FOREST2SEQ: Revitalizing Order Prior for Sequential Indoor Scene Synthesis

Qi Sun[1⋆], Hang Zhou[2⋆], Wengang Zhou[1], Li Li[1], and Houqiang Li[1]

[1] USTC
[2] Simon Fraser University

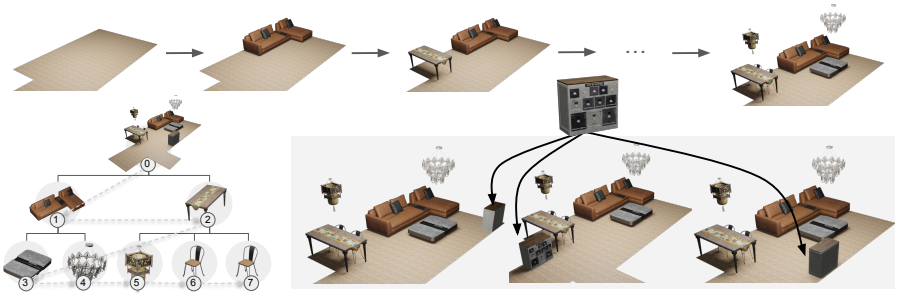

Fig. 1: We present FOREST2SEQ that mines the implicit hierarchy from the scene (bottom left), employing the tree-derived ordering as significant prior to direct the sequential indoor scene synthesis (top). The presence of placing-adaptable furniture items (bottom right), exemplified by the cabinets, necessitate the evolution from a single tree to scene forest representation.

Abstract. Synthesizing realistic 3D indoor scenes is a challenging task that traditionally relies on manual arrangement and annotation by expert designers. Recent advances in autoregressive models have automated this process, but they often lack semantic understanding of the relationships and hierarchies present in real-world scenes, yielding limited performance. In this paper, we propose FOREST2SEQ, a framework that formulates indoor scene synthesis as an order-aware sequential learning problem. FOREST2SEQ organizes the inherently unordered collection of scene objects into structured, ordered hierarchical scene trees and forests. By employing a clustering-based algorithm and a breadth-first traversal, FOREST2SEQ derives meaningful orderings and utilizes a transformer to generate realistic 3D scenes autoregressively. Experimental results on standard benchmarks demonstrate FOREST2SEQ's superiority in synthesizing more realistic scenes compared to top-performing baselines, with significant improvements in FID and KL scores. Our additional experiments for downstream tasks and ablation studies also confirm the importance of incorporating order as a prior in 3D scene generation.

Keywords: Auto-regressive model · Indoor scene synthesis · Order

⋆ Equal contributions; Work carried out at SFU by Hang.

1 Introduction

Creating realistic virtual 3D indoor scenes has long been an expensive, labor-intensive process [7, 43, 69], requiring expert designers to manually arrange and annotate every piece to populate these rich environments [10, 31, 49]. Recent advances in indoor scene synthesis, however, enable automating this process by generating plausible room layouts simply by taking high-level room type and layout [47,53,59,63] as inputs. This has immense potential, enabling virtual product showcasing for retailers [1,35], automating environment creation for movies [33], games [74] and complex visualization [48], and providing rich training data for 3D scene understanding AI models [4, 9, 58].

Early indoor synthesis approaches were formulated as a prior constraint optimization task [17,30,43,55], achieving promising results. They encoded rules and priors about functional relationships between objects [68,69] (like couches facing TVs) as well as human use-case constraints [42, 73], which are time-consuming and skill-dependent. As a result, this hand-crafted prior approach lacks flexibility and generality as the 3D indoor scene becomes more complex [47, 53].

As an alternative to hand-crafted priors, deep generative models have been employed to learn scene priors directly from data, without the need for manually specified rules or constraints. Such approaches include autoregressive transformer models [46,47,64] and CNN-based methods [63,66]. While autoregressive models generate objects in sequential order, a key limitation is that this order is arbitrary [13, 62, 67] and lacks semantic understanding of the relationships and hierarchies that exist in real 3D indoor scenes [59].

