Visual-Relation Conscious Image Generation from Structured-Text

Duc Minh Vo[1(✉)] and Akihiro Sugimoto[2]

[1] The Graduate University for Advanced Studies, SOKENDAI, Tokyo, Japan
vmduc@nii.ac.jp
[2] National Institute of Informatics, Tokyo, Japan
sugimoto@nii.ac.jp

Abstract. We propose an end-to-end network for image generation from given structured-text that consists of the visual-relation layout module and stacking-GANs. Our visual-relation layout module uses relations among entities in the structured-text in two ways: comprehensive usage and individual usage. We comprehensively use all relations together to localize initial bounding-boxes (BBs) of all the entities. We use individual relation separately to predict from the initial BBs relation-units for all the relations. We then unify all the relation-units to produce the visual-relation layout, i.e., BBs for all the entities so that each of them uniquely corresponds to each entity while keeping its involved relations. Our visual-relation layout reflects the scene structure given in the input text. The stacking-GANs is the stack of three GANs conditioned on the visual-relation layout and the output of previous GAN, consistently capturing the scene structure. Our network realistically renders entities' details while keeping the scene structure. Experimental results on two public datasets show the effectiveness of our method.

1 Introduction

Generating photo-realistic images from text descriptions (T2I) is one of the major problems in computer vision. Besides having a wide range of applications such as intelligent image manipulation, it drives research progress in multimodal learning and inference across vision and language [1–3].

The GANs [4] conditioned on unstructured text description [5–9] show remarkable results in T2I. Stacking such conditional GANs has shown even more ability of progressively rendering a more and more detailed entity in high resolution [6,9]. However, in more complex scenarios where sentences are with many entities and relations, their performance is degraded. This is because they use

The authors are thankful to Zehra Hayırcı for her valuable comments on this work.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58604-1_18) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12373, pp. 290–306, 2020.
https://doi.org/10.1007/978-3-030-58604-1_18

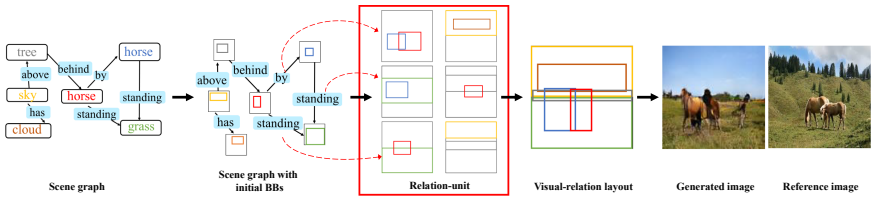


Fig. 1. Overall framework of our method. Given a structured-text (scene graph), our method predicts initial BBs for entities using all relations together, next takes individual relation one by one to infer a relation-unit for the relation, then unifies all the relation-units to produce visual-relation layout. Finally, the layout is converted to an image. Color of each entity BB corresponds to that in the scene graph. Red dotted arrow means the individual usage of relations. (Color figure online)

only entity information in given text descriptions for rendering a specific entity, leading to a poor layout of multiple entities in generated images.

In the presence of multiple entities, besides the details of each entity, how to localize all the entities so that they reflect given relations becomes crucial for better image generation. Indeed, recent work [1,10–12] show the effectiveness of inferring the scene layout first from given text descriptions. Johnson+ [1], Li+ [10], and Ashual+ [11] use structured-text, i.e., scene graphs [2], first to construct a scene layout by predicting bounding boxes and segmentation masks for all entities, then convert it to an image. Hong+ [12] constructs a semantic layout, a scene structure based on object instances, from input text descriptions and converts the layout into an image. However, those mentioned methods [1,10–12] aggregate all relations in which each entity is involved, and then localize all entities' bounding-boxes at the same time. As a result, the predicted bounding-boxes do not preserve the relations among entities well. Localizing entities faithfully by preserving their relations given in text descriptions is desired.

We leverage advantages of the pyramid of GANs and inferring the scene layout, proposing a GAN-based model for T2I where our network steps further in relation usage by employing not only all available relations together but also individual relation separately. We refer the former usage of relations as *comprehensive* while the latter as *individual*. Our network has two steps: (1) inferring from input the *visual-relation layout*, i.e., localized bounding-boxes for all the entities so that each of which uniquely corresponds to each entity and faithfully preserves relations between the entities, and (2) progressively generating coarse-to-fine images with the pyramid of GANs, namely stacking-GANs, conditioned on the visual-relation layout. The first step takes the comprehensive usage of relations first to generate initial bounding-boxes (BBs) for entities as in [1,10–12], and then takes the individual usage to predict a relation-unit for each *subject–predicate–object* relation where all the relations in the input are extracted through its scene graph [2]. Each relation-unit consists of *two* BBs that participate in the relation: one for a "subject" entity and one for an "object" entity. Since one entity may participate in multiple relations, we then unify all

the relation-units into refined (entity) BBs (including their locations and sizes) so that each of them uniquely corresponds to one entity while keeping their relations in the input text. Aggregating the refined BBs allows us to infer the visual-relation layout reflecting the scene structure given in the text. In the second step, three GANs progressively generate images where entities are rendered in more and more details while preserving the scene structure. At each level, a GAN is conditioned on the visual-relation layout and the output of previous GAN. Our network is trained in a fully end-to-end fashion.

The main contribution of our proposed method is our introduction to the individual usage of *subject–predicate–object* relations for localizing entity bounding-boxes, so that our proposed *visual-relation layout* surely preserves the visual relations among entities. In addition, we stack and condition GANs on the visual-relation layout to progressively render realistic detailed entities that keep their relations even from complex text descriptions. Experimental results on COCO-stuff [13] and GENOME [14] demonstrate outperformances of our method against state-of-the-arts. Figure 1 shows the overall framework of our proposed method.

2 Related Work

Recent GAN-based methods have shown promising results on T2I [1,5,6,8,9,12,15]. They, however, struggle to faithfully reproduce complex sentences with many entities and relations because of the gap between text and image representations.

To overcome the limitation of GANs conditioned on text descriptions, a two-step approach was proposed where inference of the scene layout as an intermediate representation between text and image is followed by using the layout to generate images [1,10–12]. Since the gap between the intermediate representation and image is smaller than that of text and image, this approach generates more realistic images. Zhao+ [16] and Sun+ [17] propose a combination of ground-truth (GT) layout and entity embeddings to generate images. Hong+ [12] infers a scene layout by feeding text descriptions into a LSTM. More precisely, they use a LSTM to predict BBs for all entities independently, then employ a bi-directional conv-LSTM to generate entity shapes from each predicted BB without using any relation. The function of the bi-directional conv-LSTM used here is just the putting-together. They then combine the layout with text embeddings obtained from the pre-trained text encoder [7], and use a cascade refinement network (CRN) [18] for generating images.

