Learning Object Context for Novel-view Scene Layout Generation

Xiaotian Qiao[1] Gerhard P. Hancke[2] Rynson W.H. Lau[2]

[1]School of Computer Science and Technology, Xidian University
[2]Department of Computer Science, City University of Hong Kong

Abstract

Novel-view prediction of a scene has many applications. Existing works mainly focus on generating novel-view images via pixel-wise prediction in the image space, often resulting in severe ghosting and blurry artifacts. In this paper, we make the first attempt to explore novel-view prediction in the layout space, and introduce the new problem of novel-view scene layout generation. Given a single scene layout and the camera transformation as inputs, our goal is to generate a novel view scene layout for a specified viewpoint. Such a problem is challenging as it involves accurate understanding of the 3D geometry and semantics of the scene from as little as a single 2D scene layout. To tackle this challenging problem, we propose a deep model to capture contextualized object representation by explicitly modeling the object context transformation in the scene. The contextualized object representation is essential in generating geometrically and semantically consistent scene layouts of different views. Experiments show that our model outperforms several strong baselines on many indoor and outdoor scenes, both qualitatively and quantitatively. We also show that our model enables a wide range of applications, including novel-view image synthesis, novel-view image editing, and amodal object estimation.

1. Introduction

Multi-view prediction of a scene is of great importance in 3D scene understanding and has been studied for a long time [6, 10], with potential applications such as robotics, Virtual Reality (VR), and Augmented Reality (AR). There is a line of research on generating images of a scene from new viewpoints [5, 31, 42]. However, these works render a scene in the image space via pixel-wise prediction directly, often resulting in severe ghosting and blurry artifacts.

In this paper, we take a step towards novel-view prediction of a scene in the layout space. We introduce the new problem of *novel-view scene layout generation*. Having a robust novel-view scene layout model not only enables generating consistent and sharp scene layouts from different views even with large camera movements, but also provides scene understanding priors for a wide range of applications, including novel-view image synthesis, image editing, and amodal object estimation. As shown in Figure 1, given a single layout of an indoor scene (a) or an outdoor scene (b) as input, our goal is to generate novel-view layouts of the scene for different viewpoints ($2nd$ and $3rd$ columns).

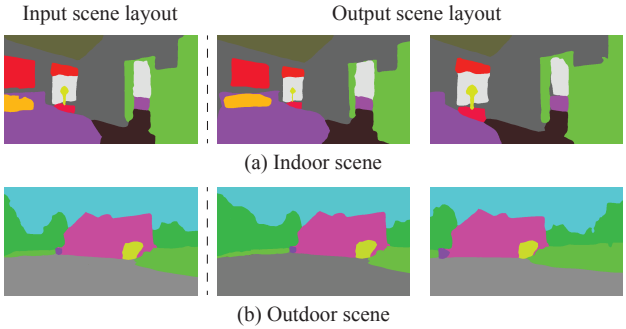

Figure 1. Novel-view scene layout generation. Given a single layout of an indoor scene (a) or an outdoor scene (b) as input ($1st$ column), our model can generate plausible novel-view scene layouts for different viewpoints ($2nd$ and $3rd$ columns).

However, generating plausible novel-view scene layouts is a highly challenging problem. It requires an accurate understanding of the 3D geometry and semantics of the scene from just a single 2D scene layout. The sizes, positions, and shapes of the objects may change a lot in both observed regions and unseen regions across different viewpoints. Hence, this problem is highly under-determined, as a result of the ambiguity of the input single 2D scene layout.

Recall how the human cognitive system works in this task. Consider the input layout of an indoor scene as shown in Figure 1 (a). Human beings may use the geometric priors and semantic relations of the objects in the input layout to infer the scene layout of a different viewpoint. The geometric priors contain the common object properties (*e.g.*, the shape of the window), semantic relations represent interactions among different types of objects to indicate how these objects (*e.g.*, bed and table) should be composited in the target view. Inspired by this observation, we propose a learning-based model for novel-view scene layout generation, by explicitly modeling the object spatial and seman-

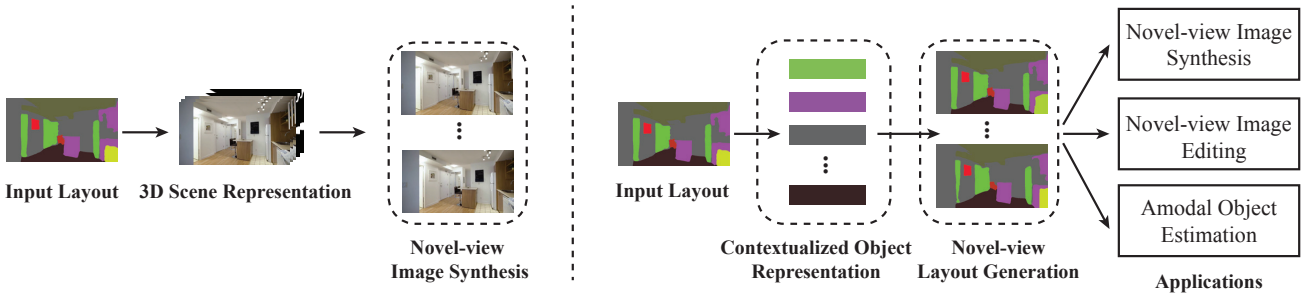

Figure 2. A brief comparison between existing novel-view layout-to-image synthesis methods [12, 14] (left) and the proposed novel-view scene layout generation method (right). Existing works directly map the input layout to the 3D scene representation (*e.g.*, MPI [14] or hybrid representation [12]), and then perform pixel-by-pixel projection in the image space. In contrast, our approach considers spatial and semantic interactions among objects in the layout space without explicit 3D modeling. The novel-view scene layouts generated from the contextualized object representation are crucial for scene understanding and enable a variety of applications.