To address these limitations, we propose Forest2Seq, a framework that formulates indoor scene synthesis as an order-aware sequential learning problem. Forest2Seq organizes the inherently unordered collection of scene objects into structured, ordered hierarchical scene trees and forests. Specifically, Forest2Seq first establishes orderings that prioritize the placement of dominant furniture pieces before the associated secondary objects, aligning with intuitive spatial reasoning principles for scene composition.

As illustrated in Figure 1 (bottom left), Forest2Seq used a clustering-based algorithm to parse the scene into a tree, which is then linearized into an ordered sequence via breadth-first traversal. To handle flexible objects like cabinets that can belong to multiple functional zones, as depicted in Figure 1 (bottom right), we extend this to an ensemble of trees forming a scene forest representation. With these derived ordering, Forest2Seq employs a transformer coupled with a denoising strategy. At inference time, Forest2Seq generates plausible 3D scenes auto-regressively by sequentially placing furniture instances guided by the predicted order as shown in Figure 1 (top).

We demonstrate the capability of Forest2Seq to synthesize more realistic scenes using the 3D-FRONT dataset, outperforming the top-performing baseline with an average margin of FID score of 2.58 and a KL score of 1.78. Additionally, we illustrate how Forest2Seq enhances scene rearrangement and completion tasks. Our extensive ablation studies further confirm that order-awareness as a prior significantly improves the generation of 3D indoor scenes.

2 Related Works

Scene synthesis with handcrafted priors. Early works reasoned scene synthesis with various probabilistic models over scene exemplars in the view of object functionality criteria [17, 42, 73] and human activities [18, 41, 55]. For example, Fisher *et al.* [17] investigated Bayesian network and Gaussian mixture model to model object co-occurrence. Make-it-Home [68] that pioneered in progressive synthesis, offered an interactive layout modeling tool by optimizing cost function that encoding spatial relation between furniture objects. Ma *et al.* [41] formulated an action graph with nodes represented as human actions, guiding interior sythesis from human activity. In contrast, our approach is an end-to-end differentiable pipeline and free of externally introduced elements previously proposed for scene synthesis.

Scene synthesis via graph representation. Modeling scenes as graphs [23, 26, 37, 59, 63, 65, 75] has been an intuitive approach for scene synthesis and been extensively studied recently. GRAINS [37] proposed a recursive auto-encoder network for scene synthesis, where novel scenes are generated hierarchically. SceneGraphNet [75] utilized message-passing graph networks to model long-range relationships among objects. SceneHGN [23] defined a fine-grained hierarchy in room-object-part order, allowing multi-level scene editing. LEGO-Net [65] and DiffuScene [59] represented scene as a fully-connected graph with denoising diffusion probabilistic models (DDPMs). CommonScenes [72] modeled scene jointly with layout and shape via latent diffusion models. Our work represents scene as directed rooted forest and learns scene synthesis with transformers.

Scene synthesis using language modeling. Recently, great success have been made in language modeling [51, 61] for content generation [5, 6, 14, 38, 44, 54]. SceneFormer [64] modeled each scene object property with individual transformer network, where object order is predefined by class frequency. ATISS [47] represented object properties as span using a single transformer network and removed positional encoding for order permutation-invariance. CLIP-Layout [39] learned to synthesize style-consistent indoor scenes with multi-modal CLIP [50] encoder. LayoutGPT [16] leveraged rich visual concepts and notable zero-shot capabilities of large language models (LLMs), *i.e.* ChatGPT [45]. COFS [46] adapted masked language models and modeled scenes with a standard BART [36]-like generative model, formed by a bidirectional encoder over corrupted input and an auto-regressive decoder. In our method, we leverage the decoder-only casual transformer and denoising strategy to enhance the generation ability.

