

# Synthesizing Indoor Scene Layouts in Complicated Architecture Using Dynamic Convolution Networks

HAO JIANG, SIQI WANG, and HUIKUN BI\*,

Institute of Computing Technology, Chinese Academy of Sciences,

University of Chinese Academy of Sciences,

Beijing Key Laboratory of Mobile Computing and Pervasive Device, China

XIAOLEI LV, BINQIANG ZHAO, and ZHENG WANG, Alibaba Group, China

ZHAOQI WANG, Institute of Computing Technology, Chinese Academy of Sciences, China

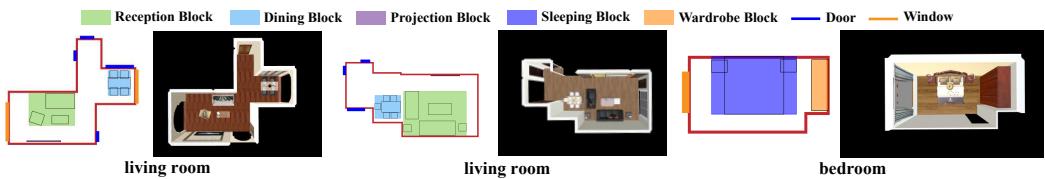


Fig. 1. We present a framework to synthesize indoor scene layouts using dynamic convolution networks, enabling us to easily generate plausible 3D indoor scenes in irregular-shaped architecture rooms. The indoor objects are grouped and then arranged by functional blocks. Here, we present two indoor scenes from the top-down view of the living rooms and one of the bedrooms. In each group, the left one shows 2D functional blocks, and the right one is the corresponding 3D scene.

Synthesizing indoor scene layouts is challenging and critical, especially for digital design and gaming entertainment. Although there has been significant research on the indoor layout synthesis of rectangular-shaped or L-shaped architecture, there is little known about synthesizing plausible layouts for more complicated indoor architecture with both geometric and semantic information of indoor architecture being fully considered. In this paper, we propose an effective and novel framework to synthesize plausible indoor layouts in various and complicated architecture. The given indoor architecture is first encoded to our proposed representation, called InAiR, based on its geometric and semantic information. The indoor objects are grouped and then arranged by functional blocks, represented by oriented bounding boxes, using dynamic convolution networks based on their functionality and human activities. Through comparisons with other approaches as well as comparative user

\*Corresponding author.

Authors' addresses: Hao Jiang, jianghao@ict.ac.cn; Siqi Wang, 18829026208@163.com; Huikun Bi, bihuikun@ict.ac.cn, Institute of Computing Technology, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing Key Laboratory of Mobile Computing and Pervasive Device, No.6 Kexueyuan South Rd, Haidian, Beijing, China, 100190; Xiaolei Lv, yanjun.lx1@alibaba-inc.com; Binqiang Zhao, binqiang.zhao@alibaba-inc.com; Zheng Wang, lanjing.wz@alibaba-inc.com, Alibaba Group, Beijing, China; Zhaoqi Wang, zqwang@ict.ac.cn, Institute of Computing Technology, Chinese Academy of Sciences, No.6 Kexueyuan South Rd, Haidian, Beijing, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

2577-6193/2021/5-ART \$15.00

<https://doi.org/10.1145/3451267>

studies, we find that our generated indoor scene layouts for diverse, complicated indoor architecture are visually indistinguishable, which reach state-of-the-art performance.

**CCS Concepts:** • Computing methodologies → Computer graphics; • Shape analysis;

Additional Key Words and Phrases: indoor layouts, 3D indoor scene generation, convolution networks

#### ACM Reference Format:

Hao Jiang, Siqi Wang, Huikun Bi, Xiaolei Lv, Binqiang Zhao, Zheng Wang, and Zhaoqi Wang. 2021. Synthesizing Indoor Scene Layouts in Complicated Architecture Using Dynamic Convolution Networks. *Proc. ACM Comput. Graph. Interact. Tech.* 4, 1 (May 2021), 16 pages. <https://doi.org/10.1145/3451267>

## 1 INTRODUCTION

Modeling and synthesis of structured, real, and informative indoor scenes has become a challenging and essential task, especially for computational design, gaming entertainment, VR&AR applications, and robotic navigation training in virtual environments [Dai et al. 2018; Lai et al. 2014; Sünderhauf et al. 2017; Taira et al. 2018]. Since we spend significant amount of time indoors, the indoor scene layouts continue to attract the attention of interior design companies (e.g., IKEA, HOME-STYLER) [Hom [n.d.]; IKE [n.d.]; pla [n.d.]], and researchers in the field of computer graphics and vision [Li et al. 2019; Qi et al. 2018; Ritchie et al. 2019; Wang et al. 2019; Zhang et al. 2018]. The goal of indoor scene layout modeling and synthesis is to automatically, quickly, and realistically reconstruct virtual 3D indoor scenes, including different types of rooms (e.g., living room, dining room, bedroom), based on our daily habits and activities.

Research has recently emerged to synthesize indoor scenes, which aims at generating plausible arrangements of a set of objects. Although there are many effective approaches in some architectures, it is challenging to generate an indoor layout in complicated architecture, since both the human activities and the geometric and semantic information of indoor architecture are not comprehensively considered.

### 1). Geometric and semantic information of indoor architecture

Many previous works of indoor layout modeling and synthesis [Li et al. 2019; Ritchie et al. 2019; Wang et al. 2019] used SUNCG [Song et al. 2017], a comprehensive dataset of realistic indoor scenes encompassing scene types (e.g., living room, bedroom, kitchen, office). The majority of these top-down view rooms, are rectangular-shaped or L-shaped (Fig. 2). However, in reality, indoor scenes tend to have more complicated and diverse architecture (Fig. 2(d)). Such simplifications of the rectangular-shaped or L-shaped architecture for indoor scenes cannot satisfy various and realistic human needs in the real world. For more complicated indoor architecture, the implied geometric and semantic information might have a direct impact on indoor layouts. For example, a sectional sofa usually is placed against a wall whose length is longer than the sofa. Also, the indoor lighting caused by windows on the wall will influence the indoor layouts.