Johnson+ [1], Li+ [10], and Ashual+ [11] employ a scene graph [2] to predict a layout and then condition CRN [18] on the layout. The graph convolution network (GCN) used in these methods aggregates available relations of all the entities together along the edges of the scene graph. Namely, only the comprehensive usage of relations is employed. Consequently, individual relation information is lost at the end of GCN because of the averaging operation on entity embeddings. Averaging entity embeddings means mixing different relations in which a single entity is involved, resulting in failure of retaining individual relation information. Different from [1], [10] retrieves entity appearances from a pre-defined

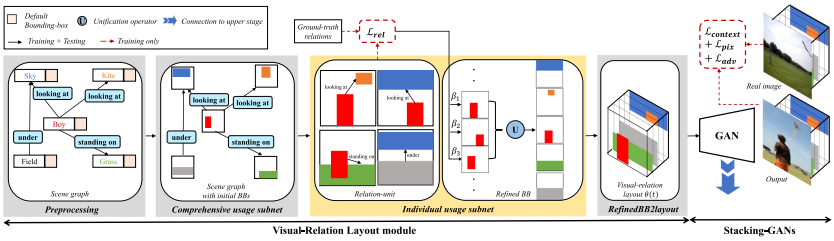


Fig. 2. Our proposed network model consisting of the visual-relation layout module and the Stacking-GANs.

tank while [11] adds entity appearances to the layout before feeding it to the generation part. The layout in [1,10–12] is constructed through only the comprehensive usage of relation among entities for BBs' localization, leading poor scene structure as a whole even if each entity is realistically rendered.

Our main difference from the aforementioned methods is to construct the visual-relation layout using *subject–predicate–object* relations between entities extracted from an input structured-text not only comprehensively but also individually. Recursively conditioning stacking-GANs on our constructed visual-relation layout enables us to progressively generate coarse-to-fine images that consistently preserve the scene structure given in texts.

3 Proposed Method

Our method is decomposed into two steps: (1) inferring the visual-relation layout $\theta(t)$ from structured-text description t, and (2) generating an image from the visual-relation layout, namely $\hat{I} = G(\theta(t))$. To this end, we design an end-to-end network with two modules: the visual-relation layout module and the stacking-GANs (Fig. 2). We train the network in a fully end-to-end manner.

3.1 Visual-Relation Layout Module

The visual-relation layout module constructs the visual-relation layout $\theta(t)$ from a given structured-text description t (Fig. 3) where t is assumed to be converted into a scene graph [2], i.e., the collection of *subject–predicate–object*'s. After the pre-processing on converting t to its scene graph, the comprehensive usage subnet in this module predicts initial BBs for all the entities in t by aggregating all available relations together through GCN ("comprehensive usage"). The individual usage subnet takes each *subject–predicate–object* relation from the scene graph one by one and select the pair of initial BBs involved in the relation (predicate): one for "subject" entity and one for "object" entity. The subnet then adjusts the location and size of the pair of initial BBs using the relation ("individual usage") to have a relation-unit for the relation. Since one entity may participate in multiple relations, it next unifies relation-units so that each

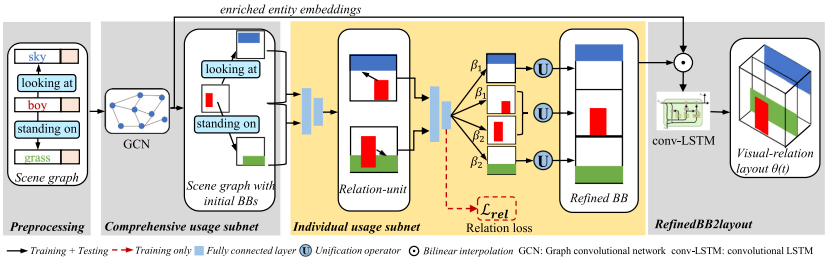


Fig. 3. Details of visual-relation layout module. This figure illustrates the prediction for two *subject–predicate–object* relations.

entity uniquely has a single BB (called refined BB) that is further adjusted in location and size using weights learned from all the participating relations. The RefinedBB2layout subnet constructs the visual-relation layout by aggregating all the refined BBs together using conv-LSTM.

Preprocessing. Similar to [1], we convert the structured-text t to its scene graph (E, P) where $E \subseteq \mathcal{C}$ and $P \subseteq \mathcal{C} \times \mathcal{R} \times \mathcal{C}$. $\mathcal{C}$ and $\mathcal{R}$ are the set of categories and the set of relations given in a dataset. An edge of (E, P) is associated with one *subject–predicate–object*. It is directed and represented by $(e^{\mathrm{s}}, p, e^{\mathrm{o}})$ with entities $e^{\mathrm{s}}, e^{\mathrm{o}} \in E$ and predicate $p \in \mathcal{R}$ (s and o indicate subject and object).

Like [1], we use a learned embedding layer to produce the entity embedding with the size of $1 \times |\mathcal{C}|$ and the predicate embedding with the size of $1 \times |\mathcal{R}|$ for any of all the entities and predicates appearing in (E, P). Any entity embedding is associated with a single default BB presented by $[x, y, w, h] \in [0, 1]^4$ where x is the *left coordinate*, y is the *top coordinate*, w is the *width*, and h is the *height*. We set $x = y = 0$ and $w = h = 1$ as default. This process ensures that all the entities appear in the image. In practice, we concatenate the default BB and its associated entity embedding to produce the vector with the size of $1 \times (|\mathcal{C}| + 4)$.

Comprehensive Usage Subnet. This subnet applies the comprehensive usage to predict a single initial BB for each entity appearing in t as in [1,10–12]. This subnet gives us initial locations and sizes of entities and they do not necessarily satisfy the relations given in t.

In order to aggregate all information along the edges in the scene graph, we employ GCN [1]. Our GCN is mostly identical to [1] with a modification that produces 388 outputs instead of 384 not only to enrich entity/predicate embeddings as in [1,10,11] but also to infer initial BBs. We do not use the average pooling layer on top of GCN to retain individual relation information.