Input as sequence or set. A significant limitation of language modeling is it can only be applied to problems whose inputs are represented as *sequences*. Hence, many research efforts [13, 29, 62, 67] have been made to perform mappings from different data structure, like set, to sequences. Set2Seq [62] proposed read, process, write block to process the input set and find the optimal orderings while training, the results of which show that order matters in various tasks. Tree2Seq [8] added unsupervised hierarchical structure on the source sentence to Seq2Seq model for improving low-resource machine translation. CODE-NN [29] showcased the effectiveness of transforming structured code into sequence cou-

pled with LSTM [28] for summarization. Another line of research initiatives have been focusing on processing the *set*-input data. DeepSets [70] handled set data by ensuring permutation-invariance and enabling pooling over sets. Set Transformer [34] featured as induced set attention block and attentional pooling module to aggregate set input attributes. Since the indoor scenes do not provide explicit ordering, following Set2Seq, our method ventures to search the optimal priori choice of the indoor scene ordering for sequential modeling.

3 Method

Given a 2D floor/layout, we aim to develop a generative model to produce diverse and plausible object arrangements. Figure 2 shows the framework of our proposed method.

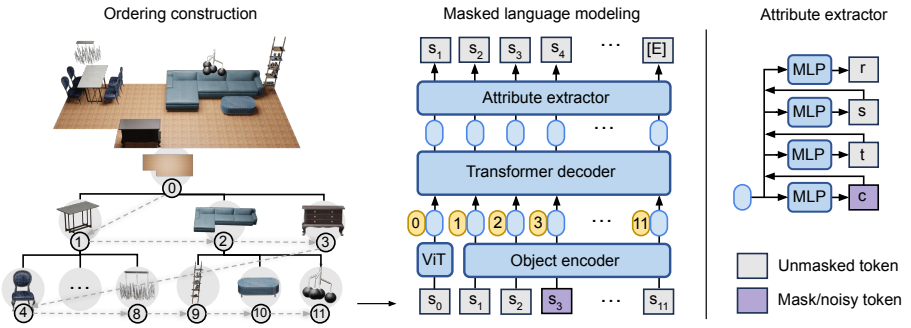

Fig. 2: Training framework of our FOREST2SEQ. On the left, we depict the construction of a tree/forest from parsing the scene and its subsequent flattening into a sequence through breadth-first search. The right panel illustrates our use of a causal transformer equipped with a denoising strategy for sequential data learning.

3.1 Ordering construction

Scenes are represented as sets of oriented bounding boxes $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$, with each box $o_i = (c_i, t_i, b_i, r_i)$ containing a semantic class $c_i \in \mathbb{R}^C$, a 3D translation $t_i \in \mathbb{R}^3$, a bounding box size $b_i \in \mathbb{R}^3$, and a rotation angle $r_i \in \mathbb{R}$. To enable transformer-based sequence-to-sequence learning, we seek a permutation π that transforms the set $\mathcal{O}$ into a sequence $\mathcal{S} = \pi(\mathcal{O})$.

Scene tree. In Figure 2 (left), the implicit tree structure of the living room is evident: Each subtree corresponds to a distinct functional zone, such as relaxation zone or dining zone, with the primary object forming the parent node and the ancillary objects as children. This arrangement aligns well with intuitive spatial reasoning, where primary objects are positioned first, followed by

their associated items. We posit this *hierarchy*, which we refer to as scene tree ordering π_T, naturally suits indoor scene composition and aligns with practical spatial organization.

To realize the goal, we build a tree with Modified Euclidean Distance Clustering (MEDC), where on the top-down projected 2D bounding boxes $\overline{o}_i = (\overline{t}_i, \overline{s}_i)$, we can calculate the distance between each pair, formulating a distance matrix $M = \{m_{ij}\}_{i,j=1}^{N}$ defined as:

$$m_{ij} = d_{ij} + \lambda \cdot (1 - \text{GIoU}(\overline{o}_i, \overline{o}_j)), \tag{1}$$

where $\text{GIoU}(\cdot,\cdot) \in [-1, 1]$ is proposed in [52] to evaluate of distance of two bounding boxes $\overline{o}_i, \overline{o}_j \in \mathbb{R}^4$, and d_{ij} is the Euclidean distance between two bounding box centers. We employ the DBSCAN algorithm [15] on the distance matrix to segment the furniture into multiple clusters and identify outliers. Within each cluster, the largest object is designated as the root node, forming a subscene, with the remaining items as child nodes. To eliminate inherent ordering among siblings, we randomly shuffle nodes under the same parent. Then, utilizing breadth-first search (BFS) [3], we linearize the shuffled tree into a sequence $\mathcal{S}_T = \pi_T(\mathcal{O})$.