### 2). Human activities

State-of-the-art methods tend to arrange indoor objects one after another iteratively [Ritchie et al. 2019; Wang et al. 2018]. However, people are more accustomed to roughly dividing the room into several blocks based on their activities and indoor architecture, such as a reception block, a projection block, and a dining block as illustrated in Fig. 2. After that, a group of objects will be placed in each block. Thus, the strategy in these works, which treating objects as individuals, cannot utilize the geometric and semantic information of indoor architecture sufficiently. Some pioneering works [Fisher et al. 2015; Merrell et al. 2011] grouped objects with the specific functionality and proposed to utilize indoor structured blocks to facilitate specific human activities. To describe semantic relations among objects, some works recently exploited the spatial-based relations to organize functional groups of objects [Li et al. 2019; Wang et al. 2019]. There are only some specific

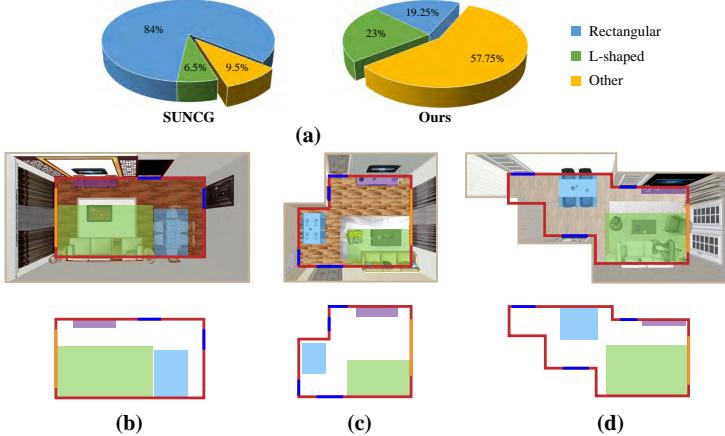


Fig. 2. Illustration of the statistical indoor architecture in SUNCG and our dataset. We randomly selected 400 indoor scenes from SUNCG and our dataset, respectively. The representative indoor architecture from SUNCG is (b) rectangular-shaped or (c) L-shaped. However, indoor scenes in reality tend to have more complicated and diverse architecture, shown in (d). The purple, green, and blue blocks refer to the projection block, reception block, and dining block, respectively. The yellow and blue lines on the red outline of each room represent the windows and doors, respectively.

indoor architectures and human activities being considered in the prior works, which are not sufficient for all indoor scenes.

To address these problems, we build an effective and novel framework to generate plausible indoor scene layouts. We assume that any indoor scene with complicated architecture is referred to as an irregular top-down view polygon. Given an indoor architecture around with a varied number of walls, we first encode the geometric and semantic information of indoor architecture into our proposed representation, called InAiR. Indoor objects to be arranged are organized by functional blocks, represented with oriented bounding boxes (OBBs), based on their functionality, human activities, and spatial relations. Each functional block here is assumed to be relative to a wall, called anchor-relative wall. We then employ dynamic convolution networks to predict the anchor-relative wall for given functional blocks, based on the captured multi-level geometric and semantic features of the indoor architecture. Then, the detailed layout of the functional block will be computed based on the selected anchor-relative wall. The final indoor layout will be generated after replacing each functional block OBB with a group of 3D shape objects.

The main contributions of this work include: i) we propose an intuitive, structured indoor architecture representation, called InAiR, to extract geometric and semantic information of realistic and complicated indoor architecture quickly and automatically. To our best knowledge, we are the first to encode the geometric and semantic information of complicated indoor architecture, not limited to rectangular-shaped or L-shaped architecture, to generate indoor scene layouts. ii) An effective and novel framework is built to synthesize indoor layouts of complicated architecture, which is satisfied with human habits and activities. Using dynamic convolution networks, the generated layout can jointly consider multi-level features. Through comparisons with state-of-the-art methods, our model achieves superior performance, especially in various challenging realistic indoor scenes.

## 2 RELATED WORK

There has been significant research on the indoor scene layout synthesis. Prior works synthesized indoor scene layout mainly focuses on modeling object-object relationships, as well as the occurrence and arrangement of objects within a room. Early work adopted rule-based constraints [Xu et al. 2002] and optimization of the cost functions based on design principles [Merrell et al. 2011; Yu et al. 2011] to model the object-object relationships. Gaussian mixtures [Fisher et al. 2015], relations annotations [Fu et al. 2017], and graphical models [Henderson et al. 2017] are frequently used to introduce the spatial relations of a group of objects.

With the availability of 3D indoor scene dataset (e.g., SUNCG [Song et al. 2017]), recent works on indoor scene synthesis proposed learning-based methods. Zhang et al. utilized a human-centric probabilistic grammar model to synthesize 3D room layouts [Qi et al. 2018]. Zhang et al. built a generative model using a feed-forward neural network that maps a prior distribution to the distribution of primary objects in indoor scenes for indoor environments synthesis [Zhang et al. 2018]. Other research showed the effectiveness of image-based deep convolutional generative models on indoor scenes generation [Ritchie et al. 2019; Wang et al. 2018]. The generative recursive autoencoders, called GRAINS, were designed to perform indoor objects grouping in the encoding and decoding phase [Li et al. 2019]. Combining a high-level relation graph representation with spatial prior neural networks, Wang et al. proposed a novel “plan-and-instantiate” conceptual framework for layout generation [Wang et al. 2019]. However, there has been little focus on synthesizing indoor scenes, especially generating layouts with more complicated architectures, which meet human habits and activities.

## 3 OVERVIEW

In this work, our goal is to build an effective and robust framework to generate plausible indoor scene layouts, which is satisfied with diverse architecture and human living habits. A room in our work refers to a completely enclosed space using walls, doors, and windows. Therefore, the outline of a top-down view room may be an irregular polygon. Considering the functionality and practicability of room types, we limited focus on living rooms, dining rooms, and bedrooms in this paper. We note that living rooms and dining rooms usually are interconnected spaces in ground truth indoor scenes, and such rooms are also considered in our work. Furthermore, our framework is easily extended to more room types.

Given an indoor scene and a set of objects’ labels, our framework is decomposed into four steps. Fig. 3 illustrates the pipeline of our framework to synthesize indoor scene layouts in complicated architecture.

*Indoor Architecture Representation.* To describe the complicated indoor architecture sufficiently, we first encode the given indoor scene with complicated architecture to a specific representation InAiR (see Sec. 4.1). All the geometric and semantic information of dynamic walls in architecture can be captured.

*Assign Functional Blocks and Corresponding Objects.* In order to describe the functionality and spatial relations of objects, we propose to use a functional block to organize a group of indoor labeled objects (see Sec. 4.2). Each functional block is represented by an OBB and structured to facilitate specific human activities, based on the statistics and analysis of indoor scene layouts. For example, a living room typically includes a reception block and a projection block, a dining room includes a dining block, a bedroom includes a sleeping block and a wardrobe block, etc.