tic interactions in the scene. Our approach contains three main stages. First, given an input scene layout, we propose an Object Context Transformation (OCT) module to extract the contextualized object representation that encodes object shapes, positions, and sizes in the scene. The contextual relationships among objects in the target view are learned via a view-aware attention mechanism. Second, we propose an Object Layout Generation (OLG) module to produce the shape, size, and position for each object in the target view. Finally, we use an Object Layout Composition (OLC) module to composite all the predicted object layouts and generate a plausible novel-view scene layout as output.

To evaluate the effectiveness of our model, we conduct extensive experiments on numerous indoor and outdoor scenes. Results show that our model can generate geometrically and semantically more consistent novel-view scene layouts, compared with the baselines. In addition, we show a wide range of applications of our model, including novel-view image synthesis, novel-view image editing, and amodal object estimation.

In summary, the main contributions of this paper include:

- We make the first attempt to investigate the new problem of novel-view scene layout generation by learning object context in the scene.
- We propose a new model that consists of an OCT module to capture the contextualized object representation, an OLG module to predict the layouts of individual objects, and an OLC module to composite the predicted object layouts properly to an output scene layout.
- Experimental results demonstrate that our model can generate geometrically and semantically consistent novel-view scene layouts from a single input layout, enabling a wide range of applications.

2. Related Work

Novel view synthesis. Given a single or some images of a scene, novel view synthesis aims to generate images from novel viewpoints. Earlier solutions are based on multi-view reconstruction using geometric formulations [4, 6, 8, 27, 43]. Their results often suffer from occlusions and incorrect texture details due to view blending. Recent works leverage different 3D representations, including a multilayer perceptron [30], multi-plane images [41], layered depth images [28, 31], point clouds [23, 34] and neural radiance fields (NeRFs) [22]. Zhi *et al*. [39] treat NeRF as a scene-specific implicit representation for joint geometric and semantic prediction. Novel view images can be generated by warping the learned scene representation to the target view.

However, all these view synthesis methods work in the image space, which contains pixel-level appearances. In contrast, we tackle the novel-view prediction problem in the layout space. The input to our model is a single scene layout that captures the scene structure and is useful in many applications. Although few recent works [12, 14] also use semantic scene layouts as input, they just convert the input layout to different types of scene representations and follow the traditional novel-view synthesis pipeline that performs pixel-by-pixel projection in the image space. As shown in Figure 2 (left), the 3D scene representation consists of a set of fronto-parallel planes at fixed depths from a reference image. Such a design would still lead to blurry predictions when the camera movement is large. In contrast, our model (c) directly generates geometrically and semantically consistent scene layouts for different viewpoints by utilizing object context transformation without explicit 3D modeling. The generated scene layouts can further be used in a variety of applications, including novel-view image synthesis.

Scene layout generation. In recent years, we have witnessed a rising interest in scene layout generation in the vision community. LayoutGAN [19] uses a GAN model to generate semantic and geometric properties of a fixed number of elements. LayoutVAE [17] proposes a conditional VAE model to generate a feasible layout of the scene given a

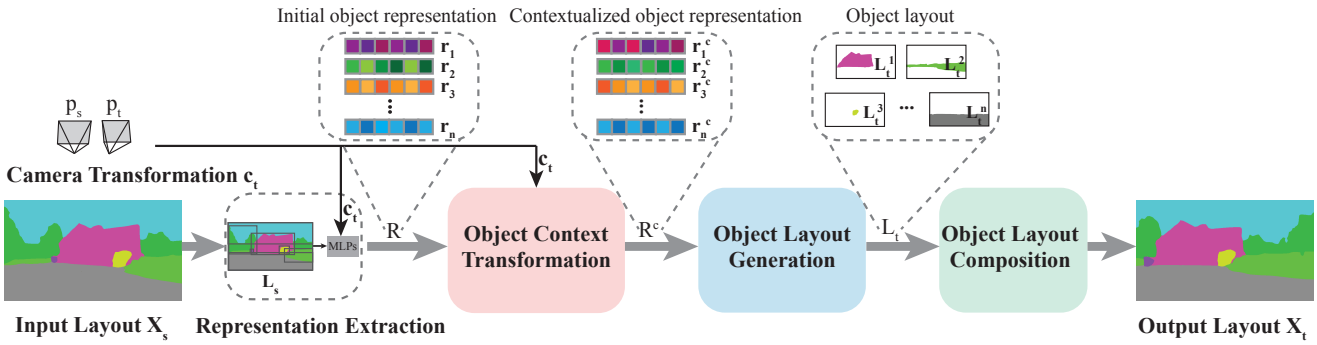

Figure 3. The overall pipeline of our model for novel-view scene layout generation. Given a single scene layout X_s in the source view and the camera transformation c_t as inputs, we first extract the initial object representation R by encoding a set of decomposed object layouts L_s and c_t. The initial object representation R is updated by the Object Context Transformation (OCT) module, resulting in the contextualized object representation R^c that captures the spatial and semantic interactions of objects in the scene. The Object Layout Generation (OLG) module then predicts the layout L_t^i of each object i at the target view from r_i^c. Finally, the Object Layout Composition (OLC) module composites the predicted object layouts L_t to generate the output scene layout X_t in the target view.

