Context-Aware Layout to Image Generation with Enhanced Object Appearance

Sen He\textsuperscript{1,2\*}, Wentong Liao\textsuperscript{3\*}, Michael Ying Yang\textsuperscript{4}, Yongxin Yang\textsuperscript{1,2}, Yi-Zhe Song\textsuperscript{1,2}
Bodo Rosenhahn\textsuperscript{3}, Tao Xiang\textsuperscript{1,2}

\textsuperscript{1}CVSSP, University of Surrey, \textsuperscript{2}iFlyTek-Surrey Joint Research Centre on Artificial Intelligence
\textsuperscript{3}TNT, Leibniz University Hannover, \textsuperscript{4}SUG, University of Twente

Abstract

A layout to image (L2I) generation model aims to generate a complicated image containing multiple objects (things) against natural background (stuff), conditioned on a given layout. Built upon the recent advances in generative adversarial networks (GANs), existing L2I models have made great progress. However, a close inspection of their generated images reveals two major limitations: (1) the object-to-object as well as object-to-stuff relations are often broken and (2) each object’s appearance is typically distorted lacking the key defining characteristics associated with the object class. We argue that these are caused by the lack of context-aware object and stuff feature encoding in their generators, and location-sensitive appearance representation in their discriminators. To address these limitations, two new modules are proposed in this work. First, a context-aware feature transformation module is introduced in the generator to ensure that the generated feature encoding of either object or stuff is aware of other co-existing objects/stuff in the scene. Second, instead of feeding location-insensitive image features to the discriminator, we use the Gram matrix computed from the feature maps of the generated object images to preserve location-sensitive information, resulting in much enhanced object appearance. Extensive experiments show that the proposed method achieves state-of-the-art performance on the COCO-Thing-Stuff and Visual Genome benchmarks. Code available at: https://github.com/wtliao/layout2img.

1. Introduction

Recent advances in generative adversarial networks (GANs) \cite{72} have made it possible to generate photorealistic images for a single object, e.g., faces, cars, cats \cite{1, 2, 3, 4}. However, generating complicated images containing multiple objects (things) of different classes against natural backgrounds (stuff) still remains a challenge \cite{4, 2, 3, 7}. This is due to the large appearance variations

\*Equal contribution
an intermediate input step of other tasks, e.g., scene graph
and text to image generation [3, 15].

Since the seminal work [48] in 2019, the very recent
layout to image (L2I) generation models [49, 40, 39] have
made great progresses, thanks largely to the advances made
in GANs [30, 20] as they are the key building blocks. From
a distance, the generated images appear to be realistic and
adhere to the input layout (see Fig. 1 and more in Fig. 3).
However, a closer inspection reveals two major limitations.
First, the relations between objects and object-to-stuff are
often broken. This is evident from the food example in
Fig. 1 (Top-Middle) – the input layout clearly indicates that
the four bowls are overlapping with each other. Using the
state-of-the-art LostGAN-v2 [39], the occluded regions be-
tween objects are poorly generated. Second, each gener-
ated object’s appearance is typically distorted lacking class-
defining characteristics. For instance, the surfing example
in Fig. 1 (Middle) and the giraffe example in Fig. 1 (Bot-
tom) show that the object appearance has as if been touched
by Picasso – one can still recognize the surfing person or
giraffe, but key body parts are clearly misplaced.

We believe these limitations are caused by two major de-
sign flaws in existing L2I models in both their GAN gener-
ators and discriminators. (1) Lack of context-aware mod-
eling in the generator: Existing models generate the fea-
ture for the object/stuff in each layout bounding box first,
and then feed the generated feature into a generator for im-
ge generation. However, the feature generation process for
each object/stuff is completely independent of each other,
therefore offering no chance for capturing the inter-object
and object-to-stuff relations. (2) Lack of location-sensitive
appearance representation in the discriminator: As in any
GAN model, existing L2I models deploy a discriminator
that is trained to distinguish the generated whole image and
individual object/stuff images from the real ones. Such a
discriminator is essentially a CNN binary classifier whereby
globally pooled features extracted from the CNN are fed
to a real-fake classifier. The discriminator thus cares only
about the presence/absence and strength of each semantic
feature, rather than where they appear in the generated im-
ages. This lack of location-sensitive appearance representa-
tion thus contributes to the out-of-place object part problem
in Fig. 1 (Middle).