We enable users to design several functional blocks to be arranged based on their intents and select the corresponding objects’ labels in each functional block. It is noted that all the indoor objects are

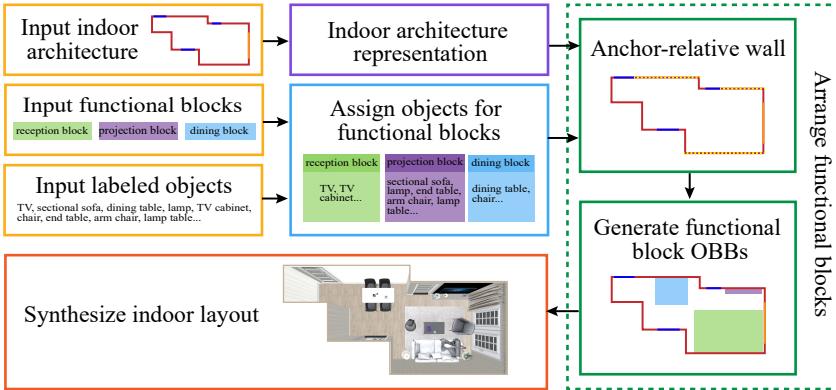


Fig. 3. Overview of our framework. Given an indoor architecture, several functional blocks, a group of labeled objects, our method encoded the indoor scene into a specific representation and organizes the objects with functional blocks based on human activities. Considering the rich geometric and semantic information of indoor architectures, our method proposes to exploit local and contextual features to synthesize indoor layouts in complicated architectures.

arranged by functional blocks. We build the corresponding functional block database, where each functional block organizes a group of 3D objects involved in certain spatial relations.

*Arrange Functional Blocks.* We assume that the location of each functional block is relative to a wall, defined as the anchor-relative wall. All the anchor-relative walls for functional blocks are computed simultaneously (see Sec. 4.2). We operate dynamic convolution networks on architecture representation to capture multi-level geometric and semantic features of room architecture, which is represented with an irregular polygon. The anchor-relative walls contribute to locating functional blocks roughly. Then, the extracted multi-level features of room architecture, as well as the embedded representation of the selected anchor-relative wall, are concatenated to compute the details of functional block OBBs, such as relative positions, size, offset, and orientations (see Sec. 4.2). The functional block OBBs are then arranged one after another.

*Synthesize Indoor Layouts.* To finalize the indoor scene layout, the generated functional block OBBs are replaced with a 3D functional block retrieved from our functional block database based on the functional block category including objects, and size information of the functional blocks.

## 4 METHODOLOGY

In this section, we describe our indoor architecture representation and present the framework for synthesizing indoor scene layouts in complicated architecture. The main symbols and corresponding explanations are shown in Table 1.

### 4.1 Indoor Architecture Representation InAiR

The rooms' architecture is a common condition that has a considerable impact on the overall indoor layouts. The indoor architecture representation, named *InAiR*, is specially proposed for a room with around  $N$  irregular walls (including the main wall, and partition wall, etc.). Each wall has both geometric and semantic information.

Each top-down view room is referred to as an irregular polygon, whose outline represents the walls of the room. For any wall  $i, (i \in [1, N])$  of room  $\tau$ , its geometric information includes length  $l^i$

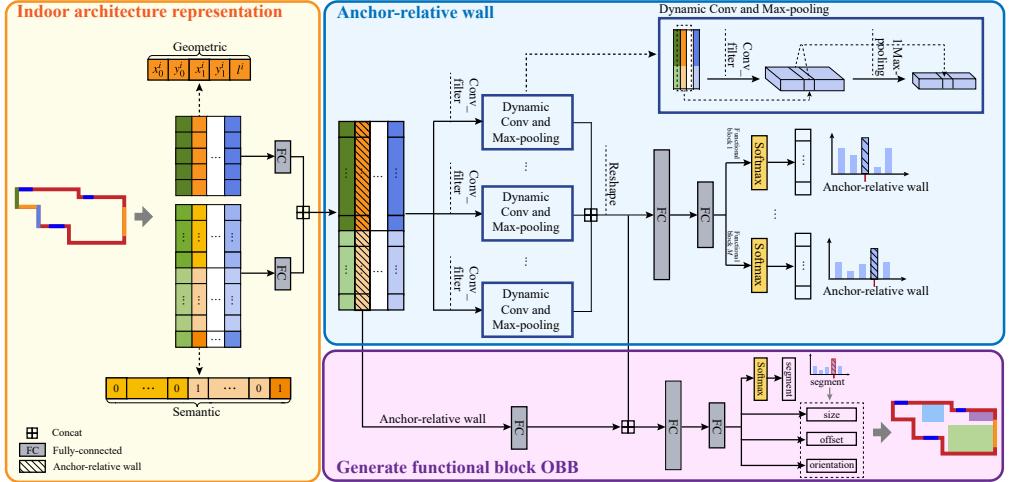


Fig. 4. Architecture of our network to generate functional block OBBs. The geometric and semantic information of the indoor architecture is extracted, encoded, and concatenated, shown in the yellow shaded region, which is then fed into the following anchor-relative wall module. We predict the anchor-relative wall for the corresponding functional block to be arranged, shown in the blue shaded region. Using the encoded information of the selected anchor-relative wall and the embedded geometric and semantic information of the indoor architecture, we predict the corresponding relative position, size, offset, and orientation of the functional block, shown in the purple shaped region.

Symbol	Explanation
$\tau$	Room
$N$	The number of walls.
$e_g^i$	The geometric information of wall $i$ .
$e_s^i$	The semantics of wall $i$ .
$e^i$	The geometric and semantic embedding of wall $i$ .
$M$	The number of functional blocks.
$E^\tau$	The embedded architecture of room $\tau$ .
$C^\tau$	The extracted features of room $\tau$ after filters and max-pooling.
$\gamma$	Functional block
$I^\gamma$	The selected anchor-relative wall for the functional block $\gamma$ .
$O^\gamma$	The corresponding OBB for the functional block $\gamma$ .

Table 1. The main symbols and corresponding explanations.

and positions, referring to 5 real numbers. It is noted that the wall with a door will be divided into two walls, as shown in Fig. 4. The position is represented with its endpoints  $(x_0^i, y_0^i)$  and  $(x_1^i, y_1^i)$ . The semantic information is represented with 13 binary indicators (4 bits for the type of walls; 8 bits for the type of connected rooms; 1 bit for windows on the wall). Namely, representation information for each wall is stored in an 18-dimensional vector. Therefore, the architecture of this room is represented with  $X^\tau \in R^{18 \times N}$ .

Room	Functional block	Object	Illustration of the representative objects of each functional block				
	Reception	sofa (sectional sofa, two seater sofa...), arm chair, swivel chair, end table, lamp, side table, accessory (plants, rug, cup, pillow...),...					
Living and dining	Projection	TV, TV cabinet, accessory (plants, vases ...),...					
Dining		dining table (round, square, ...), chair, armchair, accessory (plates, bowls, ...),...					
Sleeping		bed (king size, queen size, ...), night stand, lamp, accessory (vases, pillows, ...),...					
Bedroom	Wardrobe	wardrobe, (L-shaped, corner,...)					

Table 2. The functional blocks and some corresponding objects. Based on the statistical results of our dataset, we show that there are three main functional blocks in the living and dining room. They are reception blocks, dining blocks, and projection blocks. The representative functional blocks in bedrooms are sleeping blocks and wardrobe blocks.

