PHRASEGAN: PHRASE-BOOST GENERATIVE ADVERSARIAL NETWORK FOR TEXT-TO-IMAGE GENERATION

Fei Fang[1], *Ziqing Li*[1], *Fei Luo*[1*], *Chengjiang Long*[2], *Shenghong Hu*[3], *and Chunxia Xiao*[1*]

1. School of Computer Science, Wuhan University, Wuhan 430072, China;
2. JD Finance American Corporation, Mountain View, CA, USA;
3. Information Engineering School, Hubei University of Economics, Wuhan 430205.
fangfei369@163.com, thalialee@163.com, luofei_whu@126.com, cjfykx@gmail.com,
wuhanhush@126.com, cxxiao@whu.edu.cn

ABSTRACT

A phrase contains an object-orienting noun and some attribution-associating words. Therefore, focusing on phrases could better generate images with the objects and their tightly relevant characteristics. We propose a Phrase-boost Generative Adversarial Network (PhraseGAN) with threefold improvement for scene level text-to-image generation. First, we propose a Transformer-based encoder to encode the input words and sentences and encode related words and their targeting nouns into phrases by text correlation analysis. Second, we utilize Graph Convolution Networks to measure fine-grained text-image similarity, which could gain constraints on relative positions between different objects. Finally, we design a phrase-region discriminator to discriminate the quality of the generated objects and the consistency between the phrases and their corresponding objects. Experimental results on the Microsoft COCO dataset demonstrate that PhraseGAN can generate better images from texts than state-of-the-art methods.

Index Terms— Text-to-image generation, phrase, transformer, GCN

1. INTRODUCTION

Prior text-to-image generation methods based on Generative Adversarial Network (GAN) can generate high-quality images with single objects (*e.g.*, bird, flower). However, current methods can not get satisfactory results for scene-level text-to-image tasks. It is hard to visually construct scene images composed of multiple objects, their attributions, and their spatial relationships from one text sentence.

In this work, we focus on the phrase to improve GAN-based scene-level text-to-image generation. The fact behind such motivation is that the words in one phrase can describe both objects and their high-level attributions (*e.g.*, a yellow dog, a young girl). To generate high-quality scene images with multiple foreground objects, we propose a Phrase-boost GAN named PhraseGAN. Concretely, we first encode the input text into word and sentence embeddings using a Transformer-based text encoder (TTE). Then, we analyze the words correlation using Natural Language Processing (NLP) approach and obtain the phrase embedding from the word embedding. Since Graph Convolutional Networks (GCN) can capture the spatial relationships between objects [1], we propose a GCN-based text-image similarity model (GTISM) based on phrase embeddings and GCNs. Our GTISM can estimate the fine-grained phrase-object similarity and measure the correctness of the relative distances between objects utilizing the relative polar distances. Finally, we propose a phrase-object discriminator (POD) to determine whether the generated images for objects are realistic and consistent with the corresponding phrases.

The contributions of this paper can be summarized as follows:

(1) We propose a Transformer-based Text Encoder for multiple level embedding, including word embedding, sentence embedding, and phrase embedding;

(2) We propose a new GCN-based text-image similarity model to measure fine-grained text-image similarity based on phrase embedding;

(3) We propose a Phrase-object discriminator to improve the quality and phrase-object consistency of the generated scene images.

2. RELATED WORKS

Single object text-to-image generation. Reed et al. [2] first used conditional GAN to generate low resolution (64×64) images of single objects. Zhang et al. [3] stacked multiple GANs to generate images. Xu et al. [4] proposed AttnGAN by adding an attention mechanism to associate the sub-region images with their relevant words. They al-

This work is partially supported by the Key Technological Innovation Projects of Hubei Province (2018AAA062) and NSFC (No. 61972298). ∗ Chunxia Xiao and Fei Luo are corresponding authors.

so proposed a Deep attentional multimodel similarity model (DAMSM) to measure text-image similarity. Zhu et al. [5] proposed dynamic memory and gating mechanism to fuse the important text and the initially generated images. Qiao et al. [6] proposed a text-to-image-to-text framework by regenerating the text descriptions from the generated images to promote text-image consistency. Hu et al. [7] proposed DC-GAN to generate diverse single-object images that are semantically consistent with the same input text. Hu et al. [8] proposed SSA-GAN to operate semantic-spatial condition batch normalization, which deepened the text-image fusion through the image generation process and guaranteed text-image consistency. Such methods are relatively weak in generating images for scene-level situations, but they provide basic methods for text-to-image generation. In our work, we improve the DAMSM in [4] by proposing a novel GTISM to better measure the text-image similarities based on novel phrase embeddings.