In this paper, we provide solutions to overcome both lim-
itations. First, to address the lack of context-aware mod-
eling problem, we propose to introduce a context-aware
feature transformation module in the generator of a L2I
model. This module updates the generated feature for each
object and stuff after each has examined its relations with
all other objects/stuff co-existing in the image through self-
attention. Second, instead of feeding location-insensitive
globally pooled object image features to the discriminator,
we use the Gram matrix computed from the feature maps of
the generated object images. The feature map Gram matrix
captures the inter-feature correlations over the vectorized
feature map, and is therefore locations sensitive. Adding it
to the input of the real-fake classifier in the discriminator,
the generated images preserve both shape and texture char-
acteristics of each object class, resulting in much enhanced
object appearance (see Fig. 1 (Right)).

The contributions of this work are as follows: (1) For
the first time, we identify two major limitations of exist-
ing L2I models for generating complicated multi-object im-
ages. (2) Two novel components, namely a context-aware
feature transformation module and a location-sensitive ob-
ject appearance representation are introduced to address
these two limitations. (3) The proposed modules can be
easily integrated into any existing L2I generation mod-
els and improve them significantly. (4) Extensive exper-
iments on both the COCO-Thing-Stuff [25, 5] and Vi-
sual Genome [22] datasets show that state-of-the-art per-
formance is achieved using our model. The code and trained
models will be released soon.

2. Related Work

Generative Adversarial Networks Generative adversar-
ial networks (GANs) [11], which play a min-max game be-
tween a generator and a discriminator, is the mainstream
approach used in recent image generation works. However,
the training of a GAN is often unstable and known to be
prone to the mode collapse problem. To address this, tech-
niques like Wasserstein GAN [2] and Unrolled GAN [28]
were developed. Meanwhile, noise injection and weight pe-
nalizing [1, 34] were used in the discriminator to alleviate
the non-convergence problem for further stabilization of the
training. To generate high fidelity and resolution images,
architectures like Progressive GAN [19] and BigGAN [4]
were also proposed.

Conditional Image Generation Conditional image gen-
eration, which generates an image based on a given condi-
tion (e.g., class label, sentence description, image, semi-
matic mask, sketch, and scene graph) has been studied inten-
sively [29, 31, 17, 51, 30, 7, 3, 9] due to its potential in gen-
erating complicated natural images. In general, there are
two popular architectures for the conditional image genera-
tion. The first one is the encoder-decoder architecture used
in Pix2pix [17] and CycleGAN [51], where the encoder di-
rectly takes the conditional input and embeds it to a latent
space. The decoder then transfers the embedded representa-
tion into the target image. The second popular architecture
is the decoder-only architecture used in StyleGAN [20] and
GauGAN [30], where a decoder starts with a random input,
and then progressively transforms it to produce the desired
output. In this architecture, the conditional input is used to
generate part of the parameters in the decoder, e.g., the
affine transformation parameters in the normalization layers [20, 30, 33] or the weight parameters in convolutional kernels [26].

**Layout to Image Generation** Though the previous work [15] has already touched the concept of layout to image generation (L2I), it is just used as an intermediate step for a different generation task. The first stand-alone solution appeared in [48]. Compared to other conditional inputs such as text and scene graph, layout is a more flexible and richer format. Therefore, more studies followed up by introducing more powerful generator architectures [38, 39], or new settings [23, 27]. Sun et al. [38] proposed a new architecture inspired by StyleGAN [20], which allows their model to generate higher resolution images with better quality. Li et al. [23] introduced a new setting for high resolution street scene generation. Their model retrieves a background from a database based on the given foreground layout. Recently, Ma et al. [27] introduced attribute guided layout generation, which is more controllable on the generated objects. As mentioned earlier, all these existing models have two limitations, namely lack of context-aware modeling in their generators, and lack of location-sensitive appearance representation in their discriminators. Both limitations are overcome in this work, resulting in much improved L2I generation performance (see Sec. 5).