label set, *i.e.*, categories of all the elements. Qiao *et al*. [25] propose a generative model to predict complete scene layouts from a standalone object layout. Lee *et al*. [18] use a graph neural network to generate layouts from a set of input constraints. Luo *et al*. [20] introduce a conditional variational autoencoder to generate diverse and realistic layouts of indoor scenes. Recent works [1, 11, 35] also use transformer-based networks for layout generation and completion by capturing the high-level relationships among elements in a layout. Unlike these scene layout generation and completion works, our goal is to generate scene layouts for different viewpoints, which has not been explored.

Attention mechanism. The original attention mechanism [32] is applied to sequence-to-sequence machine translation and used in many NLP tasks. The core idea is to model long-range dependencies among the input elements. Most recently, the attention mechanism has begun to show promising results in computer vision tasks, such as image recognition [7], object detection [3], image segmentation [38], and image generation [33]. Different from the above works, we propose an object context transformation module and apply the attention mechanism on a new problem, *i.e.*, novel-view scene layout generation.

3. Approach

In this section, we introduce our approach for the novel-view scene layout generation problem. We first give an overview of this problem and the proposed pipeline, and then describe the details of the modules in our model. Finally, we specify the loss terms used in the training process.

3.1. Overview and Notation

A scene layout can be considered as composed of a set of objects with different sizes, categories, and positions in a scene. Formally, the goal of the novel-view scene layout generation problem is to develop a model $\mathcal{G}$, which can generate a scene layout X_t in the target view, by taking a single scene layout map X_s in the source view and a relative camera pose transformation c_t (from the source view to the target view) as inputs:

$$X_t = \mathcal{G}(X_s, c_t). \tag{1}$$

To model object-level context information, we decompose X_s into a set of object layouts, $L_s = \{L_s^i \in \{0,1\}^{H\times W\times C} | i = 1, ..., n\}$, where H and W are the height and width of the layout, C is the total number of object categories, and n is the number of objects in the scene. Note that multiple occurrences of a single object class will appear as a group of connected or disjoint masks in the same channel in the object layout. We fill each pixel inside the object as a one-hot vector to represent the object category and zeroing out the values outside it.

Figure 3 shows the overall pipeline of our model. It contains three key modules: an Object Context Transformation (OCT) module, an Object Layout Generation (OLG) module, and an Object Layout Composition (OLC) module. In particular, we first use two Multi-Layer Perceptrons (MLPs) to separately extract object embeddings $E = \{e_i | i = 1, ..., n\}$ and camera transformation vector c_t in a d-dimensional space. E and c_t are concatenated to another MLP to form the initial object representation $R = \{r_i | i = 1, ..., n\}$. We pass R to the OCT module to obtain the contextualized object representation $R^c = \{r_i^c | i = 1, ..., n\}$ by modeling the spatial and semantic interactions among objects. We then feed R^c to the OLG module to predict the object layouts $L_t = \{L_t^i | i = 1, ..., n\}$ in the target view. Finally, we combine all the predicted object layouts L_t by the OLC module to output the scene layout X_t. We present the details of the three key modules below.

3.2. Object Context Transformation (OCT) Module

Given the initial object representation, a trivial solution is to pass it to a decoder directly to predict the layout in the target view. However, our experiments show that it performs poorly in this task. The main reason is that the object representation is not spatially aligned between the source and target views. To leverage the inductive bias about the semantic and geometric information of the scene across different views, the object representation should take all input objects and the camera transformation into consideration.

Hence, we propose an OCT module to learn complementary information across different views by integrating the object context information into the feature representation of each object. In particular, we design a View-aware Attention Block (VAB) to exploit the local and global view-aware dependencies among objects. Figure 4 shows the details of the VAB architecture. The view-aware attention is based on the initial object representation and the camera transformation information. VAB uses three different fully connected layers to produce the queries Q, keys K and values V. We perform matrix multiplication to the query and key to obtain the attention matrix A. We add a residual connection after the attention computation. The contextualized representation r_i^c of each object i is computed as:

$$\begin{aligned} r_i^c &= (r_i + \sum_{j=1}^{n} w_{i,j} r_j W_V) W_P, \\ w_{i,j} &= \frac{exp(A_{i,j})}{\sum_{k=1}^{n} exp(A_{i,k})}, \\ A_{i,j} &= ([r_i; c_t] W_Q)(r_j W_K)^T, \end{aligned} \tag{2}$$

where $[;]$ denotes the concatenation operation. W_Q, W_K, W_V and W_P are linear transformation layers. $w_{i,j}$ is the computed weight between objects. n is the total number of objects in a scene. We also use the multi-head attention as:

$$R^c = [head_1; ...; head_m] W_O, \tag{3}$$

where m is the number of heads, and W_O is a linear transformation layer. The output from each head is combined to encapsulate multiple relationships among the objects.

