Supplemental Material for Attribute-Guided Image Generation from Layout

Ke Ma[1]
mark1123@cs.ubc.ca
Bo Zhao[4]
zhaobo.cs@gmail.com
Leonid Sigal[123]
lsigal@cs.ubc.ca

[1] University of British Columbia
Vancouver, Canada
[2] Vector Institute for AI
Toronto, Canada
[3] CIFAR AI Chair
[4] Bank of Montreal
Toronto, Canada

1 Approach Details

1.1 Layout Reconfiguration

In addition to image reconstruction path and image generation path, described in the main paper, layout reconfiguration path is introduced in our model to increase the spatial equivariance of the generator. Here we describe the layout reconfiguration path a little more completely. Similar to image generation path, an object latent code $\mathbf{z}_{obj_i}$ is sampled from a normal prior distribution $\mathcal{N}(0,1)$, and is concatenated to the object attribute embedding $M(\mathbf{w}_i \oplus \mathbf{e}_i)$. When composing the feature map F_i^{shift}, however, the input bounding boxes are randomly shifted. We limit ourselves to horizontal shifts in order to preserve coherence of the scene and not introduce perspective inconsistencies. Hence, each F_i^{shift} is composed based on the a new L^{shift}. Then, the set of F_i^{shift} feature maps are downsampled and passed to a cLSTM network to form the fused map H^{shift}, which is then decoded back to an image I^{shift}. The same image discriminator is applied to the generated image I^{shift}, and the object discriminator, the object classifier and the attribute classifier are applied to each generated object O^{shift} cropped based on the shifted bounding boxes $\mathbf{bbox}_i^{shift}$.

1.2 Discriminator

The structure of the discriminator D in our model follows the discriminator proposed in layout2im [40], but adds an additional term for the attributes (**4**):

(**1**) Image discriminator classifies the input image I as real and the generated image I^{rec}, I^{rand}, I^{shift} as fake.

(**2**) Object discriminator classifies the cropped objects O from I as real, and O^{rec}, O^{rand} and O^{shift} from I^{rec}, I^{rand}, I^{shift}, respectively, as fake.

(**3**) Auxiliary object classifier cls^{obj} predicts the category of cropped objects and is used to train the generator to synthesize correct objects. It is trained on real objects O and their labels ℓ.

(**4**) Auxiliary attribute classifier cls^{att} predicts the attribute of cropped objects and is used to train the generator to synthesize objects with correct attributes. It is trained on real objects O and their attributes $\mathcal{A}$.

1.3 Loss Function

Our model follows the Generative Adversarial Networks framework [4]. Namely, one image generator and two discriminators are jointly trained in minimax game:

$$\min_G \max_D \mathop{E}_{\mathbf{x}\sim p_{\mathbf{x}}}[\log D(\mathbf{x})] + \mathop{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[\log(1-D(G(\mathbf{z}))], \tag{1}$$

where $\mathbf{x}$ is the real image sampled from the data distribution $p(\mathbf{x})$ and $\mathbf{z}$ is the latent codes that generator uses to produce fake image. Since we have two separate discriminators for images and objects, there are two adversarial losses:

- **Image Adversarial Loss.** In each training iteration, our generator produces three images, which are: a generated image I^{rand}, a reconstructed image I^{rec} and a shifted image I^{shift}. Hence, the image adversarial loss $\mathcal{L}_{adv}^{\cdot}$ is defined as in Eq. (1) for all three types of generated images. By averaging the loss for I^{rec}, I^{shift}, I^{rec}, we obtain:

$$\mathcal{L}_{adv}^{img} = \frac{\mathcal{L}_{adv}^{I^{rand}} + \mathcal{L}_{adv}^{I^{rec}} + \mathcal{L}_{adv}^{I^{shift}}}{3} \tag{2}$$

  which generator G minimizes, and discriminator D maximizes.