**Context Modeling** Context plays an important role in many discriminative scene analysis tasks [41, 16, 8, 45, 44, 13, 43]. The main idea in context-based analysis is to tie each object instance in the scene with the global context, such that their relationship or interaction can be better understood. However, context has drawn little attention in image generation. One exception is SAGAN [46] which applied self-attention to refine the feature map in the generator for single object image generation. In this work, we introduce context modeling for layout to image generation, a more complicated image generation task with a focus on inter-objects and object-to-stuff relation modeling.

**Appearance Representation in CNNs** Works on CNN visualization clear show that feature channels, especially those at the top layers of a CNN capture semantically meaningful concepts such as body parts; and the activations of these feature channel at different locations indicate where these concepts are [50]. However, when it comes to object recognition [35] or real-fake discriminator in GAN [11], these feature maps are globally pooled before being fed into a binary classification layer. Location-sensitive information is thus largely lost, and the focus is on the presence/absence of the semantic concepts rather than where. We therefore propose to use the Gram matrix computed on the feature maps to complement the semantics-only appearance representation used in existing discriminators in order to induce location-sensitivity in object image generation. Such a Gram matrix based appearance representation has been used in style transfer [10] for style/texture representation, which seems to suggest that it only captures feature distribution but contains no spatial information. However, as pointed out in [24], this is because the use of entry-wise mean-square distance in [10] removes the location sensitivity in the feature map Gram matrix. In our model, we pass the raw matrix instead of mean-square distance to the discriminator classifier, therefore preserving the location sensitivity.

### 3. Preliminaries

#### 3.1. Problem Definition

Let $L = \{(y_i, b_i)_{i=1}^N\}$ be a layout with $N$ bounding boxes, where $y_i \in C$ is the class of the bounding box and $b_i = [x_i, y_i, w_i, h_i]$ is the position and size of the bounding box in the image lattice ($H \times W$). The goal of the layout to image (L2I) generation task is to build a model $G$, which can generate a realistic photo $I_g \in \mathbb{R}^{3 \times H \times W}$, given the coarse information in the layout $L$.

#### 3.2. Prior Models

Before introducing our proposed method in Sec. 4, we first briefly describe prior L2I models. In all previous models, the first step is always to generate a feature representation for each bounding box based on their classes:

$$p_i = \phi_0([e_i, n_i]),$$  

where $p_i \in \mathbb{R}^{d_l+d_n}$ is the feature representation of the $i^{th}$ bounding box in the layout, $\phi_0$ is a linear transformation layer, $e_i \in \mathbb{R}^{d_l}$ is the label embedding of $y_i$, and $n_i \in \mathbb{R}^{d_n}$ is a random noise sampled from a zero-mean unit-variance multivariate Gaussian distribution. The generated feature vector set $\{p_i\}_{i=1}^N$ is then fed into a generator $G$ for image generation. Depending on how the generator uses the feature vector set to generate the image, the existing models can be grouped into the following two categories.

#### L2I Models with Encoder-Decoder Generators

These models deploy an encoder-decoder generator [48, 27] which takes the feature vector set as input, and then transfers the feature vector set into a sequence of feature maps. Each feature map is generated by filling the corresponding feature vector into the region in the image lattice based on their bounding box. The generated feature maps are then fed into an encoder, which embeds each feature map into a latent space separately. Those embedded feature maps are merged into a single one through a convolutional LSTM network [37]. Finally, a decoder transforms the combined feature into the target image. Mathematically, the encoder-decoder based method can be formulated as:

$$I_g = D(cLSTM(E(\{F(p_i, b_i)\}_{i=1}^N))),$$  

where $E$ and $D$ are encoder and decoder, respectively.
where \( \mathcal{F}(\cdot, \cdot) \) is a filling operation, \( E \) is the encoder, cLSTM is the convolutional LSTM network, and \( D \) is the decoder.

