer vision and machine learning.

Feng Lu (Senior Member, IEEE) received the B.S. and M.S. degrees in automation from Tsinghua University in 2007 and 2010, respectively, and the Ph.D. degree in information science and technology from The University of Tokyo in 2013. He is currently a Professor with the State Key Laboratory of Virtual Reality Technology and System, School of Computer Science and Engineering, Beihang University. His research interests include computer vision, human–computer interaction, and augmented intelligence.