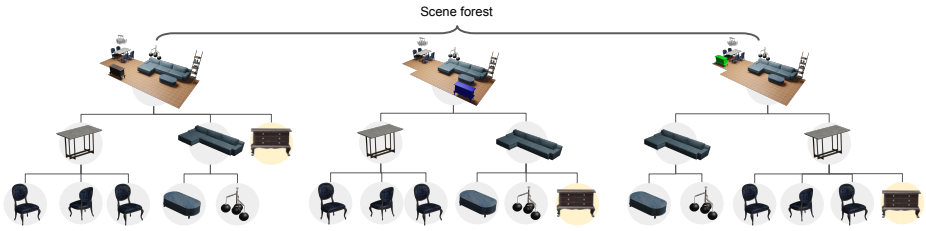

Fig. 3: An example to illustrate the motivation of scene forest. The whole room is clearly divided to 2 subscenes according to the human activity. However, "cabinet" is an exception as it can reasonably belong to any subscene or the entire scene. Note that some items in the base tree are ignored for simplification.

Scene forest. As depicted in in Figure 3, flexible furniture like cabinets capable of serving relaxation, dining, or the entire room, introduces ambiguity into the conventional scene tree representation. To resolve this, we propose an enriched tree structure that integrates these outliers by associating them with each potential parent node, resulting in san ensemble of trees, as the forest scene representation. During training, we randomly select a tree from the forest (see Figure 3) and employ BFS to convert it into a sequence, denoted as $\mathcal{S}_F = \pi_F(\mathcal{O})$. sNote that this forest ordering approach, which produces inherently non-unique orderings, broadens the permutation space beyond what is possible with a single scene tree. More details of the algorithm are provided in the supplementary.

3.2 Masked language modeling

Upon establishing the scene priori ordering of the indoor scenes, see Figure 2 (right), our objective is to employ a transformer-based generative model to populate the ordered sequence $\mathcal{S} = \{s_i\}_{i=1}^{N}$. The framework primarily comprises four components, including layout encoder, object encoder, transformer decoder, and attribute extractor.

Two encoder networks including layout encoder and object encoder transform raw input into sequential embeddings. The layout encoder, a small vision transformer [12, 22], extracts the binary layout mask $s_0 \in \mathbb{R}^{64\times64}$ into a start token $x_0 \in \mathbb{R}^{512}$, establishing spatial constraints for object placement. The object encoder handles attributes (c_i, t_i, b_i, r_i), coupled with sinusoidal positional encoding for extracting the discrete variable into a unified token $x_i = [\lambda(c_i); \psi(t_i); \psi(b_i); \psi(r_i)] \in \mathbb{R}^{512}$.

The transformer decoder [51], denoted as f_θ, is tasked with the prediction of subsequent object embeddings. It achieves this by accumulating context from both the start layout token and previous object tokens, sleverages masked self-attention mechanisms. To enhance this process, we integrate absolute positional encodings to the sequence of object embeddings, thereby enpowering the model with crucial sequence order information:

$$\hat{x}_i = f_\theta(x_{<i}; x_0), \tag{2}$$

where $\hat{x}_i$ represents the predicted embedding of the next object, and $x_{<i}$ is the sequence of all previous object embeddings up to the i-th position.