For each edge k of (E, P), the triplet $(\boldsymbol{e}^{\mathrm{s}}_k, \boldsymbol{p}_k, \boldsymbol{e}^{\mathrm{o}}_k)$ and two default BBs with the size of $1 \times (|\mathcal{C}| + |\mathcal{R}| + |\mathcal{C}| + 8)$ are processed to give enriched $\boldsymbol{e}'^{\mathrm{s}}_k$, $\boldsymbol{p}'_k$, and $\boldsymbol{e}'^{\mathrm{o}}_k$ embeddings with the size of 1×128 each, separately, and a pair of initial BBs (one for "subject" and one for "object") with the size of 1×4 each.

Individual Usage Subnet. Since the initial BBs of the entities do not always satisfy the relations given in t, we adjust their locations and sizes using each

relation separately. For each relation, we select a pair of initial BBs corresponding to the "subject" and "object" involved in the relation, and adjust the locations and sizes of the pair of BBs using the relation to have a relation-unit for the relation consisting of *two* BBs for "subject" and "object" entities in the relation. This process causes the situation where multiple BBs correspond to the same entity, as different relations may involve same entities in common. We thus move to focus on each entity to unify its corresponding BBs into a single BB (called refined BB) where we use weights learned to retain all the relations. Accordingly, the function of this subnet is two-fold: relation-unit prediction using individual relation separately and unification of multiple BBs corresponding to the same entity into a single refined BB. The subnet is built upon two fully-connected layers followed by a ReLU layer [19] producing 512 and 8 outputs.

For each edge k of scene graph (E, P), its enriched embeddings and its corresponding pair of initial BBs with the size of $1 \times 392 (= 128+4+128+128+4)$ are fed into this subnet to infer relation-unit $(\boldsymbol{b}^{\mathrm{s}}_k, \boldsymbol{b}^{\mathrm{o}}_k)$ with the size of 1×8. Each BB ($\boldsymbol{b}^{\mathrm{s}}_k$ or $\boldsymbol{b}^{\mathrm{o}}_k$) in the relation-unit is associated with enriched embedding either $\boldsymbol{e}'^{\mathrm{s}}_k$ or $\boldsymbol{e}'^{\mathrm{o}}_k$, respectively for "subject" or "object". We remark that the total number of obtained BBs is $|\{\boldsymbol{b}^{\mathrm{s}}_k, \boldsymbol{b}^{\mathrm{o}}_k\}| = 2 \times |P|$, which is in general larger than $|E|$.

To encourage the refined BB of each entity to keep its involved relations, we use the relation loss $\mathcal{L}_{rel}$ (Sect. 3.3) in a supervised manner. This is because $\mathcal{L}_{rel}$ indicates the degree of retaining the involved relations in terms of relation-unit.

For entity $e_i \in E$, let $\boldsymbol{B}_i = \{\boldsymbol{B}_{i\nu}\}$ denote the set of its corresponding BBs (appearing in different relation-units) and $\beta_i = \{\beta_{i\nu}\}$ be the set of their weights. We define the refined BB: $\hat{\boldsymbol{B}}_i = \frac{\sum_{\nu=1}^{|\boldsymbol{B}_i|}\{(1+\beta_{i\nu}) \times \boldsymbol{B}_{i\nu}\}}{\sum_{\nu=1}^{|\boldsymbol{B}_i|}(1+\beta_{i\nu})}$.

Each weight in β_i is obtained from the outputs of the softmax function in the relation auxiliary classifier using the relation loss $\mathcal{L}_{\mathrm{rel}}$.

At the beginning of training, relation-units cannot exactly reproduce their involved relations. Their weights thus tend to be close to *zero*, leading $\hat{\boldsymbol{B}}_i$ above almost similar to the simple average. Our refined BBs may be close to those of [1,10,11] at the beginning of training yet they keep their relations thanks to their weights. As training proceeds, the contribution of the relation-units retaining relations consistent with text t to the refined BB gradually increases. As a result, the location and size of the refined BB are continuously altered to keep relations consistent with t.

For entity e_i, its embeddings that are associated with $\{\boldsymbol{B}_{i\nu}\}$'s over ν are averaged. In this way, we obtain the set of refined BBs $\{\hat{\boldsymbol{B}}_i\}$ and their associated embeddings for all the entities in E. We remark that $|\{\hat{\boldsymbol{B}}_i\}| = |E|$.

If all the initial BBs completely keep their relations, the individual usage subnet works as the averaging operator as in [1,10,11] and our visual-relation layout is similar to the layout by [1,10,11]. In practice, however, the comprehensive usage of relations cannot guarantee to completely keep the relations. Our individual usage subnet plays the role of adjusting all the BBs in location and size to keep their relations as much as possible using each relation separately.

RefinedBB2layout Subnet. In order to construct the visual-relation layout, we aggregate all the refined BBs and transfer them from the bounding-box domain to the image domain. This process should meet two requirements: (i) each entity in the image should be localized and resized to match its individual refined BB, and (ii) each entity should appear in the image even if some refined BBs overlap with each other. To this end, we design *refinedBB2layout* subnet as a learnable network rather than the putting-together operation. We build this subnet using a conv-LSTM [20] with the 5 hidden states each outputting 128 channels.

For $\hat{\boldsymbol{B}}_i$ of entity e_i, we first convert it to the binary mask with the size of $64 \times 64 \times 128$ whose element is 1 if and only if it is contained in $\hat{\boldsymbol{B}}_i$, 0 otherwise. Then, we reshape its associated embedding from 1×128 to $1 \times 1 \times 128$. Finally, the reshaped embedding is wrapped to $\hat{\boldsymbol{B}}_i$ using the bilinear interpolation [21] for the layout of entity e_i ($64 \times 64 \times 128$). To produce $\theta(t)$, we feed the sequence of entity layouts into the *refinedBB2layout* subnet. The size of $\theta(t)$ is $64 \times 64 \times 128$.

3.2 Stacking-GANs

We condition three GANs, namely stacking-GANs, on $\theta(t)$ to progressively generate coarse-to-fine images with the size of $n \times n \times 3$ ($n = 64, 128, 256$). Each GAN is identical to CRN [18]. Parameters are not shared by any GANs.

The first GAN generator receives the layout $\theta(t)$ and a standard Gaussian distribution noise as input while the others receive the bilinear upsampled [21] layout $\theta(t)$ and the output of the last refinement layer from the previous GAN. The discriminators receive an image-layout pair as their inputs. Each pair is either a real sample or a fake sample. A real sample consists of a real image and a real layout while a fake one consists of a predicted layout and a generated or real image. These samples not only encourage the GAN to improve the quality of generated images but also give the helpful feedback to the layout predictor.