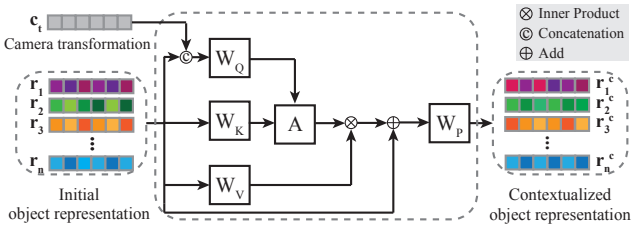

Figure 4. The View-aware Attention Block (VAB) for learning object context information.

By doing so, the contextualized object representation contains not only its own geometric and semantic information, but also the spatial and semantic interactions with other objects in the target view. The contextualized object representation is then fed into the OLG module.

3.3. Object Layout Generation (OLG) Module

Given the set of contextualized object representation $R^c = \{r_i^c | i = 1, ..., n\}$, we propose an OLG module to predict a standalone layout L_t^i for each object i. The OLG module contains an object bounding box branch and an object mask branch. The object bounding box branch predicts the position and size of each object with four parameters $\{o_i^x, o_i^y, o_i^h, o_i^w\}$, while the object mask branch predicts a binary shape mask of the object. (o_i^x, o_i^y) refer to the centroid coordinates of the object bounding box, and (o_i^h, o_i^w) refer to the height and width of the object bounding box. They are all normalized with respect to the size of the scene layout. We warp the object mask to the corresponding bounding box coordinates using a bilinear sampler [16], resulting in an object layout map L_t^i that represents the object shape, size, and position in the target view.

3.4. Object Layout Composition (OLC) Module

Finally, we propose an OLC module to combine the predicted object layouts L_t coherently into a scene layout X_t in the target view. Note that we need to deal with occlusions (*i.e.*, multiple objects appearing at the same location in X_t) and holes (*i.e.*, no objects appearing at some location in X_t) during the composition process.

To resolve the ambiguity about the partial occlusions among objects, we propose an OrderNet to determine the relative order of any two objects. The inputs to the OrderNet are two object layouts, and the output is a binary label indicating the relative order of the two object layouts. The ground truth orders are derived from the depth map in the dataset. The architecture of the OrderNet is based on VGG [29]. We train the OrderNet by using the relative order information of adjacent objects in the scene with a cross-entropy loss. Based on the outputs from the pretrained OrderNet, we composite the object layouts properly into a single layout in the target view.

The composited layout may still look unrealistic as there may exist missing regions in the target view layout. To further reduce the artifacts, we employ four refinement blocks to naturally refine the scene layout in a semantically meaningful manner. Each refinement block takes as input the coarse scene layout to produce a refined layout by bilinear upsampling and convolution operations. More details about the network architecture can be found in the Supplemental.

3.5. Training

We design several loss terms for learning the novel-view scene layout generation model from both object-level and scene-level perspectives. We adopt the bounding box loss and the mask loss for each object in the scene, and the adversarial loss for ensuring the plausibility of the generated scene layout. In particular, for each object bounding box, we define the L1 loss between the predicted object bounding box $\hat{o}_i^{bbox}$ and the corresponding ground truth o_i^{bbox} as:

$$L_{bbox}^i = \|o_i^{bbox} - \hat{o}_i^{bbox}\|_1. \quad (4)$$

For each object mask, we use a binary cross-entropy loss to penalize the pixel-wise difference between the predicted object mask $\hat{m}_i$, and the ground truth mask m^i as:

$$L_{shape}^i = -\sum_x \sum_y \hat{m}_{x,y}^i \log m_{x,y}^i + (1-\hat{m}_{x,y}^i) \log(1-m_{x,y}^i), \quad (5)$$

where $\hat{m}_{x,y}^i$ is the predicted object mask at location (x, y).

In addition, to encourage the plausibility of the generated scene layout, we train our model against a discriminator via adversarial learning as:

$$L_{adv} = \mathbf{E}_{x\sim p_{real}} \log D(x) + \mathbf{E}_{x\sim p_{fake}} \log(1-D(x)), \quad (6)$$

where $x \sim p_{fake}$ is the generated scene layout, and $x \sim p_{real}$ is the real scene layout.

In summary, we train our model with a total loss of:

$$L = \lambda_0 \sum_i L_{shape}^i + \lambda_1 \sum_i L_{bbox}^i + \lambda_2 L_{adv}, \quad (7)$$

where λ_0, λ_1, and λ_2 are the controllable loss weights.

4. Experiments

In this section, we first introduce the experimental settings in Section 4.1. Second, we conduct experiments to evaluate the performance of the proposed method with several baselines both quantitatively and qualitatively in Section 4.2. We further conduct ablation studies to analyze the proposed modules in Section 4.3. Finally, we show that our model enables a variety of applications in Section 4.4.

4.1. Experimental Setup

Implementation details. We implement our network with PyTorch. The scene layouts are resized to a resolution of 128×128 in both training and testing. The activation function is leaky-ReLU and its negative slope is 0.2. The network parameters are randomly initialized. We adopt the Adam optimizer with β_1 = 0.9, β_2=0.99 and a learning rate of 0.0001. We set the loss weights λ_0, λ_1, and λ_2 to 1, 1, 0.1. The number of attention heads is set to be 4. We first train the OrderNet to obtain the relative order of adjacent object layouts, and then train the whole model end-to-end.