## 4.2 Functional Block Module

Indoor scenes often contain functional groups of objects, which are structured to facilitate specific human activities. In our work, we refer these grouped objects to *functional blocks*, represented with OBBs. Based on the statistical results of our dataset and room types, we show all the functional blocks appeared in our synthetic indoor scene and some corresponding objects in Table 2. Given an input of an indoor scene and a set of objects' labels, we encourage users to design several functional blocks and assign the corresponding objects to each functional block based on their intents and activities. All the indoor objects are arranged by functional blocks. We assume that there are  $M$  functional blocks to be arranged in the room  $\tau$ .

A functional block is placed based on its relative position, size, offset, and orientation information. We assume each functional block is relative to a wall, named anchor-relative wall. For example, a projection block including TV and a reception block with sofa in the living room are likely to be placed adjacent to a wall; all beds in sleeping blocks have at least one side leaning against a wall, etc.

Based on our proposed indoor architecture representation InAiR, we exploit dynamic convolution networks to predict anchor-relative walls for all the functional blocks simultaneously. Next, the functional block OBBs, including detailed position, size, offset, and orientation information relative to the respective predicted anchor-relative wall, are generated. Finally, each designed functional

block OBB is replaced with a corresponding 3D functional block retrieved from our functional block database based on the functional block category including objects, and size information of the functional blocks.

We exploit a hierarchical clustering method to create functional blocks other than grouping objects manually. Based on the definition of functional blocks, a group of objects are clustered into a functional block according to their spatial relations and human activities with respect to these objects. We use linkage criteria to get the smallest cluster distance among these objects. The clustering process will not stop until the cluster's diameter of a cluster exceeds the pre-defined threshold.

*Anchor-relative Wall.* To deal with the effect of indoor architecture, we arrange each functional block based on the combination of the geometric information of each wall and the contextual semantic information of a set of sequential walls. For example, a long sofa bed fails to be placed against a wall with length that is shorter than the length of the sofa, and the arrangement of a corner sofa relies on the contextual semantic information of the two adjacent walls. We divided the features of the indoor architecture into two categories: local features and contextual features. We utilized dynamic convolution networks to capture the local features and contextual features of the indoor architecture. The dynamic convolution operation is frequently used in natural language processing (NLP) to induce lexical-level automatically and sentence-level features from plain texts for event extraction [Baevski and Auli 2018; Chen et al. 2015; Kim 2014; Kim et al. 2016; Zhang and Wallace 2015]. Inspired by these works, we employ multiple dynamic convolution networks to obtain valuable multi-level features within architecture.

We used five variables to describe the geometric information of each wall in the room, which contains more diverse latent information than the semantic binary indicators. Given the architecture representation  $X^\tau \in R^{18 \times N}$  of room  $\tau$ , the geometric information of wall  $i$  are first embedded into a high-dimensional space  $e_g^i$ . The semantics of wall  $i$  is transformed into a low-dimensional embedding  $e_s^i$ . The geometric and semantic embeddings are then concatenated as the input  $e^i$  of the following modules, as shown in Fig. 4. The embedded architecture of room  $\tau$  is  $E^\tau$  with shape of  $h \times N$ .

The convolution layer aims to capture multi-level compositional semantics of a group of walls and compress this valuable information into feature maps. Here, a convolution operation involves  $n$  filters (size of  $h \times w$ ), which is applied to a window of  $w$  walls to extract a new semantic feature<sup>1</sup>. Each filter is operated on the whole information of several geometrically consecutive walls. The shape of the feature after every  $n$  convolution filters with a fixed  $w$  for room  $\tau$  is  $1 \times N \times n$ . In order to consider different level semantics of architecture, we utilize a set of filters with a window size  $w \in \{f_1, f_2, \dots, f_m\}$  for convolution.

As the shape of the feature after convolution is varied with the number of walls ( $N$ ) in  $\tau$ . We then take a max-pooling to extract the most valuable information from  $N$  walls. Using the max-pooling, the features of the varied architecture in each dynamic convolution network are transformed into a  $1 \times 1 \times n$  feature. Therefore, the concatenated features after filters with size  $w \in \{f_1, f_2, \dots, f_m\}$  and max-pooling can represent the multi-level geometric and semantic features of the architecture. We reshape it into a vector with a shape of  $mn \times 1$ , denoted by  $C^\tau$ . The dynamic geometric and semantic information of indoor architecture is turned into fixed dimensions.

For the functional block  $\gamma$  ( $\gamma \in 1, M$ ) to be arranged in room  $\tau$ , we feed the vector after two fully connected layers into a softmax layer. We compute the probabilities of each  $i$  to be the anchor-relative

---

<sup>1</sup>In order not to lose the information on the edge of  $E^\tau$ , for any wall  $i$  (when  $i < \lceil \frac{w}{2} \rceil$ ), we sequentially pad the embedded information of the adjacent  $\lceil \frac{w}{2} \rceil - i$  walls for wall  $i$ . Similarly, when  $i > N - \lceil \frac{w}{2} \rceil$ , the embedded information of the adjacent  $\lceil \frac{w}{2} \rceil - i - N$  walls for wall  $i$  will be padded for subsequent wall  $N$ .

wall. To predict the anchor-relative wall for the functional block  $\gamma$  in room  $\tau$ , we employ the cross-entropy loss:  $L_{anc} = \sum_{\gamma=1}^M \sum_{i=1}^N (p_{anc}^{i,\gamma} - \hat{p}_{anc}^{i,\gamma})^2$ , where  $p_{anc}^{i,\gamma}$  is the ground truth, and  $\hat{p}_{anc}^{i,\gamma}$  is predicted probability of wall  $i$  that is the anchor-relative wall for the functional block  $\gamma$ .

*Generate Functional Block OBBs.* We use OBBs to represent functional blocks. Based on the selected anchor-relative wall  $I^\gamma$  for the functional block  $\gamma$ , we compute the relative position, size, offset, and orientation information of the functional block  $\gamma$  to place the corresponding OBB  $O^\gamma$ .

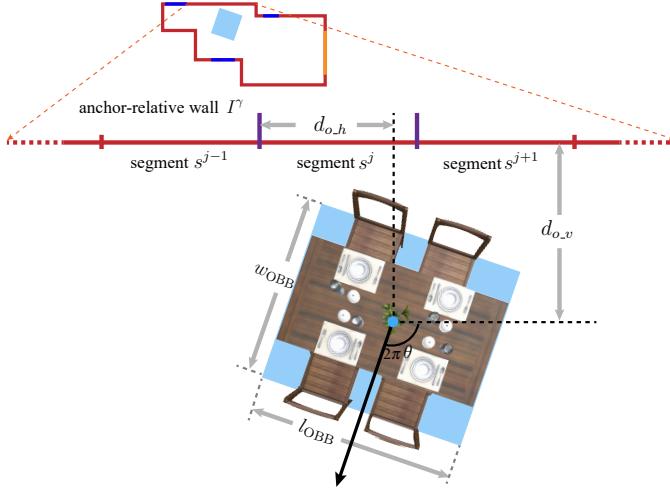


Fig. 5. Illustration of a functional block OBB respect to the predicted anchor-relative wall. The dining block, used as an example, comprises four chairs and a dining table.