3.3 Loss Function

Relation Loss: $\mathcal{L}_{\text{rel}}$ is a cross entropy between relation-units and their GT relations that is obtained by a relation auxiliary classifier. The classifier is built upon two fully-connected layers producing 512 and $|\mathcal{R}|$ outputs. The first layer is followed by a ReLU layer while the second one ends with the $softmax$ function.

For each edge k of (E, P), its relation-unit and involved embeddings, i.e., $\boldsymbol{e}'^{\text{s}}_k$, $\boldsymbol{b}^{\text{s}}_k$, $\boldsymbol{e}'^{\text{o}}_k$, and $\boldsymbol{b}^{\text{o}}_k$, are concatenated in this order to have an input vector of 1×264. We then feed this vector into the relation auxiliary classifier to obtain the probability distribution $\boldsymbol{w}_k$ of the relations over $\mathcal{R}$. $\boldsymbol{w}_k$ is a vector of $1 \times |\mathcal{R}|$ and contains all the predicates $p_k \in \mathcal{R}$. We first obtain the *index* of predicate $p_k \in \mathcal{R}$. Since the order of predicates in $\boldsymbol{w}_k$ is the same as that in $\mathcal{R}$, the value at *index* in $\boldsymbol{w}_k$ is the weight of p_k, which is used as the weight of the relation-unit $(\boldsymbol{b}^{\text{s}}_k, \boldsymbol{b}^{\text{o}}_k)$ in the individual usage subnet. Note that the weight of a relation-unit is used for the weight of both $\boldsymbol{b}^{\text{s}}_k$ and $\boldsymbol{b}^{\text{o}}_k$ involved in the relation-unit.

The relation loss is defined as: $\mathcal{L}_{\text{rel}} = -\sum_{k=1}^{|P|}\sum_{\nu'=1}^{|\mathcal{R}|} \boldsymbol{p}_k[\nu'] \log(\boldsymbol{w}_k[\nu'])$. Minimizing the relation loss encourages relation-units to adjust their locations and sizes to meet the "predicate" relation. This is because the relation reflects the relative spatial locations among its associated relation-units.

Pixel Loss: $\mathcal{L}_{\text{pix}} = ||I - \hat{I}||_2$, where I is the ground-truth image and $\hat{I}$ is a generated image. The $\mathcal{L}_{\text{pix}}$ is useful for keeping the quality of generated images.

Contextual Loss [22]**:** $\mathcal{L}_{\text{context}} = -\log(CX(\Phi^l(I), \Phi^l(\hat{I})))$, where $\Phi^l(\cdot)$ denotes the feature map extracted from layer l of perceptual network Φ, and $CX(\cdot)$ is the function that computes the similarity between image features. $\mathcal{L}_{\text{context}}$ is used to learn the context of an image since refined BBs may lose the context such as missing pixel information or the size of entity.

Adversarial Loss [4]**:** $\mathcal{L}_{\text{adv}}$ encourages the stacking-GANs to generate realistic images. Since the discriminator also receives the real/predicted layout as its input, the $\mathcal{L}_{\text{adv}}$ is helpful in training the visual-relation layout module as well.

In summary, we jointly train our network in an end-to-end manner to minimize: $\mathcal{L} = \lambda_1\mathcal{L}_{\text{rel}} + \lambda_2\mathcal{L}_{\text{pix}} + \lambda_3\mathcal{L}_{\text{context}} + \sum_{i=1}^{3}\lambda_4\mathcal{L}_{\text{adv}i}$, where λ_i are hyper-parameters. We compute $\mathcal{L}_{\text{adv}}$ at each level in the stacking-GANs, while $\mathcal{L}_{\text{pix}}$ and $\mathcal{L}_{\text{context}}$ are computed at the third GAN.

4 Experiments

4.1 Dataset and Compared Methods

Dataset. We conducted experiments on challenging COCO-stuff [13] and Visual GENOME [14] datasets, which have complex descriptions with many entities and relations in diverse context. We followed [1] to pre-process all the datasets: $|\mathcal{C}| = 171$ and $|\mathcal{R}| = 6$ (COCO-stuff [13]), and $|\mathcal{C}| = 178$ and $|\mathcal{R}| = 45$ (GENOME [14]).

Compared Methods. We employed Johnson+ [1] as the baseline (64×64). To factor out the influence of image generator, we replaced the CRN in [1] by our stacking-GANs to produce higher resolution images (128×128 and 256×256). We also compared our method with Hong+ [12], Zhang+ [6], Xu+ [9], Li+ [10], Ashual+ [11], Zhao+ [16], and Sun+ [17]. We reported the results in the original papers whenever possible. For the methods that released at least one reference pre-trained model [23] and [24], we trained authors' provided codes (Zhang+ [6] and Xu+ [9]) on GENOME dataset.

Evaluation Metrics. We use the inception score (IS) [25], and Fréchet inception distance (FID) [26] to evaluate the overall quality of generated images (implemented in [27,28]). We also use four metrics to evaluate the layout: the entity recall at IoU threshold ($R@\tau$), the relation IoU ($rIoU$), the relation score (RS) [29], and the BB coverage. To evaluate the relevance of generated images and input text descriptions, we use the image caption metrics: $BLEU$ [30], $METEOR$ [31], and $CIDEr$ [32]. For the diversity of generated images, we use the diversity score [33] (implemented in [34]).

To evaluate how much the predicted layout is consistent with the ground-truth (GT), we measure the agreement in size and location between predicted (i.e., refined) and GT BBs using the entity recall at IoU threshold: $R@\tau = |\{i \mid IoU(\hat{\boldsymbol{B}}_i, \boldsymbol{GT}_i) \geq \tau\}|/N$, where $\hat{\boldsymbol{B}}_i$ and $\boldsymbol{GT}_i$ are predicted and GT BBs for entity e_i, $N = \min(|\{\hat{\boldsymbol{B}}_i\}|, |\{\boldsymbol{GT}_i\}|)$ (we always observed $|\{\hat{\boldsymbol{B}}_i\}| = |\{\boldsymbol{GT}_i\}|$), τ is a IoU threshold, and $IoU(\cdot)$ denotes Intersection-over-Union metric. Note that we used only the BBs that exist in both $\{\hat{\boldsymbol{B}}_i\}$ and $\{\boldsymbol{GT}_i\}$ to compute $R@\tau$.

We also evaluate the predicted layout using *subject–predicate–object* relations. For each *subject–predicate–object* relation, we computed the IoU of the predicted "subject" BB and its corresponding GT, and that for the "object". We then multiplied the two IoUs to obtain the IoU for the relation. $rIoU$ is the average over all the *subject–predicate–object* relations.