Dataset. We collect training pairs of each frame from the RealEstate10K dataset [41], which is licensed by Google LLC under a CC-BY 4.0 License. It consists of 80,000 indoor and outdoor video clips with camera poses for all frames. Specifically, to extract pairs of scene layouts, we apply PSPNet [37] to obtain semantic segmentation annotations, and remove noises and fill holes in the obtained object masks via connected component labeling. We then derive the bounding boxes of objects from the semantic segmentation results, and apply a simple multi-object tracker [2] to find correspondences between objects across different views. We also apply the pre-trained MiDaS [26] to obtain depth information. Note that no training or additional data is used in these steps. We set the coordinates of an object to zeros if it does not appear in one of the views.

Compared methods. As this is the first work for novel-view scene layout generation, there is no existing method that we can compare with directly. We therefore propose several strong baselines that address related problems, including UNet [15], LayoutGAN [19] and GVSNet [12]. Of these three methods, UNet [15] learns a generic mapping between an input layout and an output layout using a fully convolutional encoder-decoder architecture. We concatenate the input scene layout with camera transformation information as the input and retrain UNet. LayoutGAN [19] learns the layout mapping in an object-wise manner. We modify the original LayoutGAN model so that it takes our initial object representation as input. Similar to the OLG module in Section 3.3, we add an object mask branch in LayoutGAN to output an object layout for each object. We composite the generated object layouts into a scene layout and retrain the model. GVSNet [12] uses MPI semantics to synthesize novel-view images from a single input layout. We adapt their method for novel-view scene layout generation by projecting the intermediate MPI semantics in the network to the scene layout in the target view directly.

Evaluation metrics. We evaluate the quality of the generated scene layout from different aspects. We first compute the Fréchet Inception Distance (FID) score [13] to measure the visual quality between the generated layouts and the real layouts by using the layout features from the last convolution layer of the layout discriminator. We also compute the average Negative Log-Likelihood (NLL) score [17, 25] to measure the overall plausibility of the generated scene layouts. In addition, the object semantics should be consistent when a scene is rendered from different viewpoints. To measure such consistency, we define a new metric called View Semantic Consistency (VSC) as:

$$\|\mathcal{W}_s(X_t^1) - \mathcal{W}_s(X_t^2)\|, \quad (8)$$

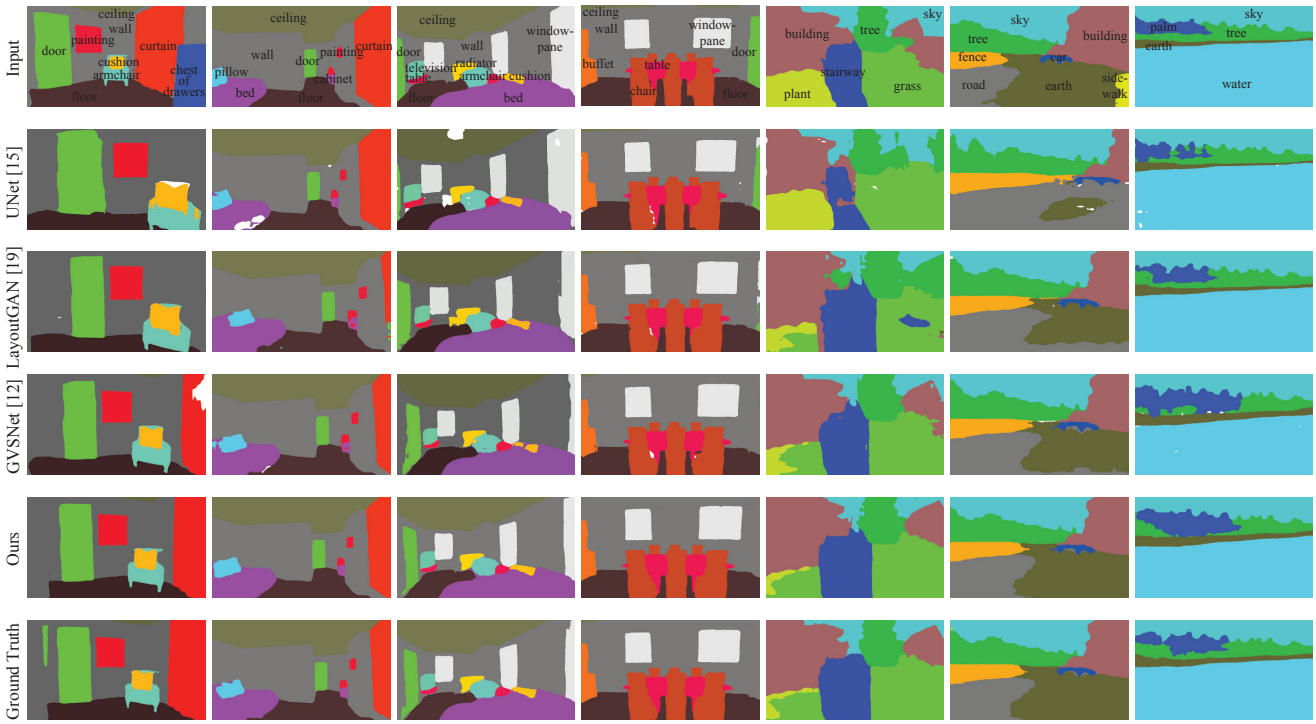

Figure 5. Qualitative comparison between the baselines and our model. Given the input scene layout ($1st$ row), we show the novel-view scene layouts from the baselines ($2nd$, $3rd$, and $4th$ rows), our model ($5th$ row), and the ground truth ($6th$ row).