### L2I Models with Decoder-Only Generators

These models [38, 39, 40] use a decoder-only generator to first generate an auxiliary mask\(^1\) for each bounding box for a fine-grained shape or structure prediction:

\[
\mathcal{M}_i = \mathcal{R}_S(\psi(\mathbf{p}_i), b_i),
\]

where \( \psi \) is a small convolutional neural network, \( \psi(\mathbf{p}_i) \in \mathbb{R}^{H \times W} \), and \( \mathcal{R}_S(\cdot, \cdot) \) is a resize operator, which resizes each generated mask and fit it to the corresponding region in the image lattice via up/down sampling. Then the decoder receives a zero-mean unit-variance multivariate random noise \( \mathbf{n}_0 \in \mathbb{R}^{C_0 \times H_0 \times W_0} \) as input, and decode it into the target image by modulating the affine transformation in the normalization layer:

\[
\hat{f}_l = \text{BatchNorm}(f_l, \varphi_l(\sum_{i=1}^N \mathbf{p}_i \otimes \mathcal{M}_i)),
\]

where \( \hat{f}_l \) and \( f_l \) are the feature maps before and after normalization at the \( l^{th} \) layer in the decoder, \( \varphi_l \) is a small convolutional block to generate the pixel-wise affine transformation parameters, \( \mathcal{M}_i \) is the resized version of \( \mathcal{M}_i \) to match the corresponding feature map’s scale, and \( \otimes \) is the outer product, by which a vector \( \mathbf{p}_i \) and a matrix \( \mathcal{M}_i \) produce a 3D tensor.

### 4. The Proposed Method

The main architecture of our proposed method is illustrated in Fig. 2. The proposed context-aware feature transformation module and location-sensitive Gram matrix based object appearance representation are integrated into the generator and discriminator respectively of a decoder-only L2I generation architecture [38, 39, 40]. Similarly they can be easily integrated with those employing an encoder-decoder architecture [48, 27].

#### 4.1. Context-Aware Feature Generation

Let us first look at the feature transformation module. It is clear that the prior models process each bounding box independently (either in the feature generation stage or the mask generation stage in the decoder-only methods), disrespecting the other objects and stuff in the scene. As a result, the generated objects do not appear in harmony with other co-existing objects and stuff in the scene and often appear to be out of place (see Fig. 1 and Fig. 3). To overcome this limitation, we propose a context-aware transformation module, which integrates contextual information into the feature representation of each bounding box by allowing each feature to cross-examine all other features via self-attention [42]. Concretely, the contextualized representation of each bounding box is computed as:

\[
\mathbf{p}^c_i = \sum_{j=1}^N w_{i,j} \mathbf{p}_j \mathbf{W}_v,
\]

\[
w_{i,j} = \frac{\exp(\alpha_{i,j})}{\sum_{k=1}^N \exp(\alpha_{i,k})},
\]

\[
\alpha_{i,j} = (\mathbf{p}_i \mathbf{W}_q) (\mathbf{p}_j \mathbf{W}_k)^T,
\]

where \( \mathbf{W}_q, \mathbf{W}_k \) and \( \mathbf{W}_v \in \mathbb{R}^{(d_l+d_n) \times (d_l+d_n)} \) are linear transformation layers. With the transformation, the contextualized representation of each bounding box not only has its own information, but also the global context in the layout. It is thus able to avoid the poor occlusion region generation problem shown in Fig. 1 (Top-Middle). Note that this module can be used for feature map filling in the encoder-decoder based methods, as well as the mask generation and the feature modulation steps in the decoder-only methods. The contextualized feature representation is then fed into the generator for image generation (see Fig. 2).

---

\(^1\)The mask is not a strictly binary mask, as it is the output of a layer with sigmoid activation.
4.2. Location-Sensitive Appearance Representation

To address the issue of lacking location-sensitive appearance representation in the discriminators of existing L2I models, we introduce a feature map Gram matrix based appearance representation. In existing models’ discriminators, the input image $I_{im}$ is first processed by a convolutional neural network $\psi_D$, and represented as $f_{im} \in \mathbb{R}^{C \times H_D \times W_D}$:

$$f_{im} = \psi_D(I_{im}). \quad (8)$$

Existing L2I models then apply two losses in the discriminator to train the whole model: an image-level loss $L_{im}$ according to the globally pooled feature of $f_{im}$, and an object-level conditional loss $L_o$ based on the ROI pooled [32] feature of each object in the image, concatenated with its corresponding class information. These losses are designed to boost the realism of the generated image and the objects in the image respectively. However, using pooled feature as appearance representation means that both losses are location-insensitive, i.e., they only care about the presence/absence and strength of each learned semantic feature; much less about where the corresponding visual concept appear in the image.