Following previous work [47], our feature extractor outputs a probability distribution for bounding box parameters, using K logistic distributions for continuous parameters including position, size, and orientation:

$$p(h) = \sum_{j=1}^{T} \alpha_j \text{Logistic}(\mu_j, \sigma_j), \tag{3}$$

where h is a component (t_i, b_i, r_i), with $\alpha_j, \mu_j, \sigma_j$ being the logistic parameters. And discrete class probabilities are derived from logit vectors l_c via the softmax function:

$$p(c_i) = \text{Softmax}(l_c). \tag{4}$$

Probability distributions are represented by $3K$-dim vectors for continuous, and C-dim for discrete class variables, where C is the class number.

Training with denoising. We employ input corruption techniques [11,24,25] to mitigate overfitting in the transformer in both attribute- and object token-level. Specifically, a predefined percentage of the object embeddings are randomly substituted with a `[MASK]` token. In addition, in the auto-regressive attribute prediction process, a predefined percentage of the ground truth categories are replaced with random categories, rather than consistent teacher forcing. We use the same 5% mask/noise rate for simplification. Moreover, this design reduces the error propagation in sequential predictions.

Inference. During the inference phase, the process initiates with the layout embedding or start token. Subsequently, we employ an auto-regressive approach to iteratively sample attribute values from the distributions predicted for the subsequent object. Each newly generated object is concatenated with the preceding tokens, which then informs the subsequent generation step. This procedure is repeated until the end token is produced.
Object retrieval. Once a labeled oriented bounding box is sampled, we identify the matching furniture instance from the 3D-FUTURE dataset [20] by selecting the closest size match within the predicted object category.

3.3 Objective loss functions

The language model is trained by minimizing the negative log-likelihood of the sequence joint distribution, factorized as the product of conditional probabilities across individual tokens [2]:

$$\mathcal{L}_\theta = -\log \prod_{i=1}^{N} p_\theta(s_i|s_{<i}) = -\sum_{i=1}^{N} \log p_\theta(s_i|s_{<i}), \tag{5}$$

where $p_\theta(s_i|s_{<i})$ is the cross entropy between the probability of the next object attributes predicted by model and the ground truth, given the previous i tokens.

3.4 Implementation details

All experiments run on an NVIDIA RTX3090 GPU, with the AdamW [40] optimizer at a 1e-4 learning rate, without warm-up or decay strategies. For DBSCAN parameters, the eps and min-samples are set to 0.15 and 2 respectively, with a GIoU weight of λ=0.02 applied to the distance matrix. The attribute modeling employs a 10-component logistic mixture to accurately represent object distributions. All models are trained using a batch size of 128 across 1000 epochs, incorporating random rotations from 0° to 360° for data augmentation. We adopt standard practice for early stopping, evaluating against the validation metric every 10 epochs and selecting the best-performing iteration as the final model. All layers are applied with a universal dropout rate of 0.1 to counter overfitting. Details of the network architecture are provided in the supplementary.

4 Experiments

4.1 Experimental settings

Datasets. In alignment with prior work [46, 47, 59], we utilize the 3D-FRONT dataset [19] for training and evaluation, which includes around 10k professional 3D indoor scenes spanning bedrooms, libraries, living rooms, and dining rooms. Following the preprocessing steps of ATISS [47], the dataset is split into subsets with 3879/162 bedrooms, 230/56 libraries, 621/270 living rooms, and 723/177

dining rooms for training/validation. Given the limited library/living room/dining room data, we employ a pretrained bedroom model for initial weights for each room type to improve performance.

Metrics. In line with established works [46, 47], we employ various metrics for performance assessment, including Fréchet Inception Distance (FID) [27], classification accuracy score (CAS), and categorical Kullback-Leibler (KL) divergence. The evaluation protocol involves rendering the indoor scenes into 256×256 orthographic maps and calculating the CAS/FID scores against the ground truth.

Baselines. Our method is benchmarked against recent advancements, including FastSynth [53], SceneFormer [64], ATISS [47], COFS [46], LayoutGPT [16], and DiffuScene [59]. Notably, DiffuScene utilizes top-down semantic maps for rendering 3D scenes, while LayoutGPT computes FID using images rendered from four distinct camera viewpoints. To ensure fair comparisons, we reproduce the experiment of these two methods using their official implementations. We do not report the results of DiffuScene on the library type, since the original paper did not conduct this experiment.