- **Object Adversarial Loss.** We crop and resize objects O^{rand}, O^{rec} and O^{shift} from I^{rand}, I^{rec} and I^{shift}, respectively. By treating cropped objects as images, the object adversarial loss $\mathcal{L}_{adv}^{\cdot}$ is also defined as in Eq. (1):

$$\mathcal{L}_{adv}^{obj} = \frac{\mathcal{L}_{adv}^{O^{rand}} + \mathcal{L}_{adv}^{O^{rec}} + \mathcal{L}_{adv}^{O^{shift}}}{3} \tag{3}$$

In addition, we have another five losses to facilitate the generation of realistic images:

- **KL Loss.** $\mathcal{L}_{\text{KL}} = \sum_{i=1}^{o} E[\mathcal{D}_{\text{KL}}(Q(\mathbf{z}^r_{obj_i}|O_i)\|\mathcal{N}(\mathbf{z}^r_{obj}))]$ encourages the posterior distribution $Q(\mathbf{z}^r_{obj_i}|O_i)$ for object i to be close to the prior $\mathcal{N}(\mathbf{z}^r_{obj})$, for all of the o objects in the given image/layout.

- **Image Reconstruction Loss.** $\mathcal{L}_1^{img} = \|I - I^{rec}\|_1$ is the L1 difference between ground-truth image I and reconstructed image I^{rec} produced by the generator.

- **Object Latent Code Reconstruction Loss.** $\mathcal{L}_1^{latent} = \sum_{i=1}^{o} \|\mathbf{z}_{obj_i} - \mathbf{z}^{rand}_{obj_i}\|_1 + \|\mathbf{z}_{obj_i} - \mathbf{z}^{shift}_{obj_i}\|_1$ penalizes the $\mathcal{L}_1$ difference between the randomly sampled $\mathbf{z}_{obj_i} \sim \mathcal{N}(\mathbf{z}_{obj})$ and the re-estimated $\mathbf{z}^{rand}_{obj_i}$ and $\mathbf{z}^{shift}_{obj_i}$ from the generated objects O^{rand} and O^{shift}, respectively.

- **Auxiliar Object Classification Loss.** $\mathcal{L}_{AC}^{obj}$ is defined as the cross entropy loss from the object classifier. Cropped objects O^{real} with labels from real images are used to train the object classifier, and then the generator G is trained to generate realistic objects O^{rand}, O^{rec} and O^{shift} that minimize $\mathcal{L}_{AC}^{obj}$.

- **Auxiliar Attribute Classification Loss.** $\mathcal{L}_{AC}^{att}$ is defined as the weighted binary cross entropy loss from the attribute classifier. Similarly, real objects are used to train the classifier, and the generator G is trained to generate objects O^{rand}, O^{rec} and O^{shift} with correct attribute labels that minimize $\mathcal{L}_{AC}^{att}$.

Therefore, the generator G minimizes:

$$\mathcal{L}_G = \lambda_1 \mathcal{L}_{\text{adv}}^{\text{img}} + \lambda_2 \mathcal{L}_{\text{adv}}^{\text{obj}} + \lambda_3 \mathcal{L}_{\text{AC}}^{\text{obj}} + \lambda_4 \mathcal{L}_{\text{AC}}^{\text{att}} + \lambda_5 \mathcal{L}_{\text{KL}} + \lambda_6 \mathcal{L}_1^{\text{img}} + \lambda_7 \mathcal{L}_1^{\text{latent}} \tag{4}$$

and the discriminator D minimizes:

$$\mathcal{L}_D = -\lambda_1 \mathcal{L}_{\text{adv}}^{\text{img}} - \lambda_2 \mathcal{L}_{\text{adv}}^{\text{obj}} + \lambda_3 \mathcal{L}_{\text{AC}}^{\text{obj}} + \lambda_4 \mathcal{L}_{\text{AC}}^{\text{att}} \tag{5}$$

where λ_i are weights for different loss terms.

Implementation Details: We set image canvas size to 64×64 (128×128), and the object size to 32×32 (64×64). The $\lambda 1 \sim \lambda 7$ are 1.0, 1.0, 8.0, 2.0, 0.01, 5.0, 5.0. The dimension of the category embedding $\mathbf{w}$ and the latent code $\mathbf{z}$ are both 64. The model is trained using Adam with a learning rate of 0.0001 and a batch size of 6 for 300,000 iterations on 2 Geforce GTX 1080 Ti. In each training iteration, we first train the object classifier, the attribute classifier and the two discriminators, and then the generator.

2 Results

Due to limited space in the main paper, we provide additional evaluations here.