where X_t^1 and X_t^2 are the generated scene layouts from two different user-specified viewpoints. $\mathcal{W}_s$ is a warping function that warps the generated scene layout back to the source viewpoint by using the depth information derived from MiDaS [26]. In particular, we transform the relative disparity images into absolute disparity images by estimating the scale and shift for each image. We compute the average per-channel absolute error between the warped two scene layouts. A low VSC score indicates high view consistency of the measured scene layouts across different views.

4.2. Results

Qualitative evaluation. Figure 5 shows some qualitative results of our model, compared with those from the baselines. From the results, we can see that the layouts generated from our method are more geometrically consistent. For example, in the $2nd$ column, the layouts from both UNet and LayoutGAN do not follow the camera movement, compared with the ground truth scene layout. Although the layout from GVSNet is better due to the learned 3D scene representation, it still generates unrealistic artifacts when projecting the 3D representation to the 2D layout. In contrast, our layout follows the camera transformation to arrange all objects in the target view properly. In addition, we can see that our approach can generate more visually plausible scene layouts in the target view. For example, the input indoor scene in the $3rd$ column has multiple and complex object interactions, *e.g.*, a television on a table, a cushion on a bed, and three windowpanes on the wall. All the generated scene layouts from the baselines contain obvious artifacts (marked in white color) around object boundaries with no object label or incorrect label assigned. In contrast, benefited from the learned object context, our approach can generate much more plausible scene layouts.

Method	NLL↓	FID↓	VSC↓
UNet [15]	2.31	126	0.132
LayoutGAN [19]	2.05	112	0.098
LayoutGAN [19]+OLC	1.96	108	0.091
GVSNet [12]	1.69	103	0.079
Ours	**1.53**	**85**	**0.059**

Table 1. Quantitative comparison of the proposed method with the baselines (*i.e.*, UNet, LayoutGAN, LayoutGAN+OLC, and GVSNet). We evaluate their performances using NLL, FID, and VSC scores. The best results are highlighted in bold.

Quantitative evaluation. Table 1 shows the quantitative results. Our method achieves the best results on all the metrics as compared to the baselines, indicating the effectiveness of our method. We can see that utilizing our proposed OLC module could help improve the performance of LayoutGAN. Compared to UNet and LayoutGAN, GVSNet is closest to our method on the NLL metric. The main reason is that GVSNet converts the input scene layout to a 3D scene

Method	NLL↓	FID↓	VSC↓
w/o camera transformation	1.65	101	0.089
w/o OCT module	1.71	109	0.093
w/o OrderNet	1.58	90	0.082
w/o refinement	1.57	93	0.065
w/o adversarial loss	1.61	91	0.063
Ours (full model)	**1.53**	**85**	**0.059**

Table 2. Results of the ablation study. The best results are highlighted in bold.

representation to explore object relations. However, by considering the spatial and semantic interactions among objects in the scene, our method still outperforms GVSNet on all the metrics by a large margin. This again demonstrates the importance of learning object context transformation in the novel-view scene layout generation problem.

4.3. Ablation Study

To investigate how different modules and loss functions affect the final results, we conduct ablation studies on several ablated versions of our model:

- *w/o camera transformation*: We remove the camera transformation vector in the OCT module.
- *w/o OCT module*: We remove the OCT module to evaluate the importance of object context information.
- *w/o OrderNet*: We remove the OrderNet to evaluate the occlusion effect on the generated scene layout.
- *w/o refinement*: We remove the refinement blocks to evaluate the effect of the refinement process.
- *w/o adversarial loss*: We train the model without using the adversarial loss.

Table 2 shows the results of the ablation study. Without utilizing camera transformation information, the performance drops. This indicates that camera transformation is useful for distilling view-aware object context information in the OCT module. If the OCT module is completely removed, the performance becomes worse, implying that modeling object context transformation is crucial to novel-view scene layout generation. Without OrderNet, the VSC score is affected more than the NLL and FID scores. This is because that incorrect depth order would lead to inconsistent composition in the generated scene layouts. Finally, with the help of the refinement blocks and the adversarial loss, our model learns to generate more plausible and consistent novel-view scene layouts.

4.4. Applications

Benefited from our model for novel-view scene layout generation, we explore three applications here.

Novel-view image synthesis. The goal of this application is to generate novel-view images of a scene by using only a single 2D scene layout as input. Such an application allows users to easily draw and edit a scene layout on a digital canvas and generate multi-view images of the scene with geometrically consistent and visually plausible appearance. Specifically, given an input scene layout, we first use our model to generate a new scene layout in the target view. We then use an off-the-shelf semantic image synthesis method [24] to synthesize a photo-realistic image from the generated layout. Note that we may enforce consistency on the image contents of different views by conditioning the generated layout on the same latent style code.

