FuncScene: Function-centric Indoor Scene Synthesis via a Variational AutoEncoder Framework

Wenjie Min, Wenming Wu*, Gaofeng Zhang and Liping Zheng*

Hefei University of Technology, Hefei, Anhui 230601, PR China

ARTICLE INFO

Keywords:
Indoor scene synthesis
function group
neural network
generative model

ABSTRACT

One of the main challenges of indoor scene synthesis is preserving the functionality of synthesized scenes to create practical and usable indoor environments. Function groups exhibit the capability of balancing the global structure and local scenes of an indoor space. In this paper, we propose a function-centric indoor scene synthesis framework, named FuncScene. Our key idea is to use function groups as an intermedium to connect the local scenes and the global structure, thus achieving a coarse-to-fine indoor scene synthesis while maintaining the functionality and practicality of synthesized scenes. Indoor scenes are synthesized by first generating function groups using generative models and then instantiating by searching and matching the specific function groups from a dataset. The proposed framework also makes it easier to achieve multi-level generation control of scene synthesis, which was challenging for previous works. Extensive experiments on various indoor scene synthesis tasks demonstrate the validity of our method. Qualitative and quantitative evaluations show the proposed framework outperforms the existing state-of-the-art.

1. Introduction

We study the problem of indoor scene synthesis, which plays a significant role in the content creation of electronic games and virtual reality. The rapid expansion of digital industries has created an urgent demand for efficient indoor scene modeling. Hence, indoor scene synthesis has drawn increasing attention (Merrell, Schkufza, Li, Agrawala and Koltun, 2011; Yu, Yeung, Tang, Terzopoulos, Chan and Osher, 2011; Fu, Chen, Wang, Wen, Zhou and Fu, 2017) in the community of computer graphics and vision.

Recently, automatic indoor scene generation (Li, Patil, Xu, Chaudhuri, Khan, Shamir, Tu, Chen, Cohen-Or and Zhang, 2019; Zhang, Zhang, Xie, Luo, Yang and Fu, 2021a) made tremendous breakthroughs attributed to the development of deep learning techniques. Despite recent progress, there remains a significant gap between synthetic scenes and realistic scenes. One of the main challenges is preserving the functionality of synthesized scenes in order to create practical and usable indoor environments. Previous works have either focused on representing the whole scene as a relation graph, generating or optimizing scenes by sampling from the global representations (Zhou, While and Kalogerakis, 2019; Zhang, Yang, Ma, Luo, Huth, Vouga and Huang, 2020; Luo, Zhang, Wu and Tenenbaum, 2020), or enabled a progressive scene generation by iteratively inserting furniture items into an initial scene based on probabilistic models over layouts and geometries (Wang, Savva, Chang and Ritchie, 2018; Ritchie, Wang and Lin, 2019; Paschalidou, Kar, Shugrina, Kreis, Geiger and Fidler, 2021). The former struggles with generating fine-grained details and local relations, and the latter may face challenges in capturing global dependencies and overall scene structure. They all overlook the functional relations that inherently exist between the global structure and the local scene.

The functionality of indoor scenes refers to the categorization and arrangement of indoor furniture based on its purpose, function, and the users' needs. Emphasizing functionality means that indoor furniture must not only meet aesthetic and visual effects but, more importantly, fulfill the requirements of practical activities. Each piece of furniture and layout design should serve a specific purpose and activity; for example, dining tables for eating and sofa areas for rest and entertainment. On the one hand, functional indoor spaces, through thoughtful interior layout, ensure that every part of the interior is fully utilized while maintaining the convenience and comfort of areas. On the other hand, interior design must consider the users' habits and preferences, ensuring that the interior layout aligns with their usage patterns. Functional design should offer more solutions for personalized requirements, making indoor scenes better

*Corresponding authors: Wenming Wu, Liping Zheng
wwming@hfut.edu.cn (W. Wu); zhenglp@hfut.edu.cn (L. Zheng)
ORCID(s): 0000-0002-0640-8520 (W. Wu); 0000-0003-0536-7226 (G. Zhang); 0000-0001-5071-9628 (L. Zheng)

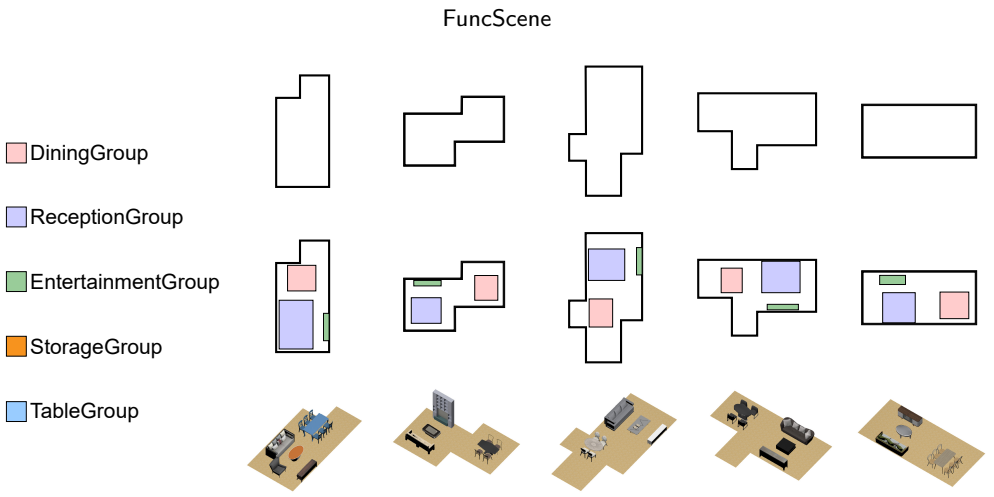


Figure 1: We introduce a novel model for generating indoor scenes, termed as FuncScene. Given a room boundary, FuncScene operates in two main steps. First, we generate function groups within the scene. Then, we instantiate these function groups by effectively searching for and aligning them with relevant instances from an indoor scene dataset. This approach enables us to achieve enhanced control over the generation of indoor scene designs, ultimately leading to improved quality of the generated scenes. Top: input room boundaries for indoor scene synthesis. Middle: generated function groups for given boundaries. Bottom: final synthesised indoor scenes.

aligned with individual preferences. This function-centric hierarchy exhibits the balance of the global scene structure and local details, which is actually very conducive to achieving indoor scene synthesis. This also aligns with the fundamental strategy employed by professional interior designers in the real world, first establishing the functional zoning of space before moving on to the specific design details, rather than individually designing each furniture item.

In this paper, we propose FuncScene, a novel function-centric indoor scene synthesis framework. Our key idea is centred around utilizing function groups as an intermediary to bridge the gap between the local scenes and the global structure. FuncScene enables a multi-level synthesis, gradually refining scene synthesis while maintaining the functionality and practicality of scenes. First, function groups of indoor scenes are generated to obtain an overview of the scene layout. Synthesis diversity is enabled by incorporating a VAE (Variational Autoencoder) into the generation framework. Then, with function groups as the reference, we instantiate the generated function groups by systematically searching and matching them with the specific instances from our indoor scene dataset. Furniture items within each function group are obtained to capture the layout details of the indoor scene.

Extensive experiments on various indoor scene synthesis tasks demonstrate the validity of our method, as shown in Figure 1. Qualitative and quantitative evaluations show the proposed framework outperforms the state-of-the-art methods. Our method also makes it easier to achieve multi-level generation control, providing flexibility and adaptability in generating indoor scenes with desired characteristics and properties. We contribute the following: (i) a novel function-centric indoor scene synthesis framework with VAE for synthesizing multiple functional indoor scenes and (ii) a multi-level generation control for indoor scene synthesis, which is challenging for previous methods.

2. Related work

2.1. Progressive indoor scene synthesis

Most of the existing models (Kermani, Liao, Tan and Zhang, 2016; Wang et al., 2018; Ritchie et al., 2019; Wang, Yeshwanth and Nießner, 2021; Paschalidou et al., 2021; Zhang, Li, He, Yang and Zhang, 2021b) for indoor scene synthesis follow an autoregressive manner, where the generation process unfolds sequentially and each furniture item is generated conditioned on the previously generated. Wang (Wang et al., 2018) learn deep convolutional priors and synthesize indoor scenes through the successive placement of new furniture items based on an autoregressive model of object position priors. In one of the follow-ups, Ritchie (Ritchie et al., 2019) further improve the deep convolutional generative model, which enables faster and more flexible indoor scene synthesis in a similar pipeline. Sceneformer (Wang et al., 2021) abandons CNN and instead leverages the Transformer to implicitly learn object relations, which accomplishes indoor scene generation by generating a sequential series of objects along with their positions and

orientations. Similarly utilizing the Transformer, ATISS (Paschalidou et al., 2021) employs an autoregressive model for synthesizing indoor scenes, with the key distinction of adopting an unordered generation sequence. Progressive indoor scene synthesis often struggles with capturing global dependencies and incorporating high-level functionality efficiently. In contrast, our function-centric framework enables a more holistic and efficient synthesis span from a coarse level to a fine-grained.