The arrangement of  $O^\gamma$  respect to  $I^\gamma$  needs to take the overall architecture of  $\tau$  into consideration, as well as the local information of  $I^\gamma$ . Therefore, the embedding of  $I^\gamma$  with semantic and geometric information, which is denoted by  $e^{I^\gamma}$ , is passed into a fully connected layer, and then concatenated with  $C^\tau$ , which carries the overall architecture semantics of  $\tau$ . After two fully connected layers, the vector is passed to the output layer.

To arrange  $O^\gamma$  more accurately, we divided the anchor-relative wall  $I^\gamma$  into  $t$  segments, shown in Fig. 5. We predict each probability of  $O^\gamma$  in segment  $s^j$  with  $j \in [1, t]$ , respectively, as well as the size, offset, and orientation of  $O^\gamma$  in the segment with the highest probability. We described a OBB using the width  $w_{OBB}$  and the length  $l_{OBB}$ . We introduce the offset with two offset values—one (i.e.,  $d_{o,h}$ ) in horizontal direction, and the other (i.e.,  $d_{o,v}$ ) in vertical direction, as shown in Fig. 5. The orientation is encoded using a 5-bit one-hot vector and a parameter  $\theta \in [0, 1]$  (that is a real number). The first 5 bits are designed to indicate whether the functional block OBB is oriented at an angle of  $0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$ —respect to the anchor-relative wall  $I^\gamma$ —or none of the above, where  $p_{ori}^k$  with  $k \in [1, 5]$  denotes each bit. In the oriented case (i.e.,  $k = 5$ ), the functional block will be rotated  $2\pi\theta$  (unit: rad) about the center (Fig. 5).

There are four loss terms when generating functional block OBBs. We noted that the variables with a hat denote predicted values, and that without a hat denote ground truth. The losses for size, offset, and orientation of  $O^\gamma$  only focus on the OBB in the segment with the highest probability. We predefined the function  $\Phi(a, b) = (\sqrt{|a|} - \sqrt{|b|})^2$ .

- We use cross-entropy loss to show the probabilities in all segments:  $L_p = \sum_{j=1}^t (p_{seg}^j - \hat{p}_{seg}^j)^2$ , where  $\hat{p}_{seg}^j$  is the predicted probability of  $O^\gamma$  located in segment  $s^j$ .

- Loss of the OBB’s size:  $L_{size} = \sum_{j=1}^t \mathbb{1}_j^{obj} (\Phi(w_{OBB}^j, \hat{w}_{OBB}^j) + \alpha \Phi(l_{OBB}^j, \hat{l}_{OBB}^j))$ , where  $\mathbb{1}_j^{obj}$  denotes the functional block OBB  $O^j$  appears in segment  $s^j$ .
- Loss of the OBB’s offset:  $L_{offset} = \sum_{j=1}^t \mathbb{1}_j^{obj} (\Phi(d_{o\_h}^j, \hat{d}_{o\_h}^j) + \beta \Phi(d_{o\_v}^j, \hat{d}_{o\_v}^j))$ .
- Loss of the OBB’s orientation is the sum of a cross-entropy loss and a reconstruction loss:  $L_{ori} = \sum_{j=1}^t \mathbb{1}_j^{obj} (\sum_{k=1}^5 (p_{ori}^{j,k} - \hat{p}_{ori}^{j,k})^2) + \eta \sum_{j=1}^t \mathbb{1}_j^{obj} \Phi(\theta^j, \hat{\theta}^j)$ .

Finally, the total loss of the network for generating each functional block OBB  $O^j$  respect to the corresponding selected anchor-relative wall  $I^j$  is defined as follows:

$$L_{total} = L_p + \mu_1 L_{size} + \mu_2 L_{offset} + \mu_3 L_{ori}, \quad (1)$$

where  $\mu_1$ ,  $\mu_2$ , and  $\mu_3$  are the balance controllers.

### 4.3 Synthesize Indoor Layouts

We used the function block OBB’s given size information to iteratively detect the existence of collision between the functional block OBB and the existing functional block OBBs. Our collision detection involves both broad phase and narrow phase. We begin by the broad phase, which used AABB (Axis-Aligned Bounding Box) to detect any possible collision. If a collision exists, SAT (Separating Axis Theorem) is then used for further detection [Bergen 2003]. When the collision had been detected, we would resample another anchor-relative wall and OBB’s information (size, offset, and orientation). After arranging all the functional block OBBs, we prepared some 3D functional blocks as candidates using the functional block category including objects via a nearest neighbor search.

## 5 EXPERIMENT

In this section, we compared our method with the state-of-the-art methods and present quantitative and qualitative evaluation results. In our work, we trained our model on the 3D-Front dataset<sup>2</sup>. We used 5000 epochs to train the anchor-relative wall module and 3000 epochs to train the functional block OBB module. And we chose  $\mu_1 = 0.5$ ,  $\mu_2 = 0.5$ , and  $\mu_3 = 0.3$  for (1). We summarize the additional details of the network in Table 3. The model was trained on a server with two P100 graphics cards and a 16-core CPU @2.50GHz. The training process took about 10 hours.

### 5.1 Results of Functional Blocks

Fig. 6 shows the predicted results of the anchor-relative wall, when arranging a reception block in a living room with distinct irregular-shaped architecture. Qualitatively, our method does a good job of predicting the anchor-relative wall, where the reception function area is located against all walls in various real complicated architecture. We note that it is essential and necessary for arranging the reception blocks with a specific size.

Fig. 7 shows the predicted results of the reception block in living rooms with various indoor architecture. We found that the location and size of our predicted functional blocks are plausible. We noted that in the cases with incorrect labels caused by manual labeling, our method still can generate robust and reasonable layout results, as shown in Figs. 7(d), 7(g), and 7(h).

Fig. 8 shows the comparison of the 2D layout of two living-dining rooms, which involve dining blocks, reception blocks, and projection blocks to be arranged. Each comparison is shown side by side: the left one is our synthesized layouts, and the right one is the one designed by designers. We found that the significant factors—architecture, semantics, and human activities—have been taking into accounts in our synthesized layouts, which is consistent with the intention and ideas of designers.