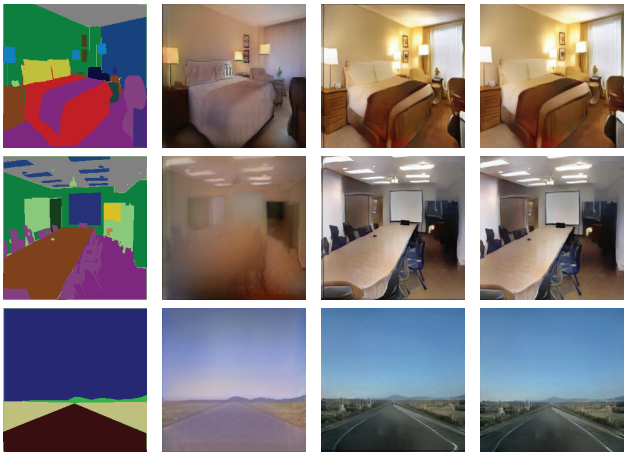

Figure 6. Novel-view image synthesis. We compare the synthesized novel-view images of two baselines (*i.e.*, SPADE [24] and SVSNet [14]) with ours.

We compare our results with two baselines on the ADE20k dataset [40] licensed under a BSD 3-Clause License. One is using SPADE [24] to generate novel-view images from the input layout directly, and the other is a recent work, SVSNet [14], which generates novel-view images by inferring a full 3D scene representation from the input scene layout. Figure 6 shows some visual comparison results on both indoor and outdoor scenes. We can see that SPADE suffers from blurry artifacts due to the unaligned mapping between the input layout and the target view image. Although the visual quality of SVSNet is better, it still generates inconsistent artifacts around the scene boundaries. In contrast, our method can generate sharper visual results based on the generated scene layout in the target view.

Novel-view image editing. Our approach can also support novel-view image editing. Given a number of synthesized novel-view scene layouts, users may select any scene layout and edit the content. The results of the editing operation will then be propagated to other viewpoints of scene layouts and corresponding images accordingly. In particular, we first edit the scene layout of the input image. We then use our model to generate novel-view scene layouts.

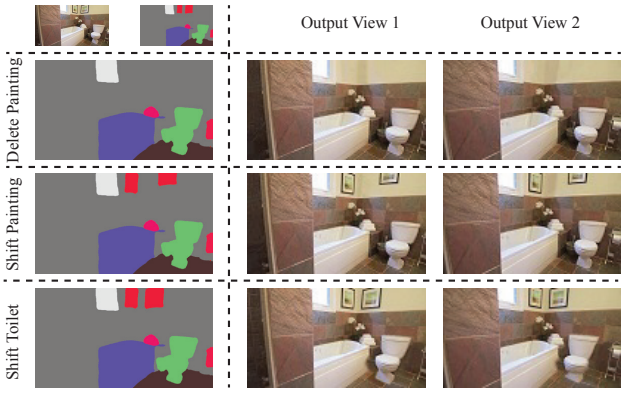

Figure 7. Novel-view image editing. Given the input image and layout (top-left), we show the generated results of two different views ($2nd$ column and $3rd$ column) based on different editing operations ($2nd$ row for deleting two paintings, $3rd$ row for shifting a painting, and $4th$ row for shifting the toilet).

Finally, we pass the generated scene layouts in different views and the original image in the source view to a cross-domain semantic transfer model [36] to produce consistent novel-view images, combining the visual appearance of the input image and the structure of the generated layouts.

We show an example in Figure 7. Given the input image and layout in the top-left corner of Figure 7, we apply different editing operations (*e.g.*, deleting two paintings in the $2nd$ row, shifting a painting in the $3rd$ row, and shifting the toilet in the $4th$ row) on the layout. The results of the generated images in two different views are shown in the $2nd$ and $3rd$ columns. We can see that a simple editing operation on the layout can be propagated seamlessly and consistently to the novel-view images.

Amodal object estimation. Given only a single image of a scene as input, the goal of this application is to reason about the amodal object layout in the bird's eye view. The resulting object layouts can be useful for perception and scene understanding in autonomous driving scenarios. Hence, we explore the use of our model for amodal object estimation to reconstruct object layouts in the bird's eye view. Given a single frontal view image, we first use a pre-trained semantic segmentation model [37] to obtain the corresponding scene layout. We pass the scene layout to our OCT module to extract the contextualized object representation. We then predict the specific object layout in the bird's eye view by using our OLG module.

We compare our method with a recent work, *i.e.*, MonoLayout [21], which leverages adversarial learning to estimate the bird's eye view scene layout from a single image. We follow the dataset processing step in [21] and retrain our model on the KITTI dataset [9] for the sidewalk layout recovery in the bird's eye view. We adopt mean Intersection-over-Union (mIOU) as the evaluation metric.

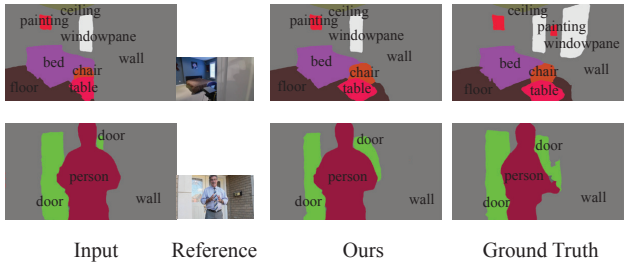

Figure 8. Failure cases. Our model may fail to recover objects that are completely occluded in the input layout ($1st$ row). In addition, our model may not be able to predict the changes of deformable objects in the target view ($2nd$ row).