Scene-level text-to-image generation. Scene-level text-to-image generation needs to construct more things, including high-level attributions of objects, spatial layout, etc. Some early methods [9, 10] built scene graphs for the foreground objects or utilized optimization algorithms to handle this constrained scene image generation task. Li et al. [11] took a layout-to-image strategy to synthesize scene images, but this method have complicated training and generating process. Liang et al. [12] proposed a CPGAN and built memory information for every word in the vocabulary. In addition, they integrated each word with its visual context, which was composed of relevant object region features. This technique can effectively improve the quality of generated images. However, it would lead to mode dropping (decline in the diversity for each kind of object), as they did not distinguish the instances. Some GAN-based methods directly utilize additional scene information from the dataset annotations. For example, Hinz et al. [13] introduced bounding boxes and labels of the objects from dataset annotations to generate all the objects in the scene images. Then they produced scene images by merging the generated objects from the object pathway and the background from the global pathway. Our proposed PhraseGAN builds corresponding relationships between phrases in the input text and instances in the training and generated images, which will alleviate the mode dropping of the generated objects.

3. METHOD

The architecture of PhraseGAN is shown in Fig. 1. It first uses a Transformer-based text encoder (TTE) to encode the input text into word embedding, sentence embedding, and phrase embedding. Then the GCN-based text-image similarity model (GTISM) estimates the text-image similarity and models the relative positions between YOLOv4 detected objects. Finally, the proposed Phrase-object discriminator (POD) evaluates the quality of synthesized scene images and estimates phrase-object consistency to guide image generation.

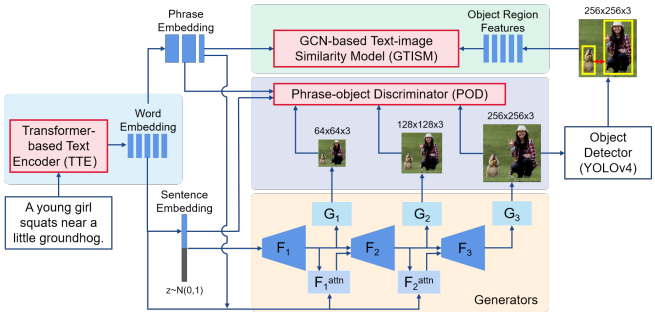

Fig. 1. The overview of the proposed PhraseGAN. The contributions are denoted by red boxes.

3.1. Transformer-based text encoder

3.1.1. Word and sentence embedding

We firstly encode the input sentence into the word embedding matrix $E \in \mathbb{R}^{t \times d}$ using the encoder part of Transformer [14], where t is the number of words in the input sentence, and d is the feature dimension of each word. Note that we concatenate the memory features constructed by [12] with the initial word embeddings before Transformer encoding.

Next, we encode the sentence embedding based on the obtained word embedding. First, we compute a weight vector $W = [w_0, w_1, ..., w_{t-1}]^T \in \mathbb{R}^{t \times 1}$ for the word embedding matrix E by the Softmax normalization:

$$w_i = \frac{\exp(e_i)}{\sum_{k=0}^{t-1} \exp(e_k)}, \tag{1}$$

where e_i is the i-th word embedding , and w_i indicates the importance of the i-th word in all words. Then the sentence embedding $S \in \mathbb{R}^{1 \times d}$ is computed by:

$$S = E^T \cdot W. \tag{2}$$

The sentence embedding S is further concatenated with a noise vector $z \in \mathbb{R}^{1 \times 100}$ which is sampled from the standard normal distribution and input into GANs to generate images.