Our method draws inspiration from ATISS but diverges significantly. On one hand, designers typically prioritize the functionality of specific areas before arranging furniture in interior design. Our method adheres to this guideline, whereas ATISS generates indoor scenes based on individual furniture items, resulting in lower efficiency. This is confirmed by the quantitative evaluation later in the paper, where the average time for our method to generate an indoor scene is 28.61% faster than ATISS. On the other hand, our method can be divided into several modules. This allows for multi-level control in indoor scene generation, catering to personalized generation needs, whereas ATISS directly iterates predictions for every furniture item, which makes it difficult to achieve high-level generation control. We show more comparison results in the evaluation. Furthermore, our method combines VAE and Transformer, further enhancing the diversity of generated results compared to ATISS, while the input of the attribute extractor in ATISS is a constant when generating the indoor scene.

2.2. Graph-based scene synthesis

Using graphs to represent scenes is straightforward, which has also given rise to a great line of graph-based indoor scene synthesis (Fisher, Savva and Hanrahan, 2011; Fisher, Ritchie, Savva, Funkhouser and Hanrahan, 2012; Yeh, Yang, Watson, Goodman and Hanrahan, 2012; Liu, Chaudhuri, Kim, Huang, Mitra and Funkhouser, 2014; Fu et al., 2017; Yan, Chen and Zhou, 2017; Liang, Xu, Zhang, Lai and Mu, 2018; Wang, Lin, Weissmann, Savva, Chang and Ritchie, 2019; Zhou et al., 2019). PlanIT (Wang et al., 2019) proposes a plan-and-instantiate framework for indoor scene synthesis, which involves generating a relation graph to plan the scene and then instantiating a specific scene that adheres to the plan. Our method can also be categorized as plan-and-instantiate, except that we generate function groups directly as plans of the indoor scenes. Also adopting the message-passing scheme in graph learning, SceneGraphNet (Zhou et al., 2019) uses a graph neural network to capture the relations between furniture items and predicting a probability distribution of object types on the query location. Both methods employ a progressive generation methodology. Another mainstream is to combine graph-based representation and VAE-based learning to construct generative models for indoor scene synthesis (Li et al., 2019; Zhang et al., 2020; Luo et al., 2020; Purkait, Zach and Reid, 2020). For instance, Luo (Luo et al., 2020) introduce a variational generative model for indoor scene synthesis, using relation graphs to provide guidance. However, all these methods require users to define object relationships or hierarchies within the scene beforehand. Recently, Chattopadhyay (Chattopadhyay, Zhang, Wipf, Arora and Vidal, 2023) propose a graph variational autoencoder with a structured prior for generating indoor scenes. They claim no user-defined relations, but a check of association between furniture items and room elements is needed. In contrast, we built a hierarchy for indoor scenes around functionality, without the need to provide any relations of scene objects. On the other hand, graph-based indoor scene synthesis also faces challenges in generating fine-grained details and capturing local relations.

2.3. Function-driven scene synthesis

Functionality plays a key role in scene synthesis (Savva, Chang, Hanrahan, Fisher and Nießner, 2016; Ma, Li, Zou, Liao, Tong and Zhang, 2016; Fu et al., 2017; Qi, Zhu, Huang, Jiang and Zhu, 2018; Fu, Fu, Yan, Zhou, Chen and Li, 2020; Zhang, Xie and Zhang, 2021c; Zhang et al., 2021a). Synthesizing functional scenes aims to generate plausible layouts while also emphasizing the practicality of generated scenes. In terms of function-driven scene synthesis, the most related work to ours is (Zhang et al., 2021c), which proposes a data-driven layout generation framework that implements and organizes the prior based on samples of the dataset, rather than sampling probability distributions. This approach can capture the hyper-relations among scene objects, which is applied in our data processing. Given furniture objects, Zhang (Zhang et al., 2021a) learns the priors of indoor scenes by co-occurrence analysis and statistical model fitting. Then, the input furniture objects are divided into function groups for layout optimization. However, since geometric arranging is non-learning, they tend to produce overly rigid layouts without rich structural variation. Hyper-relations or superstructures are also addressed in PlanIT (Wang et al., 2019), where two types of superstructures, hub-and-spoke, and chain, are studied. The functionality of the indoor scene serves human activities, and in turn, human activity affects the functionality of the indoor scene, which directly affects the scene layout. There has been a great deal of work on utilizing human-centric or activity-centric approaches for indoor scene synthesis (Savva et al.,

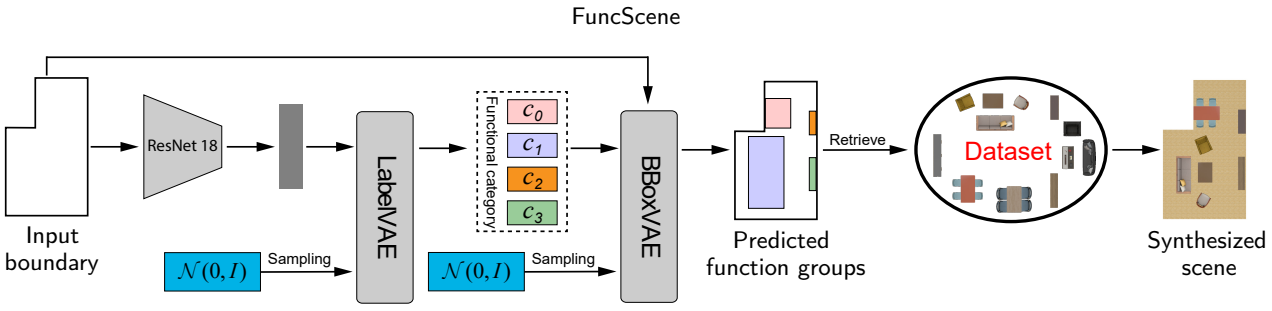


Figure 2: Overview of FuncScene, which consists of three main parts. We first acquire the number and categories of function groups of the scene through LabelVAE which predicts an ordered sequence of functional categories, and then BBoxVAE iteratively generates all necessary attributes of each function group based on the predicted functional category sequence. Finally, we instantiate the generated function groups by searching and matching them with corresponding instances from our indoor scene dataset.

2016; Ma et al., 2016; Fu et al., 2017; Qi et al., 2018; Fu et al., 2020). However, these methods value the functionality of synthesized scenes while overlooking the generation quality. In contrast, our method achieves a better balance.

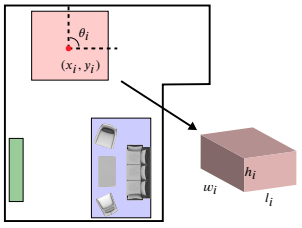
3. Overview

Figure 2 shows the pipeline of our framework. In the following, we first introduce the indoor scene representation as well as our data preparation, and then the key problem and primary challenges faced in this work are discussed. Finally, our indoor scene synthesis methodology is presented.

3.1. Indoor scene dataset

We introduce our indoor scene dataset, including the layout representation of indoor scenes and the dataset construction.

Layout representation. Most previous work (Wang et al., 2018; Ritchie et al., 2019; Wang et al., 2019) on indoor scene synthesis converts 3D scenes into a 2D layout of the top-down view. We adopt a similar representation, as shown in the right inset. Specifically, each indoor scene $\mathcal{S}$ is composed of its room boundary $\mathcal{B}$ and a set of function groups $\mathcal{G} = \{g_1, g_2, ..., g_n\}$. We represent the room boundary as a grayscale map with dimensions of 128×128, where the pixel values within the interior region of the room are set to 1, while the exterior region is set to 0. Every function group g_i is represented as its enclosed bounding box containing four parameters: (i) functional category c_i, (ii) central location (x_i, y_i), (iii) bounding box size (l_i, w_i, h_i), and (iv) front-facing orientation $\theta_i \in \{0, 1\}^8$. Note that we adopt the 2D centre representation and 3D size representation, which is different from previous work with pure 2D layout representation. For orientation, we assume each furniture category has a canonical front-facing direction. The direction of most indoor objects aligns with one of the four cardinal directions. A few objects may exhibit diagonal orientations. Consequently, we have abstracted the orientation attribute into 8 fundamental directions (*east*, *south*, *west*, *north*, *northeast*, *southeast*, *northwest*, and *southwest*), employing a one-hot coding representation. Each function group also contains a set of furniture items $g_i = \{o^i_1, o^i_2, ..., o^i_m\}$, and we apply the same representation as the function group.