To address this problem, we propose to introduce an additional appearance discriminator loss, which directly penalizes the spatial misalignment of each semantic feature between the generated and real images. Concretely, we use object feature maps’ Gram matrix [10] as a new appearance representation and feed it to the discriminator classification layer. Formally, we define the appearance of a generated object in the image as:

$$A_i = s_i s_i^T / d_s, \quad (9)$$

where $d_s = C$ is the channel dimension of the feature map, $s_i \in \mathbb{R}^{C \times (H_D \times W_D)}$ is the spatial dimension vectorized feature representation of the $i^{th}$ generated object in the image, computed as:

$$s_i = \mathcal{R}_A(f_{im}, b_i), \quad (10)$$

where $\mathcal{R}_A(\cdot, \cdot)$ is the ROI align operator [12]. For simplicity, the vectorization operation is omitted here. The new appearance loss is then defined as:

$$L_A(G, D) = \mathbb{E}_{A^r \sim p_{data}(A^r)}[\log(D(A^r|y))] + \mathbb{E}_{A^g \sim p_{data}(A^g)}[1 - \log(D(A^g|y))], \quad (11)$$

where $A^r$ and $A^g$ are the Gram matrices of object feature maps in real and generated images respectively, $y$ is their corresponding class label. More specifically, for the $i^{th}$ object in an image, its appearance loss is computed as:

$$D(A_i|y) = \frac{1}{C} \sum_{j=1}^{C} |A_{i,j}| \mathcal{E}(y_i) \mathcal{W}_A, \quad (12)$$

where $\mathcal{E}(y_i) \in \mathbb{R}^k$ is the label embedding, and $\mathcal{W}_A \in \mathbb{R}^{(C+k) \times 1}$ is a linear layer. The Gram matrix here captures the correlation between different feature channels and is clearly location-sensitive: each entry only assumes a large value when the corresponding two features are both present and activated at the same location. This loss is thus complementary to the two conventional losses ($L_{im}$ and $L_o$) which emphasize the presence of the semantics only.

4.3. Training Objectives

The final model is trained with the proposed appearance loss, together with image and object level losses [48, 38]:

$$G^* = \arg \min_G \mathbb{E}_{D \sim p_{data}} \sum \lambda_{im} L_{im}(G, D) + \lambda_o L_o(G, D), \quad (13)$$

where $\lambda_{im}$ and $\lambda_o$ are the loss weight hyperparameters, and $L_{im}$ and $L_o$ are computed as:

$$L_{im}(G, D) = \mathbb{E}_{I_{im} \sim p_{data}(I_{im})}[\log(D(I_{im})^r)] + \mathbb{E}_{I_{im} \sim p_{data}(I_{im})}[1 - \log(D(I_{im})^g)], \quad (14)$$

$$L_o(G, D) = \mathbb{E}_{O^r \sim p_{data}(O^r)}[\log(D(O^r|y))] + \mathbb{E}_{O^g \sim p_{data}(O^g)}[1 - \log(D(O^g|y))],$$

where $I_{im}^r$ and $I_{im}^g$ are real and generated images respectively, and $O^r$ and $O^g$ are objects in the real and generated images.

5. Experiments

Datasets Two widely used benchmarks, COCO-Thing-Stuff [25, 5] and Visual Genome [22] are used in our experiments. COCO-Thing-Stuff includes bounding box annotations of the 91 stuff classes in [5] and the 80 thing/object classes in [25]. Following [48, 38], only images with 3 to 8 bounding boxes are used in our experiments. Visual Genome is originally built for complex scene understanding. The annotations in Visual Genome contain bounding boxes, object attributes, relationships, region descriptions, and segmentation. As per standard in L2I generation, we only use the bounding boxes annotation in our experiments, and each layout contains 3 to 30 bounding boxes. We follow the splits in prior works [48, 38] on both datasets to train and test our model.