4.2 Scene synthesis

Quantitative comparison. Our experiments on scene synthesis, when measured against baseline models, demonstrate the robustness of our approach. As described in Table 2, Forest2Seq surpasses all baselines with superior KL divergence scores, implying a closer match to the ground truth. Furthermore, Forest2Seq achieves the most favorable scores in both the FID and CAS, indicating its ability to render more realistic scenes. Table 1 shows the basic statistics of the scene forest: living room and dining room contains the most nodes and trees, which explains performance improvement more substantial in the two room types. In terms of computational efficiency, as illustrated in Table 3b, our model demonstrates competitive inference times and requires the fewest parameters (48% fewer than the model with the second fewest parameters) among all methods compared. It is noteworthy that we employ a compact transformer decoder and a small ViT-based layout encoder (2.58M) to mitigate overfitting and improve efficiency.

Room type	bedroom	living room	dining room	library
Num. of nodes	5.00	11.7	10.9	4.61
Num. of trees	1.27	2.83	2.51	1.14

Table 1: Forest statistics.

Qualitative comparison. In Figure 4, we illustrate the visual results of scene synthesis. To ensure a balanced evaluation, the same randomly sampled room layout serves as the conditional input for DiffuScene, ATISS, COFS, and our Forest2Seq across various room types. The comparison reveals that scenes synthesized by Forest2Seq exhibit greater alignment with the given floor plans and demonstrate a reduced tendency to place furniture beyond room boundaries.

Perceptual study. In Table 3a, we conduct user study by 37 participants who assessed the realism of scenes generated by DiffuScene, ATISS, and COFS across 51 randomly selected rooms of each type for evaluation. The perceptual analysis

Method	Bedroom			Dining room			Living room			Library		
	KL	FID	CAS (%)	KL	FID	CAS (%)	KL	FID	CAS (%)	KL	FID	CAS (%)
FastSyn [53]	6.4	88.1	88.3	51.8	58.9	93.5	17.6	66.6	94.5	43.1	86.6	81.5
SceneFormer [64]	5.2	90.6	97.2	36.8	60.1	71.3	31.3	68.1	72.6	23.2	89.1	88.0
LayoutGPT [16]	17.5	68.1	60.6	—	—	—	14.0	76.3	94.5	—	—	—
ATISS [47]	8.6	73.0	61.1	15.6	47.6	69.1	14.1	43.3	76.4	10.1	75.3	61.7
COFS [46]	5.0	73.2	61.0	9.3	43.1	76.1	8.1	35.9	78.9	6.7	75.7	66.2
DiffuScene [59]	5.1	69.0	59.7	7.9	45.8	70.6	8.3	38.2	75.1	—	—	—
Ours	**4.2**	**67.9**	**58.3**	**5.5**	**40.2**	**65.6**	**5.9**	**35.2**	**68.0**	**5.2**	**69.1**	**57.3**

Table 2: Quantitative comparison with the state-of-the-art methods [46, 47, 53, 59, 64] on the task of scene synthesis. Note that for FID and KL, lower is better, and for CAS, the score closer to 50% is better.