Dataset construction. We employ 3D-FRONT (Fu, Cai, Gao, Zhang, Wang, Li, Zeng, Sun, Jia, Zhao et al., 2021a) to construct the indoor scene dataset, while the furniture is populated using 3D-FUTURE (Fu, Jia, Gao, Gong, Zhao, Maybank and Tao, 2021b). Since 3D-FRONT and 3D-FUTURE lack annotations about function groups, we extract function groups within indoor scenes based on the hyper-relations proposed in (Zhang et al., 2021c). Taking the living room as an example, all furniture items are categorized into five function groups: *DiningGroup*, *ReceptionGroup*, *EntertainmentGroup*, *StorageGroup*, and *TableGroup*. The details can be found in Table 1. Within each function group, furniture items are divided into primary and secondary objects. Primary objects play a crucial role in fulfilling the group's functionality, while secondary objects are adjunct to the primary ones. Typically, each function group has only one primary object, but it can comprise multiple secondary objects. For instance, in *ReceptionGroup*, the primary object is the coffee table, whereas the secondary objects are the sofa and chairs. To extract function groups,

Table 1: Function groups in the living room. The primary objects are shown in bold.

DiningGroup	**DiningTable**, DiningChair
ReceptionGroup	**CoffeeTable**, Sofa, Chair
EntertainmentGroup	**TVstand**
StorageGroup	**Cabinet**
TableGroup	**CornerTable**

we first traverse through each furniture item within the scene. Utilizing predefined hyper-relations, we make an initial determination of the function group partitioning. The primary objects are evident in this initial partitioning stage. Then, we compute the Euclidean distance between the primary object and potential secondary objects. If this distance falls below a specified threshold, the secondary object is assigned to the respective function group. Finally, the axis-aligned bounding box of the extracted function group is calculated. We utilize the central location and 3D sizes of the bounding box to depict the function group's location and size. The front-facing direction of the primary object is used to indicate the function group's orientation. We also filter out unreasonable samples, including cases where function groups collided with each other or where function groups extended beyond room boundaries. In the end, we have acquired a dataset that includes function groups and furniture items, consisting of 1361 living rooms and 1212 bedrooms.

3.2. Problem statement

Our goal is to establish an indoor scene synthesis framework capable of producing diverse indoor scenes that adhere to the given room boundaries. In this work, synthesizing indoor scenes is equivalent to generating layouts of indoor scenes. Our framework takes a room boundary $\mathcal{B}$ as input and generates all furniture items to obtain the indoor scene $\mathcal{S} = \{o_1, o_2, .., o_N \mid \mathcal{B}\}$.

To create practical indoor environments, one of the key challenges lies in maintaining the functionality of the synthesized indoor scenes. Scene objects that carry functional significance are inherently interconnected, with occurrences or disappearances being interdependent. However, modeling these intricate relationships among scene objects proves to be a complex task, requiring the application of collaborative analysis. Furthermore, while generating indoor scenes, we aspire to introduce a certain degree of diversity. Striking the right balance between diversity and practicality is also a challenge.

3.3. Methodology

To preserve the functionality of synthesized indoor scenes, we use function groups as an intermedium to achieve a coarse-to-fine indoor scene synthesis, represented as $\mathcal{S} = \{g_1, g_2, .., g_n \mid \mathcal{B}\} = \{o_1, o_2, .., o_N \mid \mathcal{B}\}$. To this end, we propose a function-centric indoor scene synthesis framework FuncScene. As shown in Figure 2, FuncScene mainly consists of three phases. Initially, FuncScene predicts a sequential arrangement of functional categories, effectively determining both the number and categories of function groups present in the indoor scene. Then, the detailed attributes (such as central locations, bounding box sizes, and front-facing orientations) of each function group are progressively generated, leveraging the predicted functional category sequence to obtain an overview of the scene layout. For the initial phases, we employ Transformer-based generative models. To enhance the diversity of synthesized indoor scenes and facilitate user exploration, we incorporate a VAE into these generative models to obtain LabelVAE and BBoxVAE, respectively. Lastly, utilizing function groups as the reference, we instantiate the generated function groups by systematically searching and matching them with corresponding instances from our indoor scene dataset. Through this approach, the individual furniture elements within each function group are acquired, effectively capturing the complex details in the local scene.

Previous work either generates all scene objects at once (Zhou et al., 2019) or one-by-one (Paschalidou et al., 2021). Performing effectively and flexibly is not easy. Our generative model consists of two models, LabelVAE and BBoxVAE, for semantics predication and attribute generation, making it easier to achieve multi-level generation control for indoor scene synthesis, which is challenging for previous methods.

4. Method

To preserve the functionality of synthesized indoor scenes, we propose a function-centric indoor scene synthesis framework. Given a room boundary as input, we achieve indoor scene synthesis by first generating function groups in

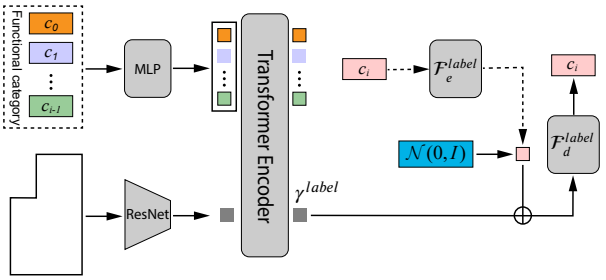


Figure 3: Network Architecture of LabelVAE. First, we extract features from the room boundary and existing functional categories, respectively. The extracted features are then input into the Transformer encoder to obtain the conditional feature γ^{label}. During the training phase (dotted line), we map the new functional category of ground truth to a latent space by $\mathcal{F}_e^{label}$. The latent variable sampled from this space and the corresponding conditional feature are then fed into $\mathcal{F}_d^{label}$ to predict a new functional category. In the testing phase, we directly sample from the standard normal distribution.

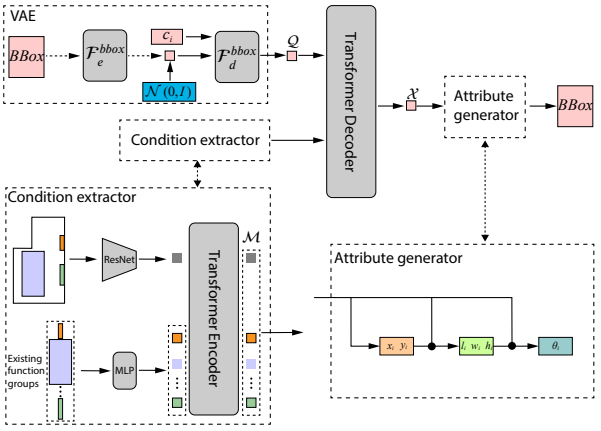


Figure 4: Network Architecture of BBoxVAE. We first extract features from the room boundary and existing function groups. The extracted features are then input into the Transformer encoder to obtain a relation matrix $\mathcal{M}$. During the training phase (dotted line), we map the new bounding box of ground truth to a latent space by $\mathcal{F}_e^{bbox}$. The latent variable sampled from this space and the corresponding functional category are then fed into $\mathcal{F}_d^{bbox}$ to obtain the query vector $\mathcal{Q}$. We combine $\mathcal{Q}$ with $\mathcal{M}$ to the Transformer decoder to obtain the attribute feature $\mathcal{X}$. Finally, we autoregressively predict the attributes of the new function group according to $\mathcal{X}$. In the testing phase, we directly sample from the standard normal distribution.

the scene and then instantiating by searching and matching them with corresponding instances from our indoor scene dataset. To produce function groups, we first acquire the number and categories of function groups in the scene through LabelVAE which predicts an ordered sequence of functional categories, and then BBoxVAE iteratively generates all necessary attributes of each function group based on the predicted functional category sequence.

4.1. Functional category prediction

Given a room boundary $\mathcal{B}$, the indoor scene might not be unique. In other words, there could be multiple possible functional arrangements. Thus, we aim to learn the distribution of possible function groups. We propose a functional category prediction model LabelVAE to acquire an ordered sequence of functional categories in an autoregressive manner, obtaining both the number and categories of function groups. Given the room boundary $\mathcal{B}$ and existing functional categories $c_{<i} = \{c_0, c_1, ..., c_{i-1}\}$, LabelVAE learns the category distribution of i-th function group $P(c_i \mid \mathcal{B}, c_{<i})$. As shown in Figure 3, the key idea of LabelVAE is to map the new functional category to a latent space using an encoder $\mathcal{F}_e^{label}$, and then mapping a latent variable z back to reconstruct c_i using a decoder $\mathcal{F}_d^{label}$. We employ the condition extractor to obtain the conditional feature for VAE from the room boundary and existing functional categories.

Condition extractor. The input to LabelVAE is $\mathcal{B}$ and $c_{<i}$, where $\mathcal{B}$ is represented by a one-channel grayscale map, and $c_{<i}$ is represented by one-hot coding. We use ResNet18 (He, Zhang, Ren and Sun, 2016) to output an embedded feature of $\mathcal{B}$ and implement feature extraction from $c_{<i}$ using Multilayer Perceptron (MLP). The shape of the $c_{<i}$ is (B, L, N), where B refers to the batch size of the training ($B = 1$ when testing), L refers to the maximum number of function groups in one indoor scene ($L = 7$ in our experiments), and N is the number of functional categories ($N = 5$ in our experiments). If the number of function groups in one indoor scene is less than L, zeros are appended to make up for the difference. To enable the MLP to extract features from $c_{<i}$, we first transform the shape of $c_{<i}$ to $(B \times L, N)$, and then encode it into an embedding c_{embed} with a shape of $(B \times L, 128)$ through the MLP, and finally restore the shape of c_{embed} to $(B, L, 128)$. Then, two features are concatenated to the Transformer enocer (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser and Polosukhin, 2017) to obtain the conditional feature γ^{label}, serving as a condition

to LabelVAE.