We use the relation score (RS) [29] for COCO-stuff to evaluate the compliance of geometrical relation between predicted BBs. For each edge k of scene graph (E, P), we define $score(\hat{\boldsymbol{B}}^{\mathrm{s}}_k, \hat{\boldsymbol{B}}^{\mathrm{o}}_k) = 1$ if and only if the relative location between $\hat{\boldsymbol{B}}^{\mathrm{s}}_k$ and $\hat{\boldsymbol{B}}^{\mathrm{o}}_k$ satisfies the relation p_k, 0 otherwise. $RS = \sum_{k=1}^{|P|} score(\hat{\boldsymbol{B}}^{\mathrm{s}}_k, \hat{\boldsymbol{B}}^{\mathrm{o}}_k)/|P|$.

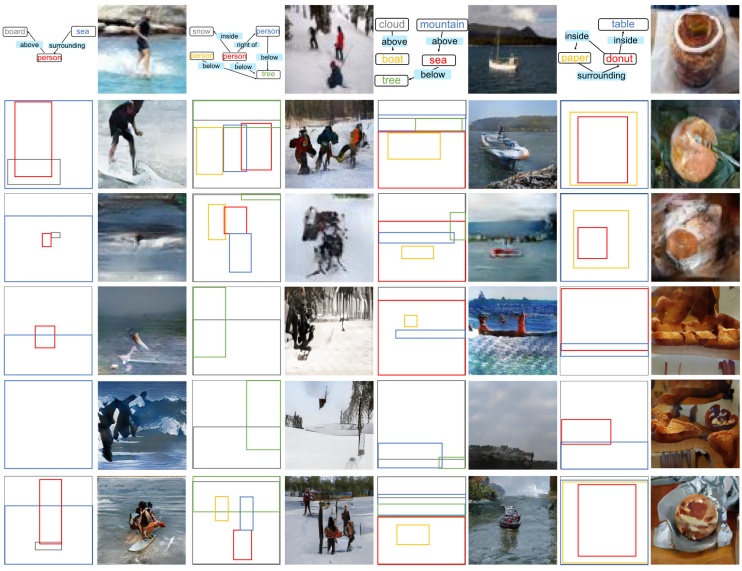
To evaluate how much BBs cover the area of the whole image, we compute the coverage of predicted BBs over the image area: $coverage = \bigcup_{i=1}^{|E|} \hat{\boldsymbol{B}}_i/(\text{image area})$.

We note that $R@\tau$ and $rIoU$ consider the consistency between predicted BBs and GT BBs, and RS and $coverage$ are independent of GT BBs. In other words, $R@\tau$ and $rIoU$ evaluate absolute locations of BBs while RS (and $coverage$ as well to some extent) does semantic relations. Therefore, they together effectively evaluate the layout in a wide range of aspects.

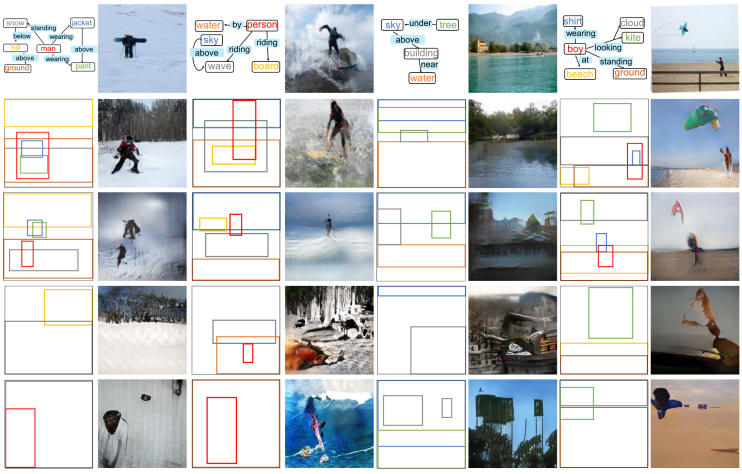
4.2 Implementation and Training Details

We optimized our model (built in PyTorch [35]) using the Adam optimizer with the recommended parameters [36] and the batch size of 16 for 500 epochs. We used VGG-19 [37] pre-trained on ImageNet as Φ, and $l = conv4\_2$ to compute $\mathcal{L}_{\mathrm{context}}$. Each model took about one week for training on a PC with GTX1080Ti $\times$ 2 while testing time was less than 0.5 s per structured-text input.

We trained the model except for the pre-processing in the end-to-end manner where we set $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1$, and do not pre-train each individual subset, meaning that we do not use any ground-truth BBs to train the visual-relation layout. The layout predictor receives signals not only directly from the relation loss but also from the other losses. In an early stage of the training, the rendering part cannot generate reasonable images because the quality of BBs is poor. This means the signals from losses are strong, leading to quick convergence of the layout predictor. As the training proceeds, the layout predictor properly works, and the rendering part gradually becomes better. $\mathcal{L}_{\mathrm{rel}}$, at that time, keeps the layout predictor stable and more accurate.



(a) COCO-stuff dataset [13].



(b) GENOME dataset. [14]

Fig. 4. Visual comparison on COCO-stuff and GENOME. For each example, we show the scene graph and reference image at the first row. From second to the last rows, we show the layouts and images generated by our method (256×256), Johnson+ [1] (64×64), Zhang+ [6] (256×256), Xu+ [9] (256×256), and Ashual+ [11] (256×256, COCO-stuff only, GT layout). The color of each entity BB corresponds to that in the scene graph. Zoom in for best view.

4.3 Comparison with State-of-the-Arts

Qualitative Evaluation. Figure 4 shows examples of the results obtained by our method and SOTAs [1,6,9,11] on COCO-stuff [13] and GENOME [14] datasets. It shows that the generated images by our method successfully preserve the scene structure given in text descriptions, indicating that our proposed visual-relation layouts are highly consistent with those of GTs. We see that the results by Johnson+ [1] have reasonable layouts, however, their layouts failed to keep all relations well and the visual impression of their results is not good. The results by Zhang+ [6] and Xu+ [9] are clear in (entities) details but they lose the scene structure (some entities disappear). The results by Ashual+ [11] (COCO-stuff only) are more impressive than ours to some extent, however, they use GT layout and pre-defined entities' appearances.

Quantitative Evaluation. We classify all the compared methods into three: (A) Johnson+ [1], Hong+ [12], Li+ [10], and Ashual+ [11] (which firstly infer a layout and then convert it to an image), (B) Zhang+ [6] and Xu+ [9] (which are directly conditioned on texts), and (C) Zhao+ [16] and Sun+ [17] (which are directly conditioned on ground-truth layouts).