A higher score represents a better performance. The mIOU of our method and MonoLayout for sidewalk is 44.31 and 42.66, respectively. This indicates that our model can hallucinate object shapes better. We attribute this to the strong contextualized object representation learned in our model.

5. Conclusion

In this paper, we take a step towards the new problem of novel-view scene layout generation. To this end, we propose a learning-based model that captures the contextualized object representation to generate geometrically and semantically consistent scene layouts across different views. Extensive qualitative and quantitative results show that our model outperforms several baselines on numerous indoor and outdoor scenes. We believe that our approach can serve as a critical step for a wide range of potential applications.

Though impressive results are achieved by our model, as the first trail to generate novel-view scene layouts, our approach is subject to some limitations. First, we cannot recover an object in the target view if it is not shown in the source view. As shown in the $1st$ row of Figure 8, two objects (*i.e.*, painting and windowpane) appeared in the ground truth are totally occluded by the wall on the right of the input layout. As a future work, we may learn a probabilistic generative model to capture the ambiguity inherent in the unseen regions across different views. New objects can be sampled at high fidelity from the learned distribution. Second, our model may not be able to predict the changes of deformable objects corresponding to viewpoint changes. See the $2nd$ row of Figure 8 for an example. It would be an interesting future work to explicitly encode object dynamics in the scene. Third, our method may not work well under some out-of-distribution camera poses due to the implicitly modeling. We believe that better modeling of object relationships under different camera poses could be an important future direction.

Acknowledgements: This work is in part supported by a GRF from RGC of Hong Kong (RGC Ref.: 11205620).

References

[1] Diego Martin Arroyo, Janis Postels, and Federico Tombari. Variational transformer networks for layout generation. In *CVPR*, 2021. 3

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, 2016. 5

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[4] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM TOG*, 2003. 2

[5] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *ICCV*, 2019. 1

[6] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 1, 2

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[8] Andrew Fitzgibbon, Yonatan Wexler, and Andrew Zisserman. Image-based rendering using image-based priors. *IJCV*, 63(2):141–151, 2005. 2

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 8

[10] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *SIGGRAPH*, 1996. 1

[11] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *ICCV*, 2021. 3

[12] Tewodros Habtegebrial, Varun Jampani, Orazio Gallo, and Didier Stricker. Generative view synthesis: From single-view semantics to novel-view images. In *NeurIPS*, 2020. 2, 5, 6

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 5

[14] Hsin-Ping Huang, Hung-Yu Tseng, Hsin-Ying Lee, and Jia-Bin Huang. Semantic view synthesis. In *ECCV*, 2020. 2, 7

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 5, 6

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *NeurIPS*, 2015. 4

[17] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In *CVPR*, pages 9895–9904, 2019. 2, 5

[18] Hsin-Ying Lee, Lu Jiang, Irfan Essa, Phuong B Le, Haifeng Gong, Ming-Hsuan Yang, and Weilong Yang. Neural design network: Graphic layout generation with constraints. In *ECCV*, 2020. 3

[19] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. Layoutgan: Synthesizing graphic layouts with vector-wireframe adversarial networks. *IEEE TPAMI*, 2020. 2, 5, 6

[20] Andrew Luo, Zhoutong Zhang, Jiajun Wu, and Joshua B Tenenbaum. End-to-end optimization of scene layout. In *CVPR*, 2020. 3

[21] Kaustubh Mani, Swapnil Daga, Shubhika Garg, Sai Shankar Narasimhan, Madhava Krishna, and Krishna Murthy Jatavallabhula. Monolayout: Amodal scene layout from a single image. In *WACV*, 2020. 8

[22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[23] David Novotnỳ, Benjamin Graham, and Jeremy Reizenstein. Perspectivenet: A scene-consistent image generator for new view synthesis in real indoor environments. In *NeurIPS*, 2019. 2

[24] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 7

[25] Xiaotian Qiao, Quanlong Zheng, Ying Cao, and Rynson WH Lau. Tell me where i am: Object-level scene context prediction. In *CVPR*, 2019. 3, 5

[26] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE TPAMI*, 2020. 5, 6

[27] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006. 2

[28] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *CVPR*, 2020. 2

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 4

[30] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *ICCV*, 2021. 2

[31] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *ECCV*, 2018. 1, 2

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3

[33] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. *arXiv:2012.09793*, 2020. 3

[34] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *CVPR*, 2020. 2

[35] Cheng-Fu Yang, Wan-Cyuan Fan, Fu-En Yang, and Yu-Chiang Frank Wang. Layouttransformer: Scene layout generation with conceptual and spatial diversity. In *CVPR*, 2021. 3

[36] Pan Zhang, Bo Zhang, Dong Chen, Lu Yuan, and Fang Wen. Cross-domain correspondence learning for exemplar-based image translation. In *CVPR*, 2020. 8

[37] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 5, 8

[38] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021. 3

[39] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021. 2

[40] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 7

[41] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM TOG*, 2018. 2, 5

[42] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *ECCV*, 2016. 1

[43] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM TOG*, 2004. 2