VAE. The input to the VAE encoder $\mathcal{F}_e^{label}$ is the new functional category c_i. Then, we obtian $\{\mu^{label}, \Sigma^{label}\} = \mathcal{F}_e^{label}(c_i)$, where $\mu^{label} \in \mathbb{R}^{64}$ and $\Sigma^{label} \in \mathbb{R}^{64}$ are the mean and convariance of a Gaussian distribution $\mathcal{N}(\mu^{label}, \Sigma^{label})$. A latent variable $z \in \mathbb{R}^{64}$ is sampled from $\mathcal{N}(\mu^{label}, \Sigma^{label})$ using the reparameterization technique (Kingma and Welling, 2014). The input to the VAE decoder $\mathcal{F}_d^{label}$ is z and γ^{label}, and we reconstruct c_i by $\widetilde{c_i} = \mathcal{F}_d^{label}(z, \gamma^{label})$. We implement $\mathcal{F}_e^{label}$ and $\mathcal{F}_d^{label}$using MLP.

The total loss to train LabelVAE is:

$$\mathcal{L}^{label} = \mathcal{L}_{rec}^{label} + \alpha \mathcal{D}_{KL}^{label}(\mathcal{N}(\mu^{label}, \Sigma^{label}) \| \mathcal{N}(\mathbf{0}, \boldsymbol{I})) \tag{1}$$

$\mathcal{D}_{KL}^{label}$ denotes the Kullback-Leibler (KL) divergence, and $\alpha = 0.3$ in our implementation. We employ the Cross-Entropy Loss to $\mathcal{L}_{rec}^{label}$.

During training, we choose a scene from the dataset and apply a random permutation to its function groups. Then, we randomly select the first $i-1$ function groups and use their functional categories and room boundary to compute the conditional feature γ^{label}. Conditioned on γ^{label}, LabelVAE predicts the i-th functional categories. To determine when to stop predictions, we introduce start and end symbols to indicate the start and end of prediction. Upon generating one functional category, we append it to the sequence of functional categories. This process persists until the predicted functional category matches an end symbol. During inference, we sample z from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and predict $\widetilde{c_i}$ by $\widetilde{c_i} = \mathcal{F}_d^{label}(z, \gamma^{label})$.

4.2. Function group generation

Using LabelVAE, we can obtain an ordered sequence of functional categories. Our subsequent goal is to generate detailed attributes for each function group based on the predicted functional category sequence. Similar to the functional category prediction, when given a room boundary $\mathcal{B}$ and a functional category sequence $\{c_0, c_1, ..., c_n\}$, the attributes of function groups may not be uniquely determined. To this end, we propose a functional attribute generative model BBoxVAE, which progressively obtains detailed attributes for each function group in accordance with the functional category sequence. BBoxVAE learns the distribution of attributes for the i-th function group, represented as $P(x_i, y_i, l_i, w_i, h_i, \theta_i \mid \mathcal{B}, c_i, g_{<i})$, where $g_{<i} = \{g_0, g_1, ..., g_{i-1}\}$ denotes the function groups already present in the indoor scene.

As shown in Figure 4, BBoxVAE maps the new bounding box of ground truth (the attributes of the new function group) to a latent space using an encoder $\mathcal{F}_e^{bbox}$. Then, we adopt a decoder $\mathcal{F}_d^{bbox}$ to map a latent variable z sampled from this latent space as well as the corresponding functional category to obtain a query vector $\mathcal{Q}$. We employ a condition extractor to obtain a relation matrix $\mathcal{M}$ from the room boundary and existing function groups. Note that $\mathcal{M}$ is a matrix that contains the feature of the scene boundary and the features of the function groups already present in the scene, with the shape $(B, L+1, 64\times 7)$, where B is the batch size of the training ($B = 1$ when testing), $L+1$ is the maximum number of function groups already present in the scene plus the room boundary ($L = 7$ in our experiments), with each has a dimension of 64×7, as we encode each parameter in the attributes of the function groups (x, y, l, w, h, θ) and the functional category c as a 64-dimensional vector. $\mathcal{Q}$ and $\mathcal{M}$ are used to obtain the attribute feature $\mathcal{X}$. Finally, we employ an attribute generator $\mathcal{F}_g^{bbox}$ to autoregressively produce attributes of the next function group to be added to the scene.

Condition extractor. The input to the condition extractor is the given room boundary $\mathcal{B}$, and existing function groups $g_{<i}$. We employ two types of representations to depict the current indoor scene: the semantic representation and the parameter representation. The semantic representation is a multi-channel image that contains both the spatial layout of the indoor scene and existing function groups. The first channel corresponds to the grayscale image of the interior region of the indoor scene. Subsequent channels represent grayscale images of top-down views of bounding boxes of different function groups. The last channel combines all previous channels, utilizing distinct pixel values to represent different function groups. We extract the semantic feature from the semantic representation using ResNet18. For the parameter representation, we first obtain the specific attributes of each function group, including location, size, and orientation. Each attribute is encoded into a 64-dimensional embedding with an MLP. These features are then concatenated to create the attribute embedding for the corresponding function group. The semantic feature and the attribute embedding are concatenated to obtain the conditional feature, which is encoded to obtain a relation matrix $\mathcal{M}$ through the Transformer encoder.

VAE. The input to the VAE encoder $\mathcal{F}_e^{bbox}$ is the bounding box of the next function group g_i, represented by location (x_i, y_i), size (l_i, w_i, h_i), and orientation θ_i. Then, we obtian $\{\mu^{bbox}, \Sigma^{bbox}\} = \mathcal{F}_e^{bbox}(x_i, y_i, l_i, w_i, h_i, \theta_i)$, where $\mu^{bbox}, \Sigma^{bbox} \in \mathbb{R}^{64}$ are the mean and convariance of a Gaussian distribution $\mathcal{N}(\mu^{bbox}, \Sigma^{bbox})$. A latent variable $z \in \mathbb{R}^{64}$ is sampled from $\mathcal{N}(\mu^{bbox}, \Sigma^{bbox})$. z is concatenated with the one-hot coding of the corresponding functional category c_i, decoding into a query vector $Q \in \mathbb{R}^{64}$ by $\mathcal{F}_d^{bbox}$. We implement $\mathcal{F}_e^{bbox}$ and $\mathcal{F}_d^{bbox}$ using MLP. The Transformer decoder receives both the query vector Q and the relation matrix $\mathcal{M}$ from the Transformer encoder to obtain the attribute feature $\mathcal{X}$ of the next function group. We follow (Vaswani et al., 2017) to implement a multi-head attention Transformer, and we do not use positional encoding since the object's attributes contain positional information.

Generator. We employ an attribute generator to autoregressively produce various attributes of the next function group to be added in the scene with the attribute feature $\mathcal{X}$.

To enable autoregressive prediction of attributes, we condition the prediction of a particular attribute on the attributes that have been previously predicted. This process resembles ATISS (Paschalidou et al., 2021). We first predict the positional attribute of the object from the attribute feature $\mathcal{X}$. Then we extract the features of the positional attribute and splice them with $\mathcal{X}$ as a way of predicting the size attribute, and finally, we splice the positional feature, the size feature, and $\mathcal{X}$ as a way of predicting the angle attribute. During training, we use the ground truth attributes of the i-th function group for embedding.

$$X \longrightarrow (x_i, y_i) \tag{2}$$

$$(X, F(x_i, y_i)) \longrightarrow (l_i, w_i, h_i) \tag{3}$$

$$(X, F(x_i, y_i), F(l_i, w_i, h_i)) \longrightarrow \theta_i \tag{4}$$

The total loss to train BBoxVAE is:

$$\mathcal{L}^{bbox} = \mathcal{L}_l^{bbox} + \mathcal{L}_s^{bbox} + \mathcal{L}_o^{bbox} + \beta \mathcal{D}_{KL}^{bbox} \tag{5}$$

Where $\mathcal{L}_l^{bbox}$, $\mathcal{L}_s^{bbox}$, and $\mathcal{L}_o^{bbox}$ calculate the reconstruction loss respect to location, size, and orientation. $\mathcal{L}_l^{bbox}$ and $\mathcal{L}_s^{bbox}$ are computed using $\mathcal{L}2$ Loss. For $\mathcal{L}_o^{bbox}$, we employ the Cross-Entropy Loss to calculate the probabilities of orientation. Meanwhile, $\mathcal{D}_{KL}^{bbox}$ is utilized to measure the distance between the distribution of function group attributes and the standard normal distribution. We set $\beta = 0.1$ for the loss balance. During training, to obtain training samples, we take rooms from our dataset and apply a random permutation to its function groups. Then, we randomly select the first $i-1$ function groups as the current scene to obtain the attribute feature $\mathcal{X}$. Conditioned on $\mathcal{X}$, BboxVAE generates attributes to obtain the i-th function group in the scene. The latent variable z is sampled from the learned Gaussian distribution $\mathcal{N}(\mu^{bbox}, \Sigma^{bbox})$.