Table 1 shows that our method (almost) outperforms (A) in IS and FID on both COCO-stuff and GENOME. In comparison with (B), our method achieves the best in FID on both the datasets, the best on GENOME and the second best on COCO-stuff in IS. Xu+ [9] achieves better IS on COCO-stuff than us because (i) Xu+ [9] focuses on generating images in good human perception based on entity information and (ii) COCO-stuff has less complex relations, in other words, layouts may be less important. On GENOME, however, text descriptions are more complex with many entities and relations, and their results are degraded due to poor layouts as seen later in Table 2. Table 1 also shows that the scores of our completed model are comparable to those of (C), meaning that our (predicted) visual-relation layout is close to the GT layout. When replacing the predicted layout by the GT (the 17th row), our results achieve the same level with (C). We thus conclude that our method is more effective than the others.

Next, we evaluated how the scene structure given in input text was preserved in generated images using $R@\tau$ (we changed τ from 0.3 to 0.9 by 0.2), $rIoU$, RS, and *coverage*, see Table 2. We remark that we computed RS only for COCO-stuff because COCO-stuff has geometrical relations only. For Zhang+ [6] and Xu+ [9], we employed Faster-RCNN [38] to estimate their predicted BBs of entities where we set the number of generated BBs to be the number of entities in an image. We note that the number of predicted BBs by ours or Johnson+ [1] was always the same with the number of entities in an image.

Table 2 shows that our method performs best, indicating that our predicted BBs more precisely agree with those in relation (location and size) of entities given in texts than the compared methods. To be more specific, $rIoU$'s in Table 2 show that our predicted BBs more successfully retain the relations of entities than the other methods. This observation is also supported by RS on COCO-stuff. Moreover, our method outperforms the others in *coverage* and achieves

Table 1. Comparison of the overall quality using *IS* and *FID*. From the 4th to the 16th rows: group (A) and (B) (the best in blue; the second best in red). From the 17th to the 19th rows: group (C) (**bold** indicates the best). Scores inside the parentheses indicate those reported in the original papers.

	IS ⇑						**FID** ⇓					
Dataset	**COCO-stuff** [13]			**GENOME** [14]			**COCO-stuff** [13]			**GENOME** [14]		
Image size	64 × 64	128 × 128	256 × 256	64 × 64	128 × 128	256 × 256	64 × 64	128 × 128	256 × 256	64 × 64	128 × 128	256 × 256
Ours w/o individual usage	7.02 ± 0.19	8.12 ± 0.41	9.95 ± 0.31	5.48 ± 0.16	5.66 ± 0.26	5.91 ± 0.41	63.28	59.52	55.21	72.42	72.02	71.49
Ours w/o weighted unification	7.10 ± 0.27	8.64 ± 0.37	10.49 ± 0.41	5.99 ± 0.22	6.61 ± 0.31	7.32 ± 0.37	61.89	57.20	49.16	69.37	60.89	57.18
Ours w/o refinedBB2layout	7.23 ± 0.20	8.70 ± 0.35	10.50 ± 0.37	6.11 ± 0.25	6.93 ± 0.29	7.87 ± 0.33	57.68	53.81	46.55	67.65	58.54	54.45
Ours w/o $\mathcal{L}_{\text{pix}}$	7.29 ± 0.17	9.26 ± 0.31	11.36 ± 0.40	6.05 ± 0.15	8.26 ± 0.27	8.66 ± 0.36	56.81	51.02	43.18	70.18	60.02	58.63
Ours w/o $\mathcal{L}_{\text{context}}$	7.56 ± 0.11	9.68 ± 0.33	11.47 ± 0.42	6.37 ± 0.16	8.41 ± 0.22	8.97 ± 0.31	50.89	47.22	40.10	68.20	56.39	53.75
Ours w/o $\mathcal{L}_{\text{adv}}$	7.31 ± 0.19	9.47 ± 0.34	11.41 ± 0.47	6.30 ± 0.19	8.39 ± 0.20	8.96 ± 0.39	56.24	50.87	41.05	68.34	57.23	53.86
Ours (completed model)	9.20 ± 0.32	12.01 ± 0.40	14.20 ± 0.45	7.97 ± 0.30	9.24 ± 0.41	11.75 ± 0.43	35.12	29.12	27.39	58.37	50.19	36.79
Johnson+ [1]	(6.70 ± 0.10)	7.13 ± 0.24	7.25 ± 0.47	(5.50 ± 0.10)	5.72 ± 0.33	5.81 ± 0.37	67.99	65.23	64.19	73.39	69.48	68.42
Hong+ [12]	—	(11.46 ± 0.09)	—	—	—	—	—	—	—	—	—	—
Li+ [10]	(9.40 ± 0.20)	—	—	(7.30 ± 0.20)	—	—	—	—	—	—	—	—
Ashual+ [11]	(7.90 ± 0.20)	(10.40 ± 0.40)	(14.50 ± 0.70)	—	—	—	(65.30)	(75.40)	(81.00)	—	—	—
Zhang+ [6]	7.79 ± 0.32	8.49 ± 0.52	(10.62 ± 0.19)	6.35 ± 0.16	6.44 ± 0.25	7.39 ± 0.38	87.21	85.37	78.19	108.68	86.17	77.95
Xu+ [9]	11.78 ± 0.14	19.11 ± 0.28	(25.89 ± 0.47)	6.38 ± 0.22	6.88 ± 0.32	8.20 ± 0.35	50.06	43.98	34.48	96.40	83.39	72.11
Ours with GT layout	$\mathbf{10.36 \pm 0.41}$	13.73 ± 0.59	$\mathbf{14.78 \pm 0.65}$	$\mathbf{8.87 \pm 0.57}$	10.04 ± 0.45	$\mathbf{12.03 \pm 0.37}$	**30.98**	**27.74**	**26.32**	45.63	40.96	**27.33**
Zhao+ [16] (GT layout)	(9.10 ± 0.10)	—	—	(8.10 ± 0.10)	—	—	—	—	—	—	—	—
Sun+ [17] (GT layout)	(9.80 ± 0.20)	$\mathbf{(13.80 \pm 0.40)}$	—	(8.70 ± 0.40)	$\mathbf{(11.10 \pm 0.60)}$	—	(34.31)	(29.65)	—	**(34.75)**	**(29.36)**	—
GT	16.25 ± 0.38	25.89 ± 0.47	32.61 ± 0.69	13.92 ± 0.42	21.43 ± 1.03	31.22 ± 0.65	—	—	—	—	—	–

Table 2. Comparison of the scene structure using $R@\tau$, $rIoU$, RS, and *coverage* (larger is better; the best in **bold**).