3.1.2. Phrase embedding

To compute the phrase embedding, we conduct text correlation analysis, which consists of adjacency analysis and similarity analysis, as shown in Fig. 2. The adjacency analysis produces an adjacent matrix $G_t^A \in \mathbb{R}^{n \times t}$ (n is the number of phrases) to determine which words belong to the same phrase. Specifically, we use the popular Stanford Core NLP [15] tools to obtain the constituency parsing and dependency parsing

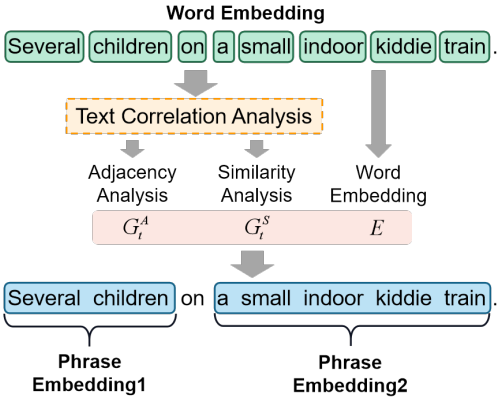

Fig. 2. The scheme of phrase embedding based on text correlation analysis and word embedding.

results to decide which words describe the same object. According to the results, we build the adjacent matrix G_t^A for all the words in the input sentence. We call the words describing the same object semantical correlated and consider them a noun phrase. We can also get the number of phrases n from the adjacency analysis.

The similarity analysis measures the correlation degree between different words with a similarity matrix $G_t^S \in \mathbb{R}^{t\times t}$. The G_t^S is viewed as a undirected fully connected graph. Inspired by [16], the normalized edge weight between the i-th and j-th node is:

$$g_{ij}^s = \frac{\exp\left(e_i^T e_j\right)}{\sum_{k=0}^{t-1} \exp\left(e_i^T e_k\right)}, \tag{3}$$

where the g_{ij}^s is the value of the i-th row and j-th column in G_t^S. Finally, we get the phrase embedding $P \in \mathbb{R}^{n\times d}$ according to G_t^A and G_t^S:

$$P = \|G_t^A G_t^S\|_2 E, \tag{4}$$

where $\|\cdot\|_2$ means L_2 normalization and n is the number of phrases.

3.2. GCN-based text-image similarity model

We propose a GCN-based text-image similarity model (GTISM) to further compute the fine-grained text-image similarity. We also use GCNs to model (1) the semantic relative position relationships, (2) relative distances, and (3) orientations between pairs of objects.

Firstly, we extract the semantic relative position relationships from the input text and detect objects in images as shown in Fig. 3 (a). The text dependency parsing is conducted using the same Stanford CoreNLP tools [15] as the building of G_t^A in Section 3.1.2. We select relationships about prepositions and verbs from the output results of the dependency parser. Then we use the pre-trained YOLOv4 [17] to detect the object regions (indicated by bounding boxes) with the highest confidence scores in the generated images. We extract feature $F^o \in \mathbb{R}^{n_o\times d}$ from each region, where n_o is the number of detected object regions and d is the feature dimension. We also use the non-max suppression technique to further ensure the detection precision to remove the redundant detection results.

Secondly, we build semantic relationship graph G_u and spatial relationship graph G_v for the phrase embeddings and the detected object region features, respectively. We visualize the graphs of G_u and G_v in Fig. 3 (b). Both of them can be represented as symmetric matrix $G_u \in \mathbb{R}^{n\times n}$ and $G_v \in \mathbb{R}^{n_o\times n_o}$, where n is the number of phrases. The G_u represents the extracted semantic relative positional relationships between different phrases specified in the input text. The G_u is a directed graph, and the graph nodes are the phrase embeddings computed in Section 3.1.2. If the positional relationships of nodes are mentioned in the input sentence, edges exist between them. The edge weight in G_u is the corresponding word embedding of the words that represent relationships.

Meanwhile, we build G_v to represent the relative positional relationships between the detected object region. The G_v is also a directed graph, and the nodes of G_v are the detected object regions. Similar to G_u, we will build edges between nodes when their positional relationships are mentioned in the input sentence. We add weights to the edges of G_v by using polar coordinates to model the spatial distances and orientations between pairs of object regions in images like [16]. The spatial distance between two object regions represents their relative spatial distance, and the orientation represents the category of the spatial relation (e.g., on, near). The spatial distance is the Euclidean distance between the center points of two objects' bounding boxes.