During inference, for a new room boundary and predicted functional categories, BBoxVAE iteratively generates the function groups based on the generated function groups in the scene until an end symbol of the functional category is given. We sample z from the standard normal distribution.

4.3. Function group instantiation

With LabelVAE and BBoxVAE, we can obtain a sequence of function groups. Our subsequent goal is to find the corresponding 3D models of function groups based on the predicted attributes of function groups. In the field of indoor scene synthesis, the instantiation of indoor scenes typically relies on retrieving furniture from a furniture library based on predicted furniture attributes. In most of the previous research (Wang et al., 2019; Paschalidou et al., 2021), furniture items are added individually to the indoor scene. However, even minor variations in the predicted furniture attributes could lead to inconsistencies in the overall furniture style within the scene. Furthermore, deep neural network models often struggle to learn lower-level geometric constraints, such as alignment and symmetry, to prevent collisions between furniture models.

To address these aforementioned issues, our method involves directly instantiating function groups from the indoor scene dataset rather than individual items of furniture. This approach offers several advantages: (i) Inherent functional coherence: Our instantiation process is naturally functional, as it operates on function groups as the fundamental retrieval units. Therefore, there won't be instances where necessary furniture is missing within the retrieved function groups. (ii) Highly consistent furniture styles: In indoor scenes, the consistency of furniture styles is typically observed within specific functional areas. Our method aligns with this by considering function groups as representations of

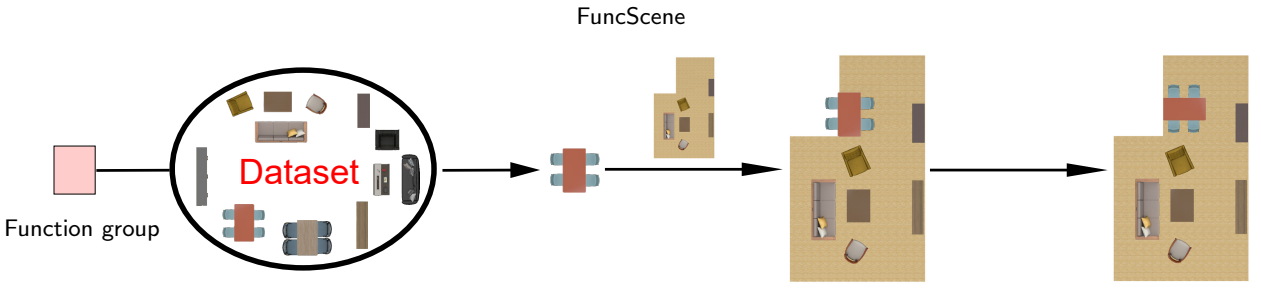


Figure 5: Function group instantiation. Given a predicted function group, we start by retrieving a list of 3D models of function groups from the database based on the predicted functional category. Then, we calculate the error in size between the given function group and function groups of the list and select the function group with the smallest error as the one to instantiate. Finally, we translate the selected function group to the predicted location and rotate the entire function group according to our predicted orientation.

such areas, thus ensuring a high degree of consistency in furniture styles. (iii) Avoiding learning low-level geometric constraints: Using function groups as the fundamental retrieval units guarantees that geometric issues like alignment and symmetry are avoided within functional groups.

Our instantiation method is simple, and straightforward, but very effective, as shown in Figure 5. Given a predicted function group, we start by retrieving a list of 3D models of function groups from the database based on the predicted functional category. Then, considering the predicted size, we calculate the error in size between the given function group and function groups in the list:

$$d_i = (l_{pred} - l_i)^2 + (w_{pred} - w_i)^2 + (h_{pred} - h_i)^2 \tag{6}$$

We select the function group with the smallest error as the one to instantiate. Finally, we translate the selected function group to the predicted location and rotate the entire function group according to our predicted orientation to obtain the final instantiation result.

5. Experiment and evaluation

To validate the efficacy of FuncScene, we have conducted a series of ablation studies and performance comparisons with state-of-the-art techniques. We have implemented FuncScene with PyTorch and used Adam optimizer (Kingma and Ba, 2015) for training. All models are trained and tested on an NVIDIA GeForce RTX 2080Ti. The training data of the function group is constructed from 3D-Front (Fu et al., 2021a), with 100 indoor scenes used for validating and testing, and the rest for training. In our experiments, we implement a training strategy aimed at ensuring effective learning and generalization of the model. Specifically, we adopt a relatively low but constant learning rate of 0.0001. Additionally, we group 64 indoor scenes into one batch for training. Through observation, we note that the model typically converges around the 900th iteration out of approximately 1000 iterations, with the entire training process taking about 8 hours. Considering the limited amount of training data available, we introduce adjustments to the arrangement order of objects within each indoor scene during the training process. This step aims to further enhance the model's generalization capability. By modeling the diversity and variability in realistic indoor scenes, we aim to strengthen the model's ability to generalize beyond the training data.

5.1. Ablation study

When generating multiple attributes of objects, one approach is similar to ours of generating attributes progressively, where previously generated attributes serve as the constraint for generating subsequent ones. Another approach involves generating all attributes of objects simultaneously. To verify the effectiveness of our step-by-step attribute generator, we have conducted an ablation study by generating all attributes at once, as shown in Figure 6(a). Due to the mutual interference between different attributes, the generative model struggles to learn the relative relationships between function groups. This limitation manifests in generated indoor scenes, where certain function group regions might overlap, as shown in the second and third columns of Figure 6(a).

Moreover, the order in which function groups are generated also affects the indoor scene synthesis. In previous work (Fu et al., 2021b), a fixed order is typically used for training and synthesizing indoor scenes. However, we

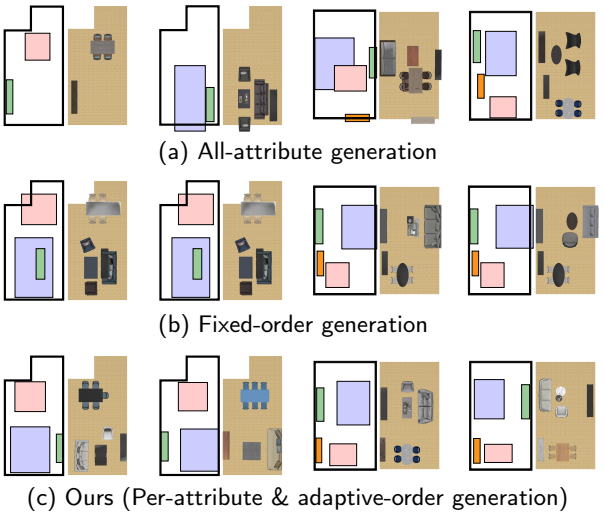


Figure 6: Ablation Study. To evaluate the effectiveness of our autoregressive per-attribute prediction, we have conducted an ablation study to simultaneously generate all attributes, as shown in (a). To evaluate the effectiveness of our indoor scene synthesis in adaptive order, we have arranged the function groups within each scene in descending order of frequency and then made the model generate indoor scenes following this fixed order, as shown in (b). The results of our method are shown in (c).

Figure 7: Diversity evaluation. We conduct the diversity evaluation of our method and baseline methods. From top to bottom, we show the generated indoor scenes by ATISS, the generated scenes by LayoutVAE, and the generated scenes by our method.

employ a non-fixed generation order and let the model determine the generation order. We think an adaptive order of indoor scene synthesis enhances training sample diversity and the model's capability of generalization. We have conducted an ablation study concerning the generation order. Specifically, we fix the generation order based on the frequency of function group occurrences within the indoor scenes, with more frequent groups generated earlier, as shown in Figure 6(b). However, due to the small size of our dataset, using an approach that predicts object attributes in a fixed order can limit the model's ability to generalize, as shown in Figure 6(b), which generates essentially identical indoor scenes with the same boundary inputs. In contrast, our method strikes a good balance between the diversity and plausibility of the generated results, as shown in Figure 6(c).

Our model differs slightly from the standard Conditional Variational Autoencoder (CVAE) architecture. Typically, both the encoder and decoder of CVAE receive conditions as input, whereas in our model, conditions are solely utilized in the decoder. There are two main considerations. Firstly, by integrating the Transformer module into our model's decoder, we can effectively understand conditional constraints and capture dependencies between objects in the scene. This ensures that our model does not overlook these constraints, preventing it from simply reproducing output directly from input. Consequently, there's no need to incorporate these constraints into the encoder, which simplifies the model. Secondly, as the objects in the scene are dynamically changing, our conditional constraints become more complex. Including these constraints in the encoder could increase the complexity of the latent space, potentially complicating the sampling process. In contrast, our current design maintains a simpler latent space, which facilitates a more straightforward sampling process. To validate our design, we perform a quantitative evaluation of ablation studies. Considering that conditions in the function group generation can be categorized into functional categories and constraints processed by the constraint extractor, we design two ablation studies: CVAE_1 (only including the constraints of functional category in the encoder of VAE) and CVAE_2 (including all constraints in the encoder of VAE), with experimental results depicted in Table 3.