2.1 Spatial Equivariance Experiments

Figure 1 and 2 demonstrate the ability of our model to generate high quality images (at 128×128 resolution) and maintain consistency of objects when the boxes are shifted. We want to draw reader attention to 4-th row from the top in Figure 2. Note how our model can generate images where `tree` shifts from left to right based on the change in the layout (cyan), while largely maintaining the structure and appearance of the `boat` unchanged. In contrast, LostGAN [31], when presented with the same sifted layout, generates an image that is entirely incoherent with the original: `boat` is no longer discernible, sky changes color, *etc*. Similar behavior can also be observed in the last row, where our model is able to generate new version of the image with shifted placement of the `elephants`, while maintaining the `tree` line and overcast `sky`. The images produced with LostGAN [31] are highly incoherent with visible changes in both foreground and background objects, as well as their appearances (despite fixing appearance latent vectors). Similar behavior is also readily observed in Figure 1. For example, consider new shifted placement of the `person` in the third row from the top, or an almost mirror image produced by shifting `trees` and the `house` from right to left and vice versa in the 5-th row. LostGAN [31], while generates plausible images, is consistently failing to maintain style, appearance, structure and placement of objects when the layout is modified to simply spatially re-arrange the same objects.

Figure 3 shows similar ability to maintain consistency with spatial shifts of objects in the layout at the lower, 64×64, resolution. Note that results of LostGAN are less blurry because, unlike all other methods in Figure 3, they are computed at 128×128 resolution (but illustrated at 64×64); authors of LostGAN do not provide a trained 64×64 model. As

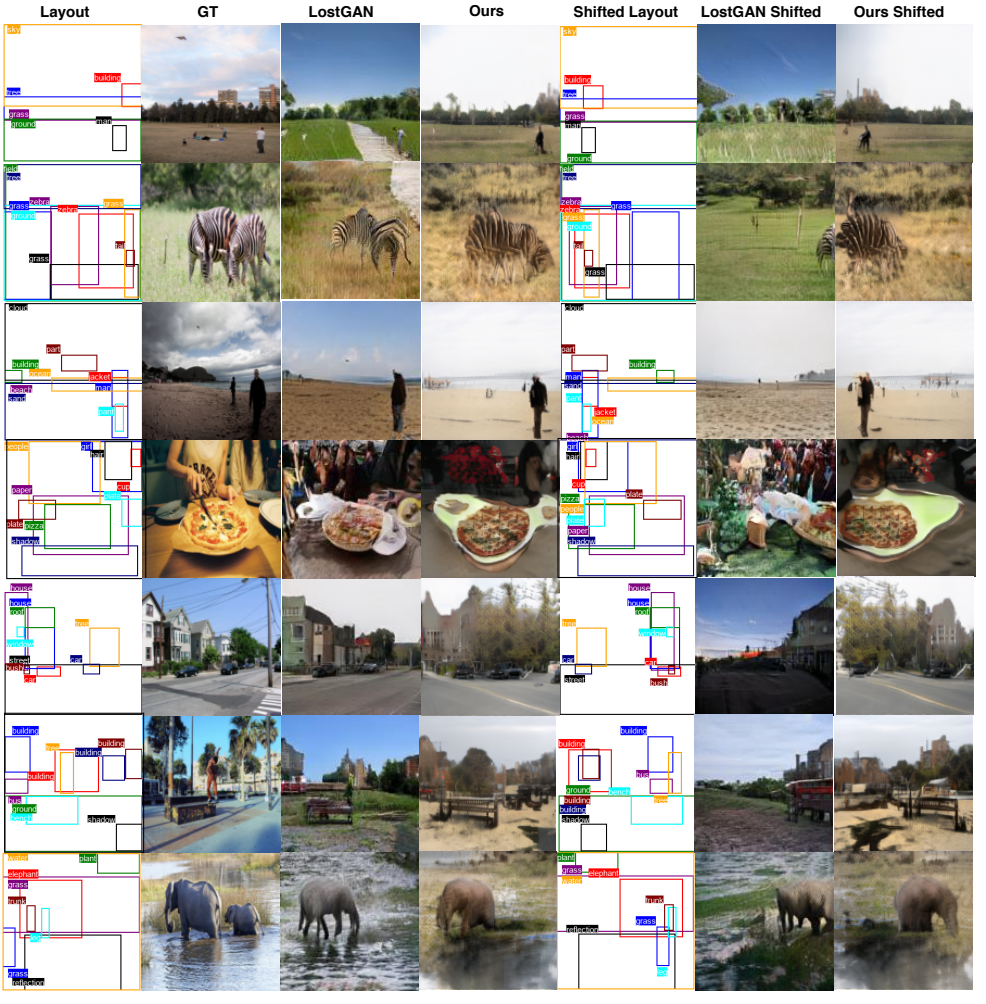

Figure 1: **Examples of 128 $\times$ 128 generated images with horizontally shifted bounding boxes** on Visual Genome datasets obtained by our proposed method.

such, the comparison to LostGAN isn't exactly fair and is less favorable to us. Despite this, our model, is able to generate high-quality images that are consistent under spatial shifts in layout (see last row).