Thirdly, we use two GCNs to further process the semantic and spatial relationship graphs G_u and G_v, respectively. We input the G_u into one GCN and G_v into another GCN. Finally, we will use the output of both GCNs to compute the phrase-object similarity. The layers in both GCNs will apply K kernels to learn how to integrate the neighborhood phrase embeddings or object region features:

$$\hat{p}_i = \sum_{k=0}^{K-1} \sigma\Big(\sum_{p_j\in N_i^u} W_k^u G_u p_j + b^u\Big),\ i = 0, 1, ..., (n-1), \tag{5}$$

$$\hat{f}_i = \sum_{k=0}^{K-1} \sigma\Big(\sum_{f_j^o\in N_i^v} W_k^v G_v f_j^o + b^v\Big),\ i = 0, 1, ..., (n_o-1), \tag{6}$$

where N_i^u is the neighbour set of the i-th phrase embedding and N_i^v is the neighbour set of the i-th object feature, σ is the ReLU activation function, W_k^u, b^u, W_k^v and b^v are the learning parameters for two GCNs, n_o is the number of object regions and n is the number of phrases, and $\hat{p}_i$ and $\hat{f}_i$ are

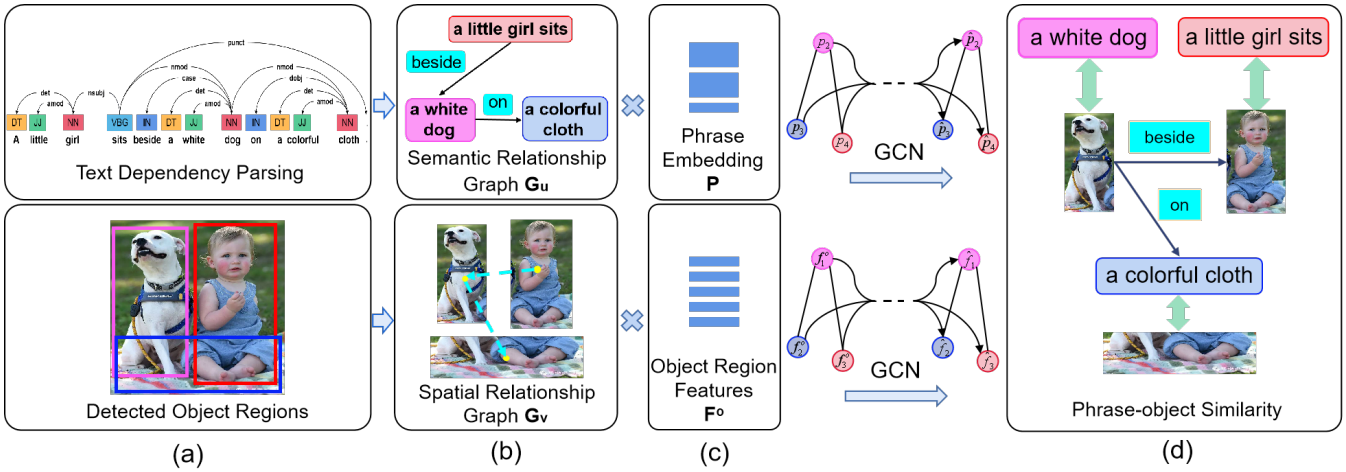

Fig. 3. The scheme of the GTISM module.

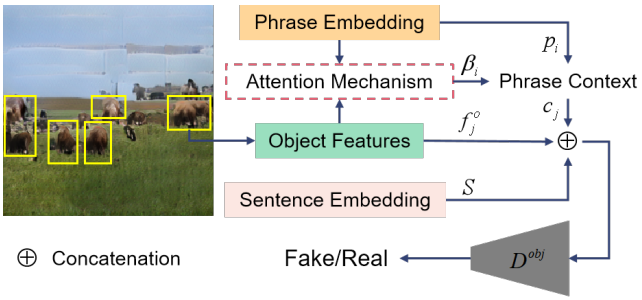

Fig. 4. The diagram of proposed conditional POD.

the contextual text features and the contextual visual features output from respective GCN.

Finally, to measure phrase-object similarity, we modify DAMSM loss [4] to a new $\mathcal{L}_{GTISM}$ by replacing the word embedding and grid region features in DAMSM loss with $\hat{p}_i$ and $\hat{f}_i$, respectively. Apart from the object regions mentioned in the input sentence, there are also background regions such as sky, lawn, and indoor scene. We adopt the OAIE [12] to compute the text-image similarity for the background regions.

3.3. Phrase-object discriminator

For conditional Phrase-object discriminator (POD), we compute the phrase context for each object region by attention mechanism. The structure of conditional POD is shown in Fig. 4.