5.2. Qualitative evalution

We use ATISS (Paschalidou et al., 2021) and LayoutVAE (Jyothi, Durand, He, Sigal and Mori, 2019) to compare against our method. Both methods perform layout generation by predicting the parameters of the object's bounding

Figure 8: Plausibility evaluation. We conduct the plausibility evaluation of our method and baseline methods. From top to bottom, we show the indoor scenes of ground truth, the generated indoor scenes by ATISS, the generated scenes by LayoutVAE, and the generated scenes by our method.

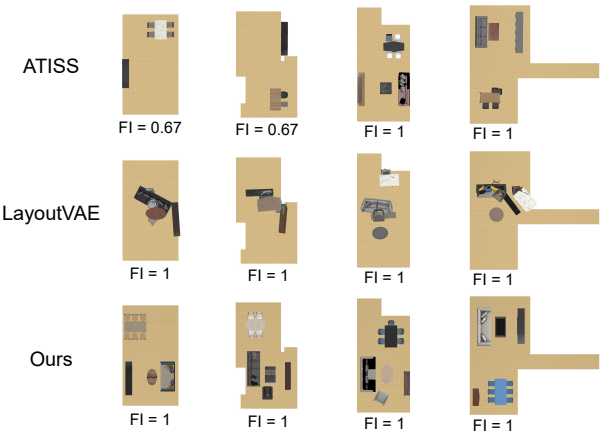


Figure 9: Some results of quantitative evaluation. From top to bottom, it's the generated scenes by ATISS, LayoutVAE, and our method. Our generated results exhibit better functional integrity than ATISS and LayoutVAE.

box. Our dataset is used in ATISS and LayoutVAE for training and testing. ATISS aims to synthesize indoor scenes progressively, which uses an autoregressive Transformer to turn the task of scene generation into an unordered generation of furniture items. LayoutVAE uses VAE to realize a generalized layout generation and divides the model into count prediction and object prediction. LayoutVAE is initially intended for planar layout generation, without boundary constraints. We enhance it by incorporating boundary constraints and utilizing RseNet18 for boundary encoding. Additionally, a network for predicting orientation is added.

We qualitatively evaluate our method and other methods in terms of diversity and plausibility. To evaluate diversity, we maintain fixed boundary constraints, as illustrated in Figure 7. To evaluate plausibility, we employ different boundary constraints, as presented in Figure 8.

In Figure 7, the indoor scenes in the third row are generated by LayoutVAE under the same boundary constraints. It can be observed that there is some variation in the scene layout, indicating a certain level of diversity of generation. However, since scenes are represented using parameters, the capability of LayoutVAE to handle spatial attributes is somewhat limited. Additionally, LayoutVAE does not model the relationships between furniture items well, resulting in a higher frequency of object collisions, which affects the overall plausibility of the generated scenes. In the third row of Figure 8, the overlaps between furniture items are evident. Furthermore, LayoutVAE generates scenes in a fixed order, which limits the generalization capability in the case of a relatively small training dataset. However, introducing VAE in LayoutVAE can improve diversity to some extent.

As for ATISS, the Transformer-based architecture is effective in capturing relationships between objects, as shown in the second row of Figure 8. However, because the query vector is a learnable parameter, during the prediction process, if other inputs remain fixed, the model's output is likely to be the same, resulting in limited diversity, as shown in the first row of Figure 7, where indoor scenes generated under the same boundary constraints are almost indistinguishable. On the other hand, furniture items are added separately to the indoor scenes. The functionality of indoor scenes is not considered in ATISS. As shown in the first row of Figure 8, the styles of dining chairs around the dining table of the first column are different from other furniture, and in the third column, there is no sofa and coffee table opposite the TV, which leads to a lack of functionality of the TV.

As depicted in the third row of Figure 7, our method generates indoor scenes with significant variation under the same boundary constraints, with no overlap between furniture items and higher diversity. As shown in the fourth row

of Figure 8, the objects in the indoor scenes generated by our method are mostly within the boundaries with fewer collisions between furniture items, resulting in higher plausibility. In contrast, our method achieves a good balance between result diversity and plausibility.

5.3. Quantitative evalution

We conduct quantitative evaluations based on several widely used metrics for generation tasks: FID (**F**réchet **I**nception **D**istance), which is used in (Heusel, Ramsauer, Unterthiner, Nessler and Hochreiter, 2017; Paschalidou et al., 2021; Koh, Agrawal, Batra, Tucker, Waters, Lee, Yang, Baldridge and Anderson, 2023); CA (**C**lassification **A**ccuracy), which is used in (Paschalidou et al., 2021; Tang, Nie, Markhasin, Dai, Thies and Nießner, 2024); TIME (Generation **TIME** per indoor scene), which is used in (Ritchie et al., 2019; Wang et al., 2019, 2021). Furthermore, we introduce a new metric, FI (Functional Integrity), to evaluate the functionality of generated scenes. We calculate these metrics on the testing dataset of the living room. The quantitative results can be found in Table 2. Some examples of quantitative evaluations are shown in Figure 9.

- FID: It is primarily used to calculate the distance between two distributions. We employ the inception network (Xia, Xu and Nan, 2017) to extract features from 2D floorplan images (i.e., top-down view of the 3D indoor scene). This is widely used in FID calculation of the indoor scene, such as ATISS (Paschalidou et al., 2021). Typically, it measures the distance between the feature vectors of generated image data and real image data. A lower FID score indicates higher similarity between the sets of images, implying that the generated results are closer to the real.

- CA: We train a classifier to determine whether input images belong to the generated scenes or the real scenes and calculate scene classification accuracy based on this. A score close to 0.5 suggests that the classifier cannot distinguish between generated scenes and real scenes, indicating a nice generation.

- TIME: It calculates the average time in seconds required to generate an indoor scene. A smaller time indicates that the method is more real-time and more efficient in generating indoor scenes.

- FI: To assess the functionality of generated scenes, we introduce the functional integrity measure FI. Observing the presence of DiningGroup, ReceptionGroup, and EntertainmentGroup in the majority of living rooms, we define FI as the presence of these three function groups in indoor scenes. We say that a scene contains a certain function group when the scene contains all the primary objects and at least one secondary primary object of that function group, then adding 1 to FI. Finally, FI is normalized to 0-1 to obtain the final score. A score closer to 1 indicates better functional integrity.

Limited by the size of our dataset and the fix-order attribute prediction in LayoutVAE, it exhibits weaker generalization and poorer plausibility during generation, resulting in the lowest FID and CA. Moreover, LayoutVAE takes more time to generate the indoor scene, which is less efficient, resulting in the lowest TIME metrics. Due to the fix-order prediction in LayoutVAE, it performs well in terms of furniture category and quantity prediction, making it slightly poorer than our method in FI.

Both ATISS and our method predict attributes in a non-fixed order, which improves the generalization of the model. However, our method uses function groups as the basic units, improving the plausibility of the generated results. We also introduce VAE to enhance the synthesis diversity. As a result, our approach is higher than ATISS in terms of FI and CA metrics. Since our method directly instantiate function groups from the indoor scene dataset rather than individual prediction in both ATISS and LayoutVAE, the cost of time for generating each indoor scene of our method is lower, reflected in better TIME metrics. Furthermore, our generated results exhibit better functionality and integrity than ATISS and LayoutVAE. As shown in Figure 9, we record the FI metrics of each generated scene. ATISS uses furniture as the basic generation object and does not consider the functionalities to be implemented in the indoor scene, obtaining the lowest FI, whereas our method uses function groups as the basic generation object, so our FI is highest with better functional integrity.

We also perform a quantitative evaluation of our method, CVAE_1 and CVAE_2, using four metrics: FID, CA, TIME, and FI. The results are shown in Table 3. It can be observed that the generative results of our method, CVAE_1 and CVAE_2 do not significantly differ across various metrics such as FID, CA, TIME, and FI. However, our model's structure is simpler compared to CVAE_1 and CVAE_2, resulting in a slightly better performance in the TIME metric. Additionally, since our model's latent space excludes constraint information, the sampling results are more targeted,

Table 2: Quantitative evaluation. We conduct a quantitative evaluation of our method, ATISS, and LayoutVAE using four metrics: FID, CA, TIME, and FI. For FID and TIME, lower values indicate better generation, while for CA, a value closer to 0.5 indicates better performance. In contrast, higher values are desirable for FI. The best results are shown in bold.