<sup>2</sup>The information of 3D-Front dataset can be found online at <https://tianchi.aliyun.com/specials/promotion/alibaba-3d-scene-dataset>

Layer Type	Input (dimensions)	Output (dimensions)	Additional Parameters
<b>Indoor Architecture Representation</b>			
Fully-connected	$X_{\text{geometric}}^{\tau}, 5 \times N$	$E_g^{\tau}, 48 \times N$	-
Fully-connected	$X_{\text{semantic}}^{\tau}, 13 \times N$	$E_s^{\tau}, 8 \times N$	-
Concat.	$E_g^{\tau}, 48 \times N, E_s^{\tau}, 8 \times N$	$E^{\tau}, 56 \times N$	-
<b>Dynamic Convolution and Max-pooling</b>			
(The parameters of only one dynamic convolution and max-pooling module are shown here. The filter size of each convolution is different, referred to $f_m$ . We separately employ $m = 9, f_m \in \{1, 3, 5, \dots, 15, 17\}$ modules to extract features.)			
Convolution	$E^{\tau}, 56 \times N$	$\text{conv}_m, 1 \times N \times 200$	act:=ReLU, kernel:=(56, $f_m$ ), $f_m \in \{1, 3, 5, \dots, 15, 17\}$ , stride:=(1,1), filter number:=200
1 Max-pooling	$\text{conv}_m, 1 \times N \times 200$	$\text{pool}_m, 1 \times 1 \times 200$	-
Concat.	$\text{pool}_m, 1 \times 1 \times 200, m = 1, \dots, 9$	$\text{pool}_m, 1 \times 9 \times 200$	-
Reshape	$\text{pool}_m, 1 \times 9 \times 200$	$C^{\tau}, 1800 \times 1$	-
<b>Anchor-relative Wall</b>			
Fully-connected	$C^{\tau}, 1800 \times 1$	$fc_1^{\tau}, 900 \times 1$	act:=ReLU
Fully-connected	$fc_1^{\tau}, 900 \times 1$	$fc_2^{\tau}, n_r * M \times 1$	act:=ReLU, for living and dining room, $n_r = 36$ ; for bedroom, $n_r = 18$
Softmax	$fc_2^{\tau}, n_r * M \times 1$	$p_{\text{anc}}, n_r * M \times 1$	-
<b>Functional Block OBBs</b>			
Fully-connected	$e^{\tau}, 56 \times 1$	$fc_3^{\tau}, 200 \times 1$	act:=ReLU
Concat.	$C^{\tau}, 1800 \times 1, fc_1^{\tau}, 200 \times 1$	$fc_2^{\tau}, 2000 \times 1$	-
Fully-connected	$fc_2^{\tau}, 2000 \times 1$	$fc_3^{\tau}, 1000 \times 1$	act:=ReLU
Fully-connected	$fc_3^{\tau}, 1000 \times 1$	$fc_4^{\tau}, 110 \times 1$	act:=ReLU
Fully-connected & Softmax	$fc_4^{\tau}, 110 \times 1$	$p_{\text{seg}}, t \times 1$	$t = 10$
Fully-connected	$fc_4^{\tau}, 110 \times 1$	$OBB_{\text{size}}, 2t \times 1$	-
Fully-connected	$fc_4^{\tau}, 110 \times 1$	$OBB_{\text{of fset}}, 2t \times 1$	-
Fully-connected & Softmax	$fc_4^{\tau}, 110 \times 1$	$OBB_{\text{ori}}, 5t \times 1$	-
Fully-connected	$fc_4^{\tau}, 110 \times 1$	$OBB_{\theta}, t \times 1$	-

Table 3. Detailed architecture of our network.

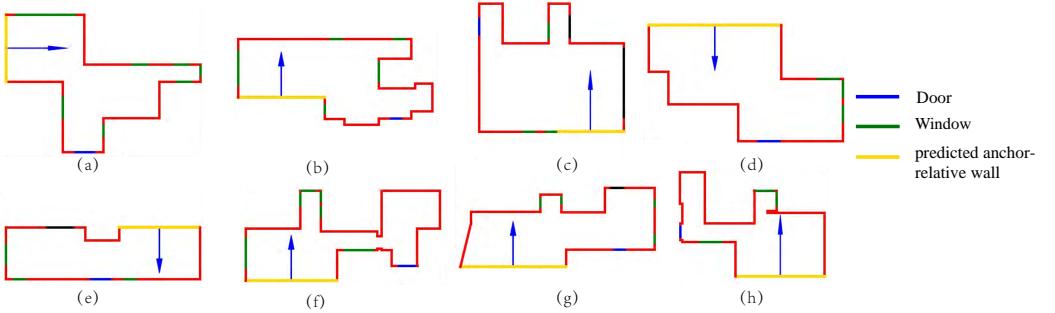


Fig. 6. The illustration of the predicted anchor-relative walls for arranging the reception blocks with our method. The red lines represent the wall. The yellow lines indicate the predicted anchor-relative wall and the blue arrows indicate the center and orientation of the corresponding anchor-relative walls. We note that our method can accurately predict anchor-relative walls in quite complicated architectures, which is consistent with the ground truth.

## 5.2 Synthesizing New Indoor Layouts

We compare our method with Fast & Flexible method [Ritchie et al. 2019] and present some synthetic indoor layouts for living and dining rooms and bedrooms in Fig. 9. As compared in the top three

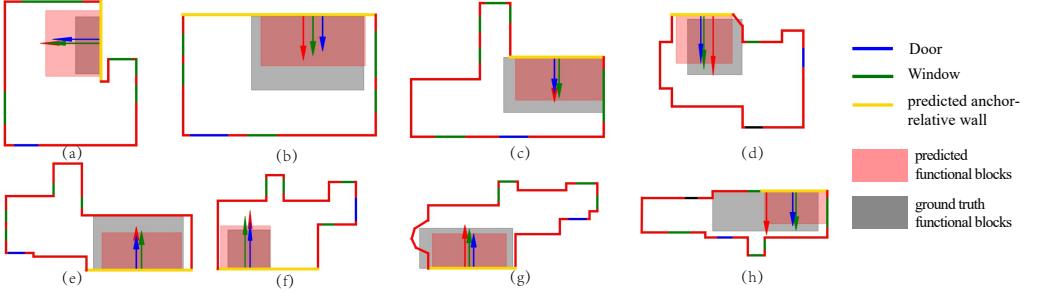


Fig. 7. Illustration of the predicted reception blocks with our method. The gray rectangles represent the manually labeled locations of the reception blocks, and the red arrows represent where the labeled blocks are pointing to each of the corresponding anchor-relative walls. The red rectangles represent the predicted results by our method. The green arrows and the blue arrows indicate the predicted top two segments to arrange blocks.