Figure 2: **Examples of 128 $\times$ 128 generated images with horizontally shifted bounding boxes** on Visual Genome datasets obtained by our proposed method.

Figure 3: **Examples of 64 $\times$ 64 generated images with horizontally shifted bounding boxes** on Visual Genome datasets by our proposed method.

2.2 Qualitative Generation Experiments

Figure 4 showcases our model's ability to generate plausible images for a wide variety of layout configurations (*e.g.* human, food, animal, furniture, house). Notably, results of sg2im [10] and layout2im [40] are of lower quality and blurry. LostGAN [31] does not perform well on human faces. Similar to Figure 3, results of LostGAN in Figure 4 are less blurry because, unlike all other methods, they are at 128 $\times$ 128 resolution; LostGAN didn't provide trained 64 $\times$ 64 model, so we use a higher resolution model instead for visualization.

2.3 Attribute Modification Experiments

Figure 5 illustrates additional examples of our model's ability to modify attributes of various objects. The change of attributes does not affect the layout or other objects in the image.

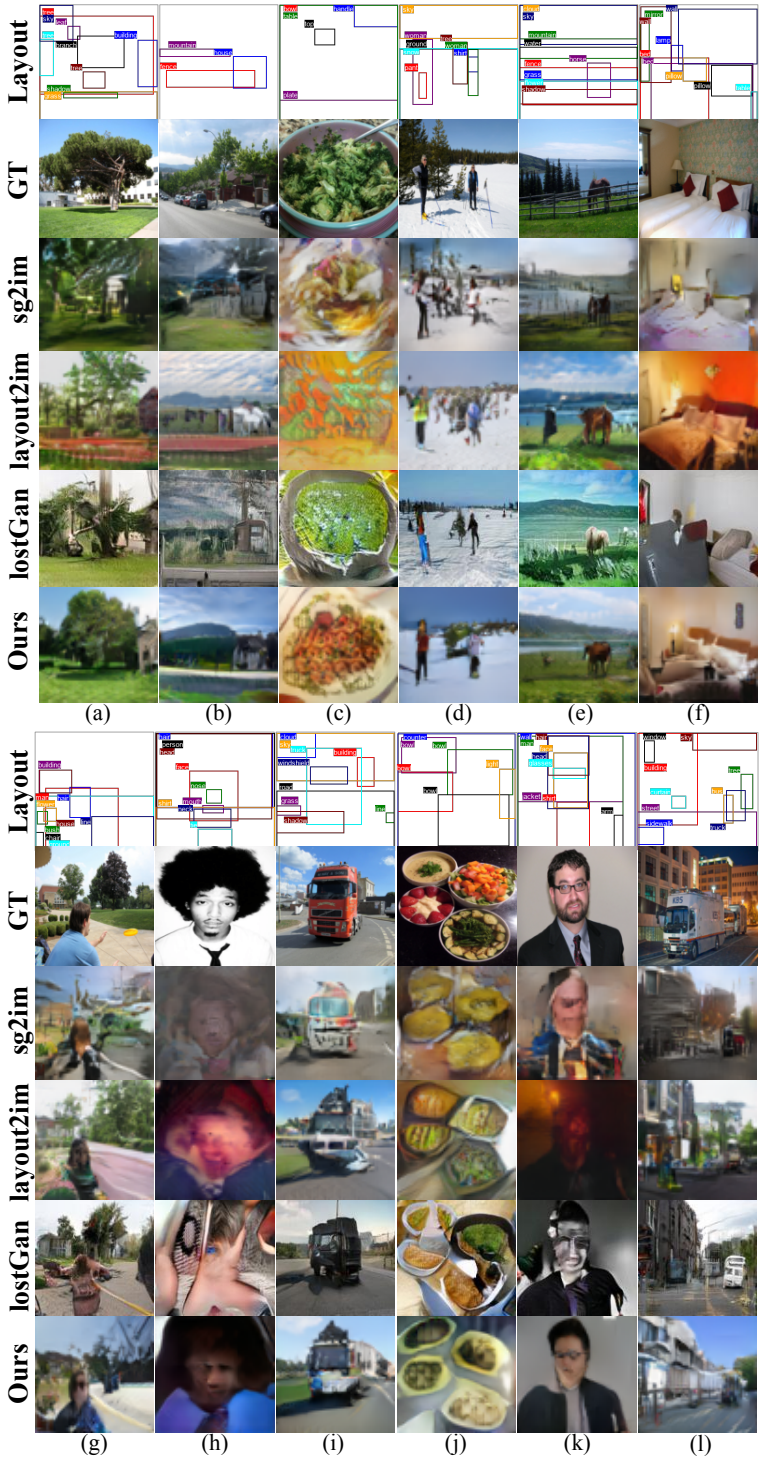

Figure 4: **Examples of 64 $\times$ 64 generated images** on Visual Genome datasets obtained by our proposed method.

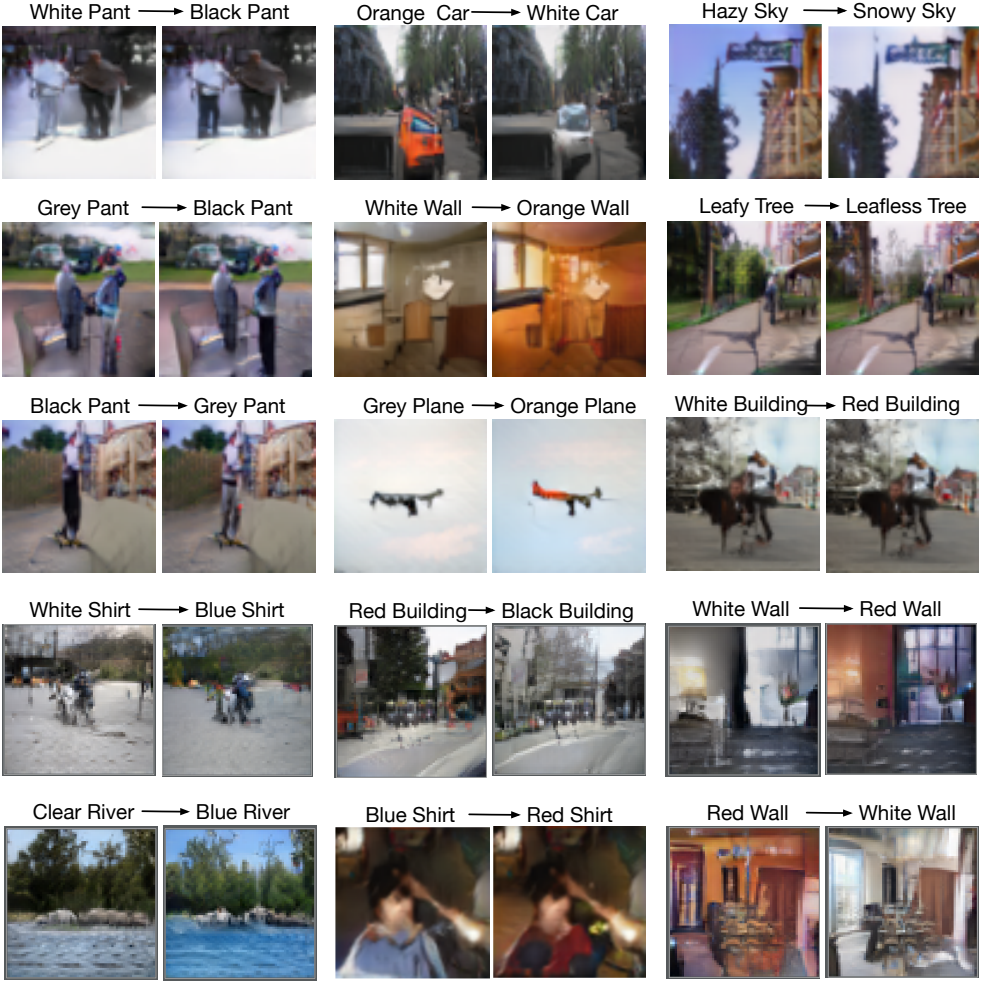

Figure 5: **Examples of 64 × 64 generated images with modified attributes** on Visual Genome datasets obtained by our proposed method.