Firstly, we compute the normalized text-image similarity $s_{i,j}$ between the i-th phrase embedding and the j-th object region feature by:

$$s_{i,j} = \frac{\exp(p_i f_j^o)}{\sum_{k=0}^{n_o-1} \exp(p_i f_k^o)}, \tag{7}$$

Next, we compute the weight of i-th phrase embedding for j-th region by:

$$\beta_i = \frac{\exp(s_{i,j})}{\sum_{k=0}^{n-1} \exp(s_{k,j})}. \tag{8}$$

Furthermore, we get the phrase context for the j-th region by computing the weighted sum of all phrases:

$$c_j = \sum_{k=0}^{n-1} \beta_k p_k, \tag{9}$$

Finally, we concatenate 1) the phrase context vector c_j, 2) the object region feature f_j^o and 3) the sentence embedding S and input the concatenated features into the down-sampling convolutional network D^{obj} to judge whether the object is consistent with the corresponding phrase:

$$D^{obj}(\hat{x}, P, S) = [\sum_{j=0}^{n_o-1} c_j, \sum_{j=0}^{n_o-1} f_j^o, S], \tag{10}$$

where $\hat{x}$ is the synthesized scene image and $[\cdot]$ means the operation of concatenation.

4. EXPERIMENTS

Dataset. We use Microsoft COCO 2014 [18] (MSCOCO14) to evaluate the performance of our PhraseGAN and compare it with other state-of-the-art methods. Most of the images in the training and testing datasets are scene images and contain 80 different kinds of objects. In addition, each image is annotated with five human-annotated captions. We randomly select one caption to generate the corresponding scene image. The training set of MSCOCO14 has 82783 images, and the validation set has 40505 images.

Table 1. Comparison with recent text-to-image generation methods on three metrics, † means the scores are computed from images generated by the open-sourced models.

Method	Venue	IS↑	FID↓	R-prec↑
AttnGAN [4]	CVPR 2018	25.89	33.10	82.98
DMGAN [5]	CVPR 2019	30.49	32.64	88.56
MirrorGAN [6]	CVPR 2019	26.47	30.22	74.52
ObjGAN [11]	CVPR 2019	27.37	25.85	86.84
OP-GAN [13]	ICLR 2019	24.76	33.35	87.90
CPGAN [12]	ECCV 2020	**52.73**	48.87†	89.23†
SSAGAN [8]	arXiv 2021	23.20	21.08†	81.92
PhraseGAN (Ours)	ICME 2022	36.35	**20.27**	**93.26**

Evaluation metrics. We use Inception score [19], *Fréchet inception distance* [20] and R-precition [4] to evaluate the performance of comparative methods. The inception score (IS) is extensively used to evaluate the quality of generated images considering both realism and diversity. The Fréchet inception distance (FID) evaluates the distance between real samples and the generated samples in feature space. Note that lower FID indicates higher image quality and diversity. The R-precision measures the semantic consistency between the input text and the generated image. Our experiments are conducted on one GPU of GeForce RTX 3090Ti.

4.1. Quantitative and qualitative evaluations

Table 1 illustrates the quantitative comparison results between the state-of-the-art methods and the proposed PhraseGAN. The PhraseGAN achieves the best R-precision score and FID, proving that PhraseGAN can better maintain text-image consistency. Our PhraseGAN gets the second-highest IS, which is lower than CPGAN. According to the principles of IS and FID, IS does not penalize the inner-class mode dropping (low diversity for each kind of object) in the generated images. On the contrary, FID is more sensitive to this situation. Considering this characteristic of IS, we believe FID can more comprehensively reflect the quality of the generated images.

The qualitative evaluation results are shown in Fig. 5, which demonstrate the advantages of the phrase boost of PhraseGAN. For example, in the first column of the resulting images, our method only generates the objects in the input text without introducing any other no-existing objects like the sky. PhraseGAN better understands the phrase meaning of a grass field, so its region takes up most of the generated image. We could find similar demonstrations in other column examples. The phrases boosted by PhraseGAN are marked with italic and bold format in the input sentences.

4.2. Ablation study

The ablation study is conducted by removing two important modules from the three proposed modules of PhraseGAN (*i.e.*

Fig. 5. Qualitative comparison between our method (last row) and three recent methods, namely ObjGAN [11], S-SAGAN [8], and CPGAN [12].

Table 2. Ablation study on three evaluation metrics.

Method	IS↑	FID↓	R-prec↑
Baseline	33.71 ± 0.29	32.36	88.45
PhraseGAN (TTE)	34.58 ± 0.64	30.22	90.21
PhraseGAN (GTISM)	**35.53 ± 0.70**	**22.64**	91.69
PhraseGAN (POD)	35.17 ± 0.57	24.13	**92.52**
PhraseGAN (All)	**36.35 ± 0.69**	**20.27**	**93.26**

TTE, GTISM, POD) to validate the performance of each proposed module. We remove all three proposed modules from PhraseGAN as the baseline and conduct quantitative comparisons with the three ablation conditions. The quantitative results are illustrated in Table 2, where the abbreviations in brackets are retained modules.