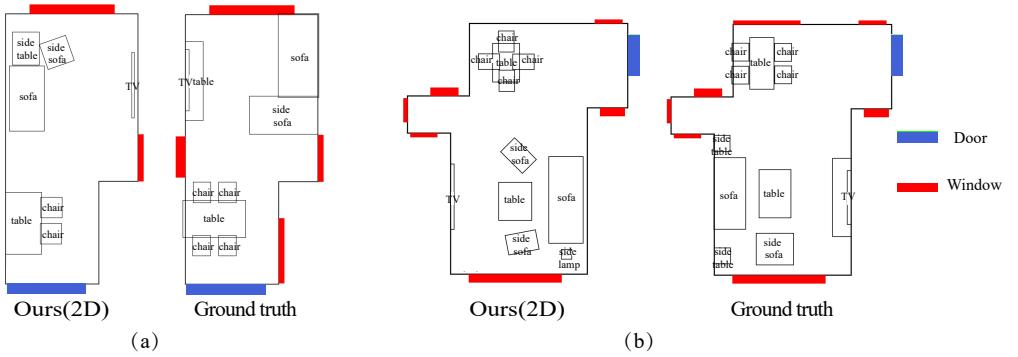


Fig. 8. Comparisons of our 2D synthesized indoor layouts and the ground truth. There are dining blocks, reception blocks, and projection blocks to be arranged. In each comparison, the left one is our synthesized layouts and the right one is the one designed by designers.

rows in Fig. 9, Fast & Flexible method fails to generate plausible indoor layouts of complicated architecture. By an iterative strategy to arrange objects one after another, Fast & Flexible was unable to compute reasonable arrangement of some given objects (Fig. 9(a)). Some objects are placed in the center of the room (Fig. 9(c)), neglecting spatial relations. Using multi dynamic convolution networks, our proposed method can capture multi-level features of the indoor architecture. Our synthetic indoor layouts are more plausible and reasonable for indoor scenes with complicated architecture. More synthesized 2D and 3D indoor layouts in Fig. 10 and Fig. 11.

### 5.3 Paired Comparison User Study

We designed a paired comparison user study to evaluate the synthetic indoor layouts using our method. We randomly selected 58 scenes, including 40 living & dining room, and 18 bedrooms. We generated top-down view images for this user study which combined with the other two groups of scenes, respectively. This yield  $2 \times 58$  comparison pairs (ours vs. ground truth and ours vs. fast & flexible, respectively), scenes from one pair are based on the same outline.

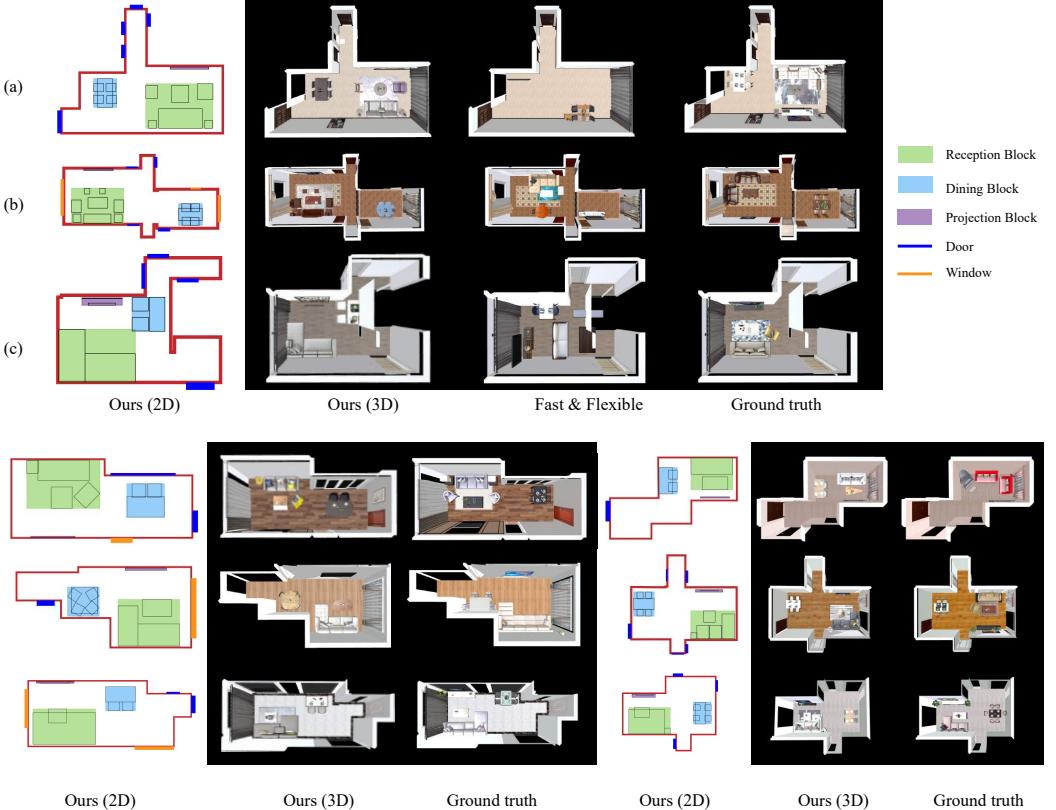


Fig. 9. Comparisons of synthesized indoor layouts. Our 2D, 3D generated results are compared with Fast & Flexible [Ritchie et al. 2019] method and ground truth in the top three rows. The bottom three rows illustrate additional compared examples.

Room type	Ours vs.	
	Fast & Flexible	Ground truth
Living and dining	$65.00 \pm 13.51$	$41.00 \pm 8.38$
Bedroom	$63.89 \pm 14.75$	$41.67 \pm 11.45$

Table 4. The experimental results of our user study that indicates the distribution of the percentage ( $\pm$  standard error) of forced-choice comparisons.

We recruited  $2 \times 10$  participants for these two comparisons, who are all graduate students in the university, to participate in our paired comparison user study. Avoiding making forced and inaccurate perception votes, they are required to perceptually select which functional block and the overall layout are more plausible based on the given two top-down view rendered scenes. To avoid the effect of material or texture appearance, the objects rendered in compared scenes here are shown with solid colors. We use one-sample t-tests to determine the Confidence Interval (CI), and paired-sample

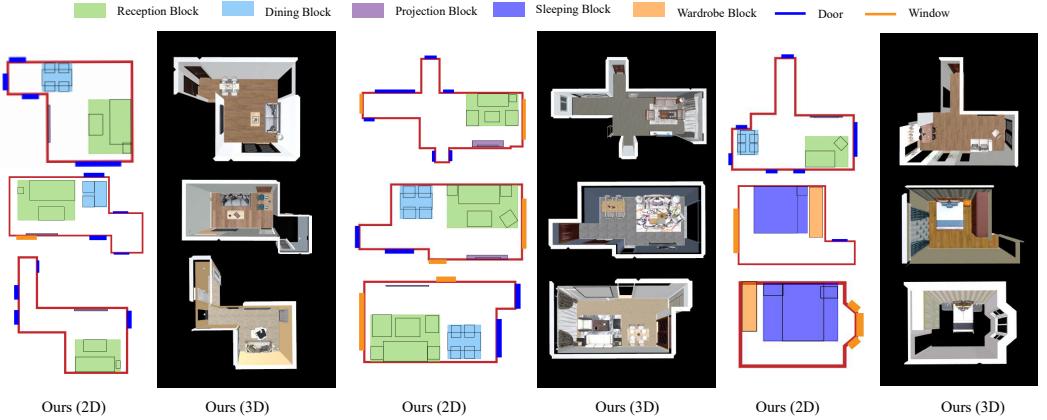


Fig. 10. Additional synthesized 2D and 3D indoor layouts.