Method	FID (↓)	CA (↓)	TIME (↓)	FI (↑)
Ours	**54.36**	**0.6607**	**1.891**	**0.9092**
ATISS	54.46	0.6996	2.432	0.7976
LayoutVAE	103.4	0.9560	2.595	0.8710

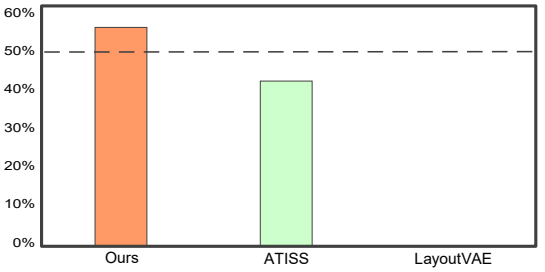


Figure 10: The result of perceptual study. The results show that most of the indoor scenes generated with our method achieve a high level of quality and enhanced functional integrity.

Table 3: Quantitative evaluation of ablation studies. We perform a quantitative evaluation of our method, $CVAE_1$ and $CVAE_2$, using four metrics: FID, CA, TIME, and FI. In $CVAE_1$, only the constraints of the functional category are included in the encoder of VAE. In $CVAE_2$, all constraints are added to the encoder. The best results are shown in bold.

Method	FID (↓)	CA (↓)	TIME (↓)	FI (↑)
Ours	**54.36**	0.6607	**1.891**	**0.9092**
$CVAE_1$	56.03	0.6479	1.939	0.8841
$CVAE_2$	55.69	**0.6380**	1.941	0.8967

resulting in a slight outperformance in both FID and FI metrics compared to $CVAE_1$ and $CVAE_2$ in both FID and FI metrics, although slightly worse in CA metrics. Overall, considering all the metrics of quantitative evaluation, our model design is reasonable.

5.4. Perceptual study

To better validate the practicality of the indoor scenes generated by our method, we have conducted a perceptual study to evaluate the generated indoor scenes.

We have recruited 20 university students who had undergone interior design training to participate in our perceptual study. To guide the participants, we design the questionnaire in two parts. The first part primarily presents a series of 3D models of common indoor furniture. This aimed to familiarize participants with various pieces of furniture that might appear in scenes. Additionally, to deepen participants' understanding of the concepts of "scene plausibility" and "functional integrity," we also show some specific counterexamples of indoor scenes. One type demonstrates the lack of "scene plausibility" by disrupting the placement of furniture in the scenes, such as having furniture placed out of bounds or encountering collisions between furniture pieces. The other type illustrates the deficiency of "functional completeness" by randomly removing some furniture from the scenes, such as missing function groups or furniture elements. In the second part of the questionnaire, we select 10 room boundaries from the testing dataset as constraints and generate corresponding indoor scenes using our method, ATISS, and LayoutVAE, respectively. In each task, participants are asked to evaluate the scene plausibility and functional integrity of the generated scenes and choose the best one from three options. A sample questionnaire has been provided in the supplementary materials. Finally, we calculate the proportions of participants who choose each of the three methods.

The results are shown in Figure 10, where the percentage of participants who chose our method, ATISS, and LayoutVAE are 57%, 43%, and 0%, respectively. LayoutVAE does not effectively model the implicit relationships between furniture items, and its training strategy involves generating objects in a fixed sequence, resulting in poorer performance. These results indicate that our method is more popular than the other two methods among users, generally producing higher-quality scenes with better functional integrity and better applicability.

5.5. Constrained Synthesis

By categorizing the furniture in the indoor scene into various function groups, our method is better suited to fulfill the constraints and controls of interior design. This is attributed to our division of the generative model into two components: LabelVAE and BBoxVAE. This separation reduces the coupling between these two parts and simultaneously

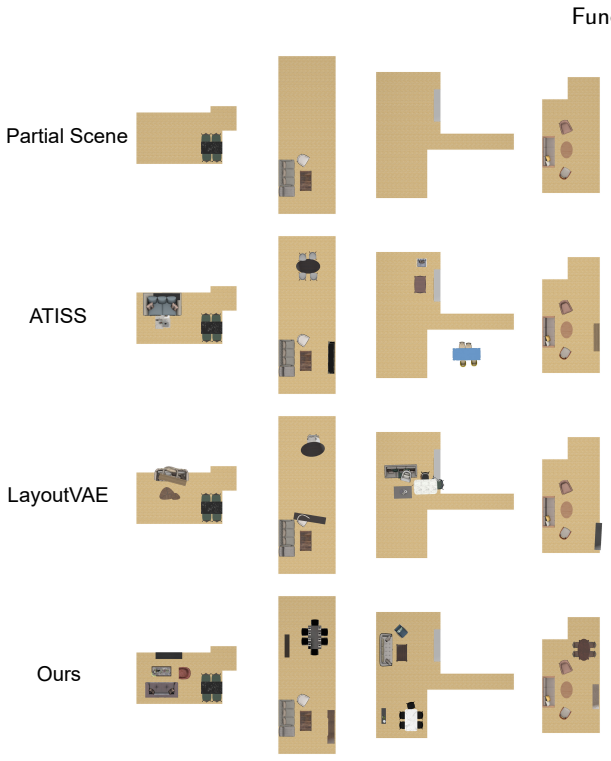


Figure 11: Scene completion. Given a partial scene, we attend automatically generate a complete indoor scene. From top to bottom, we show the indoor scenes of ground truth, the completed scene generated by ATISS, the completed scene generated by LayoutVAE, and the completed scene generated by our method.

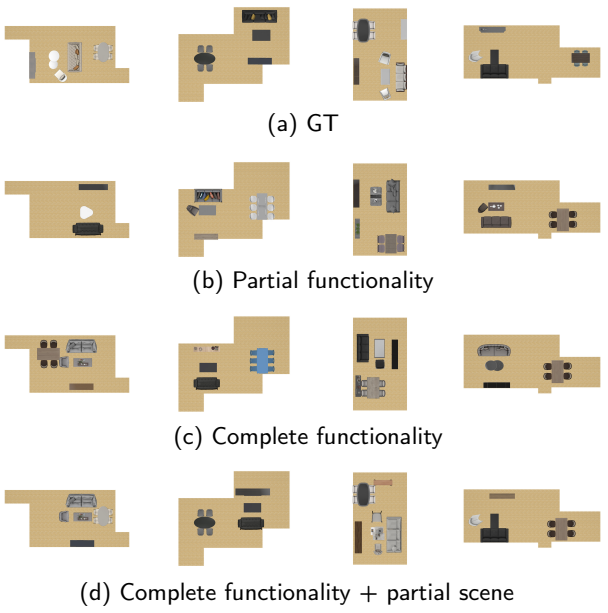


Figure 12: Synthesis control. Our method can realize multi-level generation control for indoor scene generation, which means that we can realize more constraint control during indoor scene generation to satisfy multiple needs of users. From top to bottom, it's the partial functionality constraint, complete functionality constraint, and a hybrid of complete functionality and partial scene constraint.

provides more options for synthesis control.

Scene completion. In our framework, partial scenes can serve as constraints for generating complete scenes, as shown in Figure 11. It's worth noting that LayoutVAE generates furniture items in a fixed order. Once the provided partial scene does not adhere to the predefined order, the generated results may lack coherence, as shown in the third row of Figure 11, it generates a number of furniture overlapping in the generated scenes. ATISS does not adhere to a fixed generation order when creating indoor scenes, thus resulting in relatively better consistency in the generated outcomes, as shown in the second row of Figure 11. However, due to limitations in the scale of our dataset and the fact that the input partial scenes of the scene completion are organized based on function groups, which differs from ATISS's furniture-centric generation, leading to a somewhat lower quality in the generated indoor scenes. The results produced by our method are depicted in the fourth row of the figure. It can be observed that in the context of a given partial scene, our method is capable of efficiently enhancing the quality of indoor scenes.

Synthesis control. In interior design, users often have specific requirements, such as specifying the functionality of a living room. As shown in Figure 12, we have conducted various constrained generations: partial functionality constraint, complete functionality constraint, and a hybrid of complete functionality and partial scene constraint. The partial functionality constraint involves providing the model with room boundaries and partial function group categories, constraining the model to generate a complete indoor scene. The complete label constraint requires providing room boundaries and all function group categories, constraining the model to generate a complete indoor scene. The hybrid of complete functionality and partial scene constraint means that room boundaries and all function group categories are provided, along with the bounding box of some function groups, and then the model generates a complete indoor scene. The hybrid of partial functionality and partial scene constraint equals the scene completion, which we have discussed before. The generation results are shown in Figure 12. It can be seen that our method can generate high-quality indoor scenes under various constraints. Comparing the results of ground truth in the first row, it shows

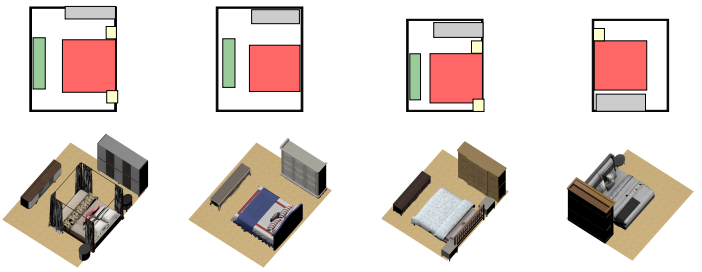


Figure 13: Bedroom synthesis. Our method can be applied to generate bedrooms. The training data of bedrooms is constructed from 3D-Front (Fu et al., 2021a), with 1212 bedrooms used for training and testing.

the high generalization of our method.