The results of PhraseGAN-TTE in Table 2 indicate that the proposed TTE module can obviously improve the three evaluation metrics of the baseline method. This result proves that the TTE module has better performance than the traditional LSTM-based text encoder. TTE can better extract the semantic features in the input text and fully exploit the relevance between different words. Compared with the baseline, the PhraseGAN-GTISM gets the best IS and FID scores, demonstrating that the GTISM module can effectively promote the realism and diversity of the generated images. In the final ablation study of PhraseGAN-POD, we use the proposed POD as the discriminator to train the image generator. We can find that the PhraseGAN-POD achieves the best R-precision score than PhraseGAN-TTE and PhraseGAN-GTISM, which

demonstrates that the proposed POD can effectively improve the semantic consistency of the input text and generated images.

5. CONCLUSION

In this paper, we propose a PhraseGAN model to generate better scene-level images. We focus on the important role of the phrase in describing objects, their attributions, and their spatial relationships. Relevant components have been proposed and applied in the PhraseGAN. The quantitative experimental results have proven the technical contributions of PhraseGAN. Especially, some representative cases have illustrated the visual advantages brought by the introduction of Phrase boost.

6. REFERENCES

[1] Xinzhi Dong, Chengjiang Long, Wenju Xu, and Chunxia Xiao, "Dual graph convolutional networks with transformer and curriculum learning for image captioning," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 2615–2624.

[2] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee, "Generative adversarial text to image synthesis," in *ICML*. PMLR, 2016, pp. 1060–1069.

[3] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas, "Stackgan++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Transactions on PAMI*, vol. 41, no. 8, pp. 1947–1962, 2018.

[4] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in *Proceedings of the IEEE Conference on CVPR*, 2018, pp. 1316–1324.

[5] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang, "Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis," in *Proceedings of the IEEE Conference on CVPR*, 2019, pp. 5802–5810.

[6] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao, "Mirrorgan: Learning text-to-image generation by redescription," in *Proceedings of the IEEE Conference on CVPR*, 2019, pp. 1505–1514.

[7] Tao Hu, Chengjiang Long, and Chunxia Xiao, "A novel visual representation on text using diverse conditional gan for visual recognition," *IEEE Transactions on Image Processing*, vol. 30, pp. 3499–3512, 2021.

[8] Wentong Liao, Kai Hu, Michael Ying Yang, and Bodo Rosenhahn, "Text to image generation with semantic-spatial aware gan," *arXiv preprint arXiv:2104.00567*, 2021.

[9] Fei Fang, Miao Yi, Hui Feng, Shenghong Hu, and Chunxia Xiao, "Narrative collage of image collections by scene graph recombination," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 9, pp. 2559–2572, 2017.

[10] Fei Fang, Fei Luo, Hong-Pan Zhang, Hua-Jian Zhou, Alix LH Chow, and Chun-Xia Xiao, "A comprehensive pipeline for complex text-to-image synthesis," *Journal of Computer Science and Technology*, vol. 35, no. 3, pp. 522–537, 2020.

[11] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao, "Object-driven text-to-image synthesis via adversarial training," in *Proceedings of the IEEE Conference on CVPR*, 2019, pp. 12174–12182.

[12] Jiadong Liang, Wenjie Pei, and Feng Lu, "Cp-gan: Content-parsing generative adversarial networks for text-to-image synthesis," in *ECCV*. Springer, 2020, pp. 491–508.

[13] T. Hinz, S. Heinrich, and S. Wermter, "Generating multiple objects at spatially distinct locations," in *ICLR*, 2019.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[15] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the ACL*, 2014, pp. 55–60.

[16] Chunxiao Liu, Zhendong Mao, Tianzhu Zhang, Hongtao Xie, Bin Wang, and Yongdong Zhang, "Graph structured network for image-text matching," in *Proceedings of the IEEE Conference on CVPR*, 2020, pp. 10921–10930.

[17] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.

[19] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, "Improved techniques for training gans," *arXiv preprint arXiv:1606.03498*, 2016.

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *arXiv preprint arXiv:1706.08500*, 2017.