Implementation Details Our model is implemented with PyTorch. To show the general applicability of our proposed method, and for fair comparison with prior works, we adopt both encoder-decoder and decoder-only generators in the two instantiations of our method (termed Ours-ED and Ours-D respectively). The encoder-decoder generator has the same architecture as used in [48], and the decoder-only generator shares the same architecture as used in [38]. Following [48, 38], the resolution of generated images is $64 \times 64$ for the encoder-decoder generator and $128 \times 128$ for the decoder-only generator. The learning rate is set to
Table 1. Comparative results on COCO-Thing-Stuff and Visual Genome. E-D means encoder-decoder based generator, D means decoder-only based generator. † means improved decoder-only generator.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Resolution</th>
<th>Generator</th>
<th>Inception Score</th>
<th>FID</th>
<th>Diversity Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>COCO</td>
<td>VG</td>
<td>COCO</td>
</tr>
<tr>
<td>Real images</td>
<td>64 × 64</td>
<td>-</td>
<td>16.3 ± 0.4</td>
<td>13.9 ± 0.5</td>
<td>-</td>
</tr>
<tr>
<td>Real images</td>
<td>128 × 128</td>
<td>-</td>
<td>22.3 ± 0.5</td>
<td>20.5 ± 1.5</td>
<td>-</td>
</tr>
<tr>
<td>pix2pix [17]</td>
<td>64 × 64</td>
<td>E-D</td>
<td>3.5 ± 0.1</td>
<td>2.7 ± 0.02</td>
<td>121.97</td>
</tr>
<tr>
<td>Layout2im [48]</td>
<td>64 × 64</td>
<td>E-D</td>
<td>9.1 ± 0.1</td>
<td>8.1 ± 0.1</td>
<td>38.14</td>
</tr>
<tr>
<td>Ours-ED</td>
<td>64 × 64</td>
<td>E-D</td>
<td>10.27 ± 0.25</td>
<td>8.53 ± 0.13</td>
<td>31.32</td>
</tr>
<tr>
<td>Grid2im [3]</td>
<td>128 × 128</td>
<td>E-D</td>
<td>11.22 ± 0.15</td>
<td>-</td>
<td>63.44</td>
</tr>
<tr>
<td>LostGAN-v1 [38]</td>
<td>128 × 128</td>
<td>D</td>
<td>13.8 ± 0.4</td>
<td>11.1 ± 0.6</td>
<td>29.65</td>
</tr>
<tr>
<td>LostGAN-v2 [49]</td>
<td>128 × 128</td>
<td>D†</td>
<td>14.21 ± 0.4</td>
<td>10.71 ± 0.76</td>
<td>24.76</td>
</tr>
<tr>
<td>OC-GAN [40]</td>
<td>128 × 128</td>
<td>D</td>
<td>14.0 ± 0.2</td>
<td>11.9 ± 0.5</td>
<td>36.04</td>
</tr>
<tr>
<td>AG-Layout2im [27]</td>
<td>128 × 128</td>
<td>E-D</td>
<td>-</td>
<td>8.5 ± 0.1</td>
<td>-</td>
</tr>
<tr>
<td>Ours-D</td>
<td>128 × 128</td>
<td>D</td>
<td>15.62 ± 0.05</td>
<td>12.69 ± 0.45</td>
<td>22.32</td>
</tr>
</tbody>
</table>

Evaluation Metrics We evaluate our model both automatically and manually. In automatic evaluation, we adopt three widely used metrics, namely Inception Score [36], Fréchet Inception Distance (FID) [14] and Diversity Score [47]. Inception Score evaluates the quality of the generated images. FID computes the statistical distance between the generated images and the real images. Diversity Score compares the difference between the generated image and the real image from the same layout. Following prior evalu-
Figure 4. Qualitative ablative experimental results. Regions with clear generation quality differences are highlighted using red dashed boxes for close examination.

Figure 5. The preference rate of our model. A bar higher than dark dashed horizontal line indicates that our model is judged to be better than the compared baseline by the AMT workers.

**Main Results**

We compare our method with existing L2I models [38, 39, 40, 27], the pix2pix model [17] which takes the input feature maps constructed from layout as implemented in [48], and the Grid2Im model [3] which receives scene graph as input. The following observations can be made on the quantitative results shown in Table 1.