Dataset	COCO-stuff [13]							GENOME [2]					
Metric	$R@\tau$				$rIoU$	RS	*coverage*	$R@\tau$				$rIoU$	*coverage*
	0.3	0.5	0.7	0.9			GT = 98.24	0.3	0.5	0.7	0.9		GT = 77.10
Ours w/o individual usage	61.45	43.22	29.71	20.05	0.2652	53.48	94.96	26.48	14.29	11.90	9.81	0.1264	50.07
Ours w/o weighted unification	61.76	45.28	30.22	20.51	0.2795	56.27	95.07	29.57	18.22	13.76	10.80	0.1501	56.77
Ours (completed model)	**65.34**	**49.01**	**35.87**	**23.61**	**0.3186**	**68.23**	**97.19**	**35.00**	**23.12**	**16.34**	**13.40**	**0.1847**	**71.13**
Johnson+ [1]	59.75	42.53	29.23	19.89	0.2532	51.20	94.82	28.13	17.17	12.30	10.47	0.1485	52.28
Zhang+ [6]	37.81	20.50	10.64	7.76	0.0824	30.72	60.15	18.38	10.84	8.11	5.82	0.0643	40.07
Xu+ [9]	21.39	10.71	8.15	5.83	0.0671	31.97	52.76	16.02	9.33	7.66	5.15	0.0579	36.82

Table 3. Comparison using caption generation metrics on COCO-stuff (larger is better; the best in blue). Scores inside the parentheses indicate those reported in [12].

Method	$BLEU-1$	$BLEU-2$	$BLEU-3$	$BLEU-4$	$METEOR$	$CIDEr$
Ours	0.561	0.352	0.217	0.139	0.157	0.325
Johnson+ [1]	0.531	0.321	0.183	0.107	0.141	0.238
Hong+ [12]	(0.541)	(0.332)	(0.199)	(0.122)	(0.154)	(0.367)
Zhang+ [6]	0.417	0.214	0.111	0.062	0.095	0.078
Xu+ [9]	0.450	0.251	0.157	0.087	0.105	0.251
GT	0.627	0.434	0.287	0.191	0.191	0.367
	(0.678)	(0.496)	(0.349)	(0.243)	(0.228)	(0.802)

Table 4. Comparison using diversity score [33] (the best in blue; the second best in red). Scores are inside the parentheses indicates those in the original papers.

Method	COCO-stuff [13]	GENOME [2]
Ours (64 × 64)	0.36 ± 0.10	0.39 ± 0.09
Ours (128 × 128)	0.45 ± 0.12	0.49 ± 0.07
Ours (256 × 256)	0.52 ± 0.09	0.56 ± 0.06
Johnson+ [1]	0.29 ± 0.10	0.31 ± 0.08
Ashual+ [11]	(0.67 ± 0.05)	—
Zhao+ [16]	(0.15 ± 0.06)	(0.17 ± 0.09)
Sun+ [17]	(0.40 ± 0.09)	(0.43 ± 0.09)

comparable levels with the ground-truth BBs. These indicate that our visual-relation layout is well-structured. Our method thus has even better ability of rendering more realistic images with multiple entities since the faithful scene structure and more BB coverage (i.e., entity information) are achieved. Note that the observation that the *coverage*'s on COCO-stuff are better than those on GENOME explains the reason why generated images on COCO-stuff are better in IS and FID than those on GENOME.

Next, we use the image caption task to evaluate how the generated image is relevant to its input text. We follow [12] to report scores on COCO-stuff [13], see Table 3. Note that we evaluated on COCO-stuff only since the pre-trained image caption model on GENOME is not available. We also note that all the scores on the ground-truth dataset in [12] are higher than our re-computation. Table 3 shows that our method outperforms the others [1,6,9,12] on $BLEU$, $METEOR$ and comparable to [12] on $CIDEr$. We thus conclude that our method performs more consistently with input texts than the others.

Finally, we show the diversity score of generated images in Table 4. Overall, our scores are higher than Johnson+ [1], Zhao+ [16], and Sun+ [17] on both

COCO-stuff and GENOME, and comparable to Ashual+ [11] on COCO-stuff. Moreover, along with our stacking-GANs, our scores become better and better. These scores also support the efficacy of our method.

We note that the number of (trainable) parameters in our model is about 41M which is comparable with Johnson+ [1] (28M), Zhang+ [6] (57M), and Xu+ [9] (23M), and significantly smaller than Ashual+ [11] (191M).

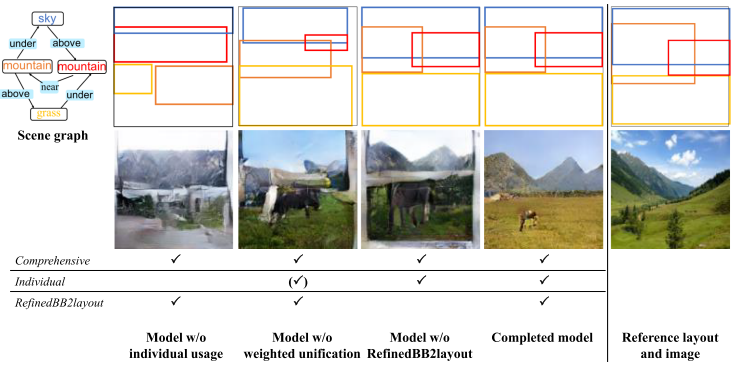


Fig. 5. Example of layouts and generated images by the ablation models. For each model, the 1st row shows the layout, the 2nd row shows the generated image. All images are at 256×256 resolution.

4.4 Ablation Study

We evaluated ablation models, see the first block of Tables 1 and 2: ours w/o individual usage denotes the model dropping the individual usage subnet; ours w/o weighted unification denotes the replacement of refining BBs with just averaging in the individual usage subnet; ours w/o refinedBB2layout denotes the replacement by just putting all entity layouts together in constructing the visual-relation layout. Figure 5 illustrates a typical output example of the ablation models. We note that model w/o comprehensive usage is not applicable since all the other subnets in our visual-relation layout module need the output by the comprehensive usage subnet.

The 4th and 5th rows of Tables 1 and 2 confirm the importance of the individual usage subnet. We also see the necessity of our learnable weights in refining BBs because model w/o weighted unification performs better than model w/o individual usage. We may conclude that the relation-unit prediction and the weighted unification together bring gain to our performance.