6. Conclusion

We have introduced FuncScene, which is a function-centric indoor scene generation framework. Function groups are the fundamental units for generating indoor scenes. This framework not only ensures the diversity and plausibility of the generated indoor scenes but also enhances their functionality and stylistic consistency. FuncScene also makes it easier to achieve multi-level control of scene generation. However, we are also facing some challenges, such as the limited dataset, and high-level design guidelines for interior design.

In this paper, we introduce our method in the context of synthesizing living rooms. However, it's important to note that our method can also be applied to generate other types of indoor scenes. As shown in Figure 13, we present some examples using our method to synthesize bedrooms. In the future, we plan to expand the model's generalization capability by increasing more high-quality data of indoor scenes. Additionally, we intend to introduce more function groups of different scenarios to further improve the generated results of our model, such as automatically generating public scenes, such as shopping centers and parks.

Acknowledgments

We would like to thank perceptual study participants for evaluating our system and the anonymous reviewers for their constructive suggestions and comments. This work is supported by the National Natural Science Foundation of China (62102126, 62372152), the National Key Research and Development Program of China (2022YFC3900800), and the Fundamental Research Funds for the Central Universities of China (JZ2023HGTB0269).

References

Chattopadhyay, A., Zhang, X., Wipf, D.P., Arora, H., Vidal, R., 2023. Learning graph variational autoencoders with constraints and structured priors for conditional indoor 3d scene generation, in: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 785–794.

Fisher, M., Ritchie, D., Savva, M., Funkhouser, T., Hanrahan, P., 2012. Example-based synthesis of 3d object arrangements. ACM Trans. Graph. 31, 1–11.

Fisher, M., Savva, M., Hanrahan, P., 2011. Characterizing structural relationships in scenes using graph kernels. ACM Trans. Graph. , 1–12.

Fu, H., Cai, B., Gao, L., Zhang, L.X., Wang, J., Li, C., Zeng, Q., Sun, C., Jia, R., Zhao, B., et al., 2021a. 3d-front: 3d furnished rooms with layouts and semantics, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10933–10942.

Fu, H., Jia, R., Gao, L., Gong, M., Zhao, B., Maybank, S., Tao, D., 2021b. 3d-future: 3d furniture shape with texture. International Journal of Computer Vision 129, 3313–3337.

Fu, Q., Chen, X., Wang, X., Wen, S., Zhou, B., Fu, H., 2017. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. ACM Trans. Graph. 36, 1–13.

Fu, Q., Fu, H., Yan, H., Zhou, B., Chen, X., Li, X., 2020. Human-centric metrics for indoor scene assessment and synthesis. Graphical Models 110, 101073.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S., 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc.. p. 6629–6640.

Jyothi, A.A., Durand, T., He, J., Sigal, L., Mori, G., 2019. Layoutvae: Stochastic scene layout generation from a label set, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9895–9904.

Kermani, Z.S., Liao, Z., Tan, P., Zhang, H., 2016. Learning 3d scene synthesis from annotated rgb-d images. Computer Graphics Forum 35, 197–206.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: International Conference on Learning Representations, pp. 1–13.

Kingma, D.P., Welling, M., 2014. Auto-encoding variational bayes, in: International Conference on Learning Representations, pp. 1–14.

Koh, J.Y., Agrawal, H., Batra, D., Tucker, R., Waters, A., Lee, H., Yang, Y., Baldridge, J., Anderson, P., 2023. Simple and effective synthesis of indoor 3d scenes, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1169–1178.

Li, M., Patil, A.G., Xu, K., Chaudhuri, S., Khan, O., Shamir, A., Tu, C., Chen, B., Cohen-Or, D., Zhang, H., 2019. Grains: Generative recursive autoencoders for indoor scenes. ACM Trans. Graph. 38, 1–16.

Liang, Y., Xu, F., Zhang, S.H., Lai, Y.K., Mu, T., 2018. Knowledge graph construction with structure and parameter learning for indoor scene design. Computational Visual Media 4, 123–137.

Liu, T., Chaudhuri, S., Kim, V.G., Huang, Q., Mitra, N.J., Funkhouser, T., 2014. Creating consistent scene graphs using a probabilistic grammar. ACM Trans. Graph. 33, 1–12.

Luo, A., Zhang, Z., Wu, J., Tenenbaum, J.B., 2020. End-to-end optimization of scene layout, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3754–3763.

Ma, R., Li, H., Zou, C., Liao, Z., Tong, X., Zhang, H., 2016. Action-driven 3d indoor scene evolution. ACM Trans. Graph. 35, 1–13.

Merrell, P., Schkufza, E., Li, Z., Agrawala, M., Koltun, V., 2011. Interactive furniture layout using interior design guidelines. ACM Trans. Graph. 30, 1–10.

Paschalidou, D., Kar, A., Shugrina, M., Kreis, K., Geiger, A., Fidler, S., 2021. Atiss: Autoregressive transformers for indoor scene synthesis. Advances in Neural Information Processing Systems 34, 12013–12026.

Purkait, P., Zach, C., Reid, I., 2020. Sg-vae: Scene grammar variational autoencoder to generate new indoor scenes, in: Computer Vision – ECCV 2020, Springer. pp. 155–171.

Qi, S., Zhu, Y., Huang, S., Jiang, C., Zhu, S.C., 2018. Human-centric indoor scene synthesis using stochastic grammar, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5899–5908.

Ritchie, D., Wang, K., Lin, Y.a., 2019. Fast and flexible indoor scene synthesis via deep convolutional generative models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6182–6190.

Savva, M., Chang, A.X., Hanrahan, P., Fisher, M., Nießner, M., 2016. PiGraphs: Learning Interaction Snapshots from Observations. ACM Trans. Graph. 35, 1–12.

Tang, J., Nie, Y., Markhasin, L., Dai, A., Thies, J., Nießner, M., 2024. Diffuscene: Denoising diffusion models for gerative indoor scene synthesis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1–21.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc.. p. 6000–6010.

Wang, K., Lin, Y.A., Weissmann, B., Savva, M., Chang, A.X., Ritchie, D., 2019. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. ACM Trans. Graph. 38, 1–15.

Wang, K., Savva, M., Chang, A.X., Ritchie, D., 2018. Deep convolutional priors for indoor scene synthesis. ACM Trans. Graph. 37, 1–14.

Wang, X., Yeshwanth, C., Nießner, M., 2021. Sceneformer: Indoor scene generation with transformers, in: 2021 International Conference on 3D Vision (3DV), IEEE. pp. 106–115.

Xia, X., Xu, C., Nan, B., 2017. Inception-v3 for flower classification, in: 2017 2nd international conference on image, vision and computing (ICIVC), IEEE. pp. 783–787.

Yan, M., Chen, X., Zhou, J., 2017. An interactive system for efficient 3d furniture arrangement, in: Proceedings of the Computer Graphics International Conference, pp. 1–6.

Yeh, Y.T., Yang, L., Watson, M., Goodman, N.D., Hanrahan, P., 2012. Synthesizing open worlds with constraints using locally annealed reversible jump mcmc. ACM Trans. Graph. 31, 1–11.

Yu, L.F., Yeung, S.K., Tang, C.K., Terzopoulos, D., Chan, T.F., Osher, S.J., 2011. Make it home: automatic optimization of furniture arrangement. ACM Trans. Graph. 30, 86.

Zhang, S.H., Zhang, S.K., Xie, W.Y., Luo, C.Y., Yang, Y.L., Fu, H., 2021a. Fast 3d indoor scene synthesis by learning spatial relation priors of objects. IEEE Transactions on Visualization and Computer Graphics 28, 3082–3092.

Zhang, S.K., Li, Y.X., He, Y., Yang, Y.L., Zhang, S.H., 2021b. Mageadd: real-time interaction simulation for scene synthesis, in: Proceedings of the 29th ACM International Conference on Multimedia, pp. 965–973.

Zhang, S.K., Xie, W.Y., Zhang, S.H., 2021c. Geometry-based layout generation with hyper-relations among objects. Graphical Models 116, 101–104.

Zhang, Z., Yang, Z., Ma, C., Luo, L., Huth, A., Vouga, E., Huang, Q., 2020. Deep generative modeling for scene synthesis via hybrid representations. ACM Trans. Graph. 39, 1–21.

Zhou, Y., While, Z., Kalogerakis, E., 2019. Scenegraphnet: Neural message passing for 3d indoor scene augmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7384–7392.