1. Our method outperforms all compared methods on all benchmarks with both architectures and under all three automatic evaluations metrics, particularly for Inception Score and FID.
2. The more recent L2I methods take a decoder-only generator. Taking the same architecture but with the
two new components, our method (Ours-D) achieves new state-of-the-art. Fig 5 shows detailed statistics in the human evaluation on AMT. Among all 40 evaluation sets, our model won 32 sets. The preference rate is clearly higher at the higher resolution with more complex images (i.e., 128×128, VG dataset). Some qualitative results are shown in Fig. 3. It is evident from these examples that the images generated using our method are much more context-aware, i.e., different objects co-exist in harmony with each other and the background. Importantly, each generated object has a sharper texture, clearer shape boundary with respect to background inside the object bounding box, and overall much more spatially-coherent than those generated by existing L2I models.

**Ablation Study**  In this experiment, we adopt LostGAN-v1 [38] as our baseline and evaluate the effects of introducing our context transformation module and location-sensitive appearance representation. The quantitative results are shown in Table 2. We can see that both our context-aware feature transformation module and new appearance representation improve the baseline significantly on their own and when combined give a further boost. Some qualitative results are shown in Fig. 4. It is clear that the model trained with our appearance representation can generate objects with much better appearance both in terms of shape and texture (TV in Fig. 4(b) and person in Fig. 4(f)(g)). Context transformation also plays an important role: the generated occluded regions become more natural (Fig. 4(b)(f)); each object’s pose is also more in tune with its surrounding objects and background, e.g. the surfing person’s body pose is more physically plausible in Fig. 4(a); so is the person’s head pose in the presence of the laptop in Fig. 4(f).

**How Our Context Transformation Module Works**  In the decoder-only generator, a mask is generated using the representation of each bounding box to predict the fine-grained shape or structure of the object in each bounding box (Eq. 3). Without the context information in the feature representation, the generated masks would interfere with each other. This could result in irregular or incomplete object shape particularly in the occluded regions, which would further affect the feature modulation defined in Eq. 4. We investigate this effect by adding more bounding boxes into a layout, and visualizing the predicted masks as well as the generated images. The visualization results in Fig. 6 show clearly that the context-aware feature transformation module reduced the negative inter-object appearance interference in a complex scene when occlusion exists, yielding better appearance for the generated objects.

![Table 2. Ablation study on COCO-Thing-Stuff dataset.](Image)

<table>
<thead>
<tr>
<th>baseline [38]</th>
<th>context</th>
<th>appearance</th>
<th>Inception Score</th>
<th>FID</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>13.8 ± 0.4</td>
<td>29.65</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td>14.97 ± 0.27</td>
<td>24.05</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>15.28 ± 0.24</td>
<td>21.73</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td><strong>15.62 ± 0.05</strong></td>
<td>22.32</td>
</tr>
</tbody>
</table>

**6. Conclusion**

In this work, we proposed a novel context feature transformation module and a location-sensitive appearance representation to improve existing layout to image (L2I) generation models. In particular, they are designed to address existing models’ limitations on lacking context-aware modeling in their generator and spatially sensitive appearance representation in their discriminator. Extensive experiments demonstrate the effectiveness of our method, yielding new state-of-the-art on two benchmarks.

**Acknowledgment**

This work was supported by the Center for Digital Innovations (ZDIN), Federal Ministry of Education and Research (BMBF), Germany under the project LeibnizKILabor (grant no.01DD20003) and the Deutsche Forschungsgemeinschaft (DFG) under Germany’s Excellence Strategy within the Cluster of Excellence PhoenixD (EXC 2122).

![Figure 6. Qualitative examples about the contribution of context transformation in the complex scene generation. From left to right, at each time, we add one more bounding box into the previous layout, visualizing the predicted masks as well as the generated image by a model with our context transformation (Ours-D), and the same model without context transformation. Regions to pay more attention to are highlighted in dashed boxes.](Image)
References


[31] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In IJCV, 2016. 1, 2


[34] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Bernt Schiele. Generative adversarial nets. In NeurIPS, 2017. 4


network: A machine learning approach for precipitation nowcasting. In NeurIPS, 2015. 3
[38] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In ICCV, 2019. 1, 3, 4, 5, 6, 7, 8
[48] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In CVPR, 2019. 1, 2, 3, 4, 5, 6, 7