From Fig. 5, we visually observe that the layout by the model w/o individual usage does not successfully reflect relations. This observation is applicable to the model w/o weighted unification as well. As a result, both the models generated images in poorer quality than our complete model. The relation-units are in

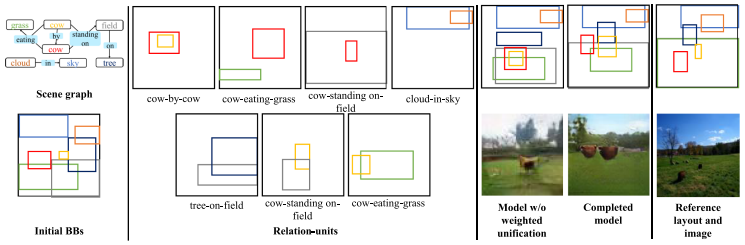


Fig. 6. Example of relation-units in the individual usage subnet; layouts and generated images by model w/o weighted unification and completed model.

Fig. 7. Example of output along with the stacking-GANs. From left to right, scene graph, visual-relation layout, the outputs at 64×64, 128×128, 256×256 resolutions, and the reference image.

diversity: entity BBs can be various in size and location because of multiple relations (see Fig. 6, for example), and thus simply averaging BBs corresponding to the same entity does not successfully retain the entity relations. The individual usage of relations is important for more consistent layout with input text.

The 6th row in Table 1 shows the significance of the refinedBB2layout. Complex descriptions with many entities and relations tend to produce overlapped BBs. The model w/o refinedBB2layout cannot necessarily produce all the entities in the layout, generating poor images.

We also evaluated the necessity of each term of the loss function through comparing our completed model with models dropping one term each: model w/o $\mathcal{L}_{\text{pix}}$, model w/o $\mathcal{L}_{\text{context}}$, and model w/o $\mathcal{L}_{\text{adv}}$ (we dropped each term in the loss function (Sect. 3.3) except for stacking-GANs). From the 2nd block of Table 1, we see that the absence of any term degrades the quality of generated images. This indicates that all the loss terms indeed contribute to performance.

Finally, we see that along with the stacking of GANs, our method progressively generates better images in terms of IS and FID (Table 1). We observe that at 64×64 resolution, generated images tend to be blurred and lose some details while the details of images are improved as the resolution becomes higher (the best result is obtained at 256×256 resolution) (see Fig. 7 as an example). We also confirmed that the visual-relation layouts of generated images at any resolutions are the same and highly consistent with texts.

When we replaced CRN in [1] with our stacking-GANs for 128×128 and 256×256 resolutions to factor out the influence of image generators, we see that the improvement of [1] on IS and FID along the resolution is worse than that of our model (the 10th and the 11th rows of Table 1). This indicates that better

layout significantly improves the performance of the final image generation and also confirms clearer contribution of our proposed visual-relation layout module.

5 Conclusion

We proposed a GAN-based end-to-end network for text-to-image generation where entity relations are comprehensively and individually used to infer a visual-relation layout. We also conditioned the stacking-GANs on the visual-relation layout to generate high-resolution images. Our layout preserves the scene structure more precisely than the layout by SOTAs.

References

1. Johnson, J., Gupta, A., Fei-Fei, L.: Image generation from scene graphs. In: CVPR (2018)
2. Johnson, J., Krishna, R., Stark, M., Li, J., Bernstein, M., Fei-Fei, L.: Image retrieval using scene graphs. In: CVPR (2015)
3. Li, Y., Ouyang, W., Zhou, B., Wang, K., Wang, X.: Scene graph generation from objects, phrases and region captions. In: ICCV (2017)
4. Goodfellow, I., et al.: Generative adversarial nets. In: NIPS (2014)
5. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text-to-image synthesis. In: ICML (2016)
6. Zhang, H., et al.: StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. In: ICCV (2017)
7. Reed, S., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: CVPR (2016)
8. Dong, H., Yu, S., Wu, C., Guo, Y.: Semantic image synthesis via adversarial learning. In: ICCV (2017)
9. Xu, T., et al.: AttnGAN: fine-grained text to image generation with attentional generative adversarial networks. In: CVPR (2018)
10. Li, Y., Ma, T., Bai, Y., Duan, N., Wei, S., Wang, X.: PasteGAN: a semi-parametric method to generate image from scene graph. In: CVPR (2019)
11. Ashual, O., Wolf, L.: Specifying object attributes and relations in interactive scene generation. In: ICCV (2019)
12. Hong, S., Yang, D., Choi, J., Lee, H.: Inferring semantic layout for hierarchical text-to-image synthesis. In: CVPR (2018)
13. Caesar, H., Uijlings, J., Ferrari, V.: COCO-stuff: thing and stuff classes in context. In: CVPR (2018)
14. Krishna, R., et al.: Visual genome: connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis. **123**, 32–73 (2017). https://doi.org/10.1007/s11263-016-0981-7
15. Reed, S., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H.: Learning what and where to draw. In: NIPS (2016)
16. Zhao, B., Meng, L., Yin, W., Sigal, L.: Image generation from layout. In: CVPR (2019)
17. Wei, S., Tianfu, W.: Image synthesis from reconfigurable layout and style. In: ICCV (2019)

18. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: ICCV (2017)
19. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: ICML (2010)
20. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W., Woo, W.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: NIPS (2015)
21. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: NIPS (2015)
22. Mechrez, R., Talmi, I., Zelnik-Manor, L.: The contextual loss for image transformation with non-aligned data. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 800–815. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_47
23. https://github.com/hanzhanggit/StackGAN-Pytorch
24. https://github.com/taoxugit/AttnGAN
25. Salimans, T., et al.: Improved techniques for training GANs. In: NIPS (2016)
26. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: NIPS (2017)
27. https://github.com/openai/improved-gan/tree/master/inception_score
28. https://github.com/bioinf-jku/TTUR
29. Tripathi, S., Bhiwandiwalla, A., Bastidas, A., Tang, H.: Using scene graph context to improve image generation. In: CVPRW (WiCV) (2019)
30. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: ACL (2002)
31. Lavie, A., Agarwal, A.: METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: ACL (2005)
32. Vedantam, R., Zitnick, C.L., Parikh, D.: CIDEr: consensus-based image description evaluation. In: CVPR (2015)
33. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
34. https://github.com/richzhang/PerceptualSimilarity
35. https://pytorch.org/
36. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: ICLR (2014)
37. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
38. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)