Fig. 11. Additional synthesized 3D indoor layouts.

t-tests to compare the difference of the true mean of two sets of data with 95% confidence. The conventional significance for the entire analysis was determined at  $\alpha = 0.05$ , two tailed.

Table 4 shows the participants' preference between the compared scenes. Compared with the Fast & Flexible, our results are preferred for both types of rooms. We found that our method is more robust when synthesizing indoor layouts of more complicated and challenging architecture. Comparing with the professionally designed scenes produced by layout designers, our method is less preferred instead. We suggested that this might be caused by a more accurate location of objects (e.g., beds are usually leaning up against a wall closely without any gaps), which is more visually sensitive to humans, especially in 2D top-down view plans. Producing all the accurate locations for all the objects is still challenging.

We conduct the second user study to verify our analysis. Ten participants (who have not participated in the previous user study) were required to rate 40 3D indoor layouts with a random order on a scale of 1 to 5, 1 means the least plausible, and 5 means the most plausible). The layouts to be rated include 20 of our synthetic results and the corresponding ground truth, which comprise 15 living-dining rooms and 5 bedrooms. In the living and dining room, the mean of the raw scores for our method is  $3.59 \pm 0.79$ , and the mean of the ground truth is  $3.75 \pm 0.72$ . In the bedroom, the mean of the raw scores for our method is  $4.12 \pm 0.66$ , and the mean of the ground truth is  $4.14 \pm 0.58$ . The experimental results show that the proportion of ours is  $48.89\% \pm 10.72\%$  in living-dining rooms and  $49.88\% \pm 8.05\%$  in bedrooms. From this experiment, we found that there is no evidence of significant preference for the layout results produced by professional layout designers.

From the two user studies, we observed that: (1). Due to the human sensitivity and subtle differences, users are capable of selecting a better result in a paired comparison of the two layout results,

which makes it challenging to produce highly accurate and robust indoor layouts in complicated architectures. (2). However, the layout results using our method is preferred compared to the fast & flexible (i.e., the state-of-the-art method). (3). Additionally, when we randomly mixed our synthetic indoor scene results with the professionally designed layouts, the two types of results were given a very consistent evaluation, indicating that our synthetic results are visually consistent and highly acceptable for the participants.

## 6 CONCLUSION

In this paper, we focus on indoor scene layouts synthesis in complicated architecture. Unlike the previous works generating indoor layouts in rectangular or L-shaped rooms, we propose an intuitive, and structured indoor architecture representation, called InAiR, to extract geometric and semantic information of indoor architecture. The outputs of InAiR are then used in an effective and novel framework that is built to synthesize indoor layouts, where we employ dynamic convolution networks to capture local and contextual multi-level features of the indoor architecture. The objects to be arranged are organized into functional blocks, which facilitate specific human activities and habits. To our best knowledge, for the first time, we synthesized indoor layouts for complicated architecture. Through comparisons with state-of-the-art methods, our model achieves superior performance, especially in various challenging realistic indoor scenes.

In future work, adopting deep learning models with stronger representative capabilities might further improve our results, such as the dynamic aggregation convolution method proposed by Chen et al [Chen et al. 2020]. Due to the complexity of the human activities, it might also help considering both the macro-arrangement of furniture from a functional point of view and the indoor decoration style to meet human needs.

## 7 ACKNOWLEDGMENTS.

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2017YFB1002602, in part by the National Natural Science Foundation of China under Grant 62002345 and 61532002, and in part by the Alibaba Research Fellowship.

## REFERENCES

- [n.d.]. EasyHome HomeStyler. <https://www.homestyler.com/int/>. Accessed 2019-8-11.
- [n.d.]. The IKEA Home Planner. <https://www.ikea.com>. Accessed 2019-8-11.
- [n.d.]. Planner5d. <https://planner5d.com>. Accessed 2019-8-11.
- Alexei Baevski and Michael Auli. 2018. Adaptive input representations for neural language modeling. *arXiv preprint arXiv:1809.10853* (2018).
- Gino Bergen. 2003. *Collision Detection in Interactive 3D Environments*. <https://doi.org/10.1201/9781482297997>
- Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. 2020. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11030–11039.
- Yubo Chen, Liheng Xu, Liu Kang, Daojian Zeng, and Jun Zhao. 2015. Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks. In *Meeting of the Association for Computational Linguistics*.
- Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. 2018. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4578–4587.
- Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. 2015. Activity-centric Scene Synthesis for Functional 3D Scene Modeling. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 179.
- Qiang Fu, Xiaowu Chen, Xiaotian Wang, Sijia Wen, Bin Zhou, and Hongbo Fu. 2017. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 201.
- Paul Henderson, Kartic Subr, and Vittorio Ferrari. 2017. Automatic Generation of Constrained Furniture Layouts. *arXiv preprint arXiv:1711.10939* (2017).
- Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*.

- Yoon Kim, Yacine Jernite, David A Sontag, and Alexander M. Rush. 2016. Character-Aware Neural Language Models. In *AAAI*.
- Kevin Lai, Liefeng Bo, and Dieter Fox. 2014. Unsupervised feature learning for 3d scene labeling. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3050–3057.
- Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. 2019. GRAINS: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics (TOG)* 38, 2 (2019), 12.
- Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. 2011. Interactive furniture layout using interior design guidelines. In *ACM transactions on graphics (TOG)*, Vol. 30. ACM, 87.
- Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song Chun Zhu. 2018. Human-centric Indoor Scene Synthesis Using Stochastic Grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5899–5908.
- Daniel Ritchie, Kai Wang, and Yu-an Lin. 2019. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6182–6190.
- Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. 2017. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1746–1754.
- Niko Sünderhauf, Trung T Pham, Yasir Latif, Michael Milford, and Ian Reid. 2017. Meaningful maps with object-oriented semantic mapping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5079–5085.
- Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. 2018. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7199–7209.
- Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. 2019. PlanIT: planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 132.
- Kai Wang, Manolis Savva, Angel X Chang, and Daniel Ritchie. 2018. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 70.
- Ken Xu, James Stewart, and Eugene Fiume. 2002. Constraint-based automatic placement for scene composition. In *Graphics Interface*, Vol. 2. 25–34.
- Lap-Fai Yu, Sai Kit Yeung, Chi-Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley Osher. 2011. Make it home: automatic optimization of furniture arrangement. *ACM Trans. Graph.* 30, 4 (2011), 86.
- Ye Zhang and Byron C. Wallace. 2015. A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification. In *IJCNLP*.
- Zaiwei Zhang, Zhenpei Yang, Chongyang Ma, Linjie Luo, and Qixing Huang. 2018. Deep Generative Modeling for Scene Synthesis via Hybrid Representations. *arXiv preprint arXiv:1808.02084* (2018).