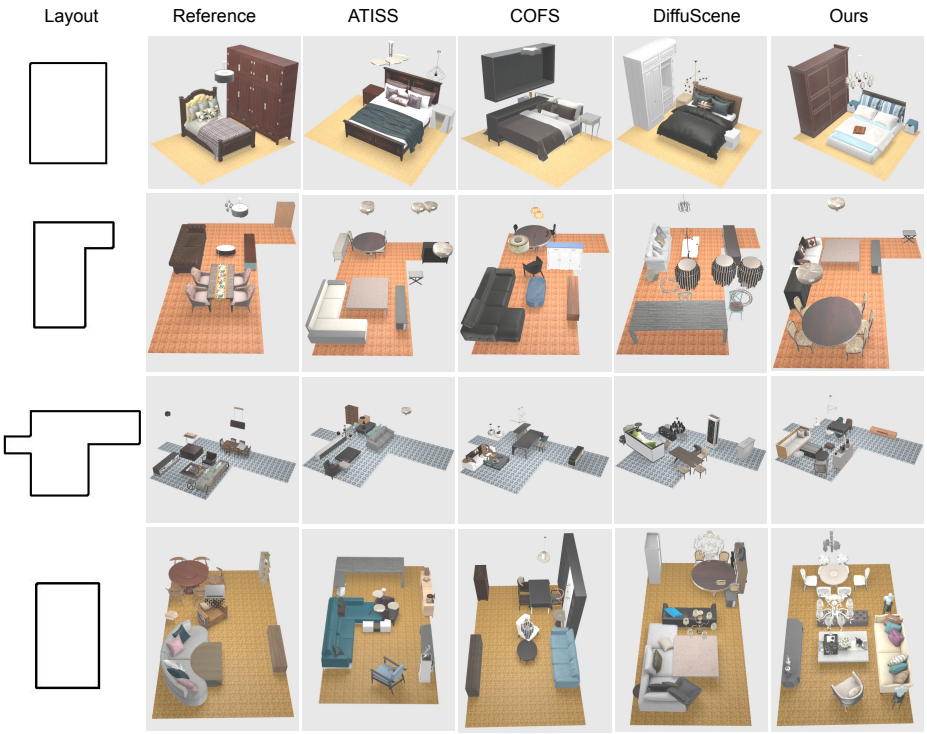

Fig. 4: Qualitative comparison with the-state-of-the-art methods [46, 47, 59] on scene synthesis for three type of scenes: bedrooms (1st row), living room (2nd and 3rd rows) and dining room (4th row). Note that reference is the scene from dataset with the same floor plan.

reveals that more than 50% of our generated scenes for living rooms and dining rooms are considered realistic than others. Furthermore, our results gain a predominant preference in bedrooms and libraries.

Method	Bedroom	Living room	Dining room	Library
ATISS	20.2	10.1	13.9	26.5
COFS	13.4	21.2	8.5	25.0
DiffuScene	23.1	16.7	19.6	—
Ours	**43.3**	**52.0**	**58.0**	**48.5**

(a) User study

Method	ATISS	COFS	DiffuScene	Ours
Parameters (MB)	36.1	19.4	74.1	**9.99**
Inference rate (s)	0.204	**0.129**	34.9	0.160

(b) Efficiency comparisons

Table 3: Additional quantitative comparison with the state-of-the-arts.

Analysis. ATISS utilizes a transformer encoder without positional encoding and randomizes object order while training to achieve approximate permutation invariance. Similarly, COFS posits that the layout is inherently unordered, leveraging BART, a masked language model, to underpin this assumption. DiffuScene represents scenes as fully-connected graphs and learns to denoise over Gaussian noise. However, these baselines do not account for the inherent ordering among objects. This oversight often results in predictions that place objects too close to one another, leading to frequent intersections and thereby affecting scene realism. On the other hand, our method introduces multiple strategies to mitigate overfitting: 1) our scene forest offers a priori ordering that guides the generative process; 2) we adopt a decoder-only architecture, which not only reduces the network capacity but also exhibits enhanced generative performance over encoder-decoder frameworks [21] such as BART; 3) the denoising training strategy improves the generalization ability of the model.

4.3 Discussion on the order prior

In Table 4, we incorporate different orderings scheme with the auto-regressive generating network, in which each scene can be represented either with a single (the first three columns) or multiple (the last three columns) ordered sequences. We also report two statistics for the set of sequences: diversity means that the average number of ordered sequences; inconsistency is evaluated by the average hamming distance between two sequence pair within the set.

Order (set)	Random(single)	Fixed	Tree+BFS	Random(multiple)	Forest+DFS	Forest+BFS
Diversity	1	1	1	∞	2.83	2.83
Inconsistency	0	0	0	9.54	4.01	1.87
KL ↓	20.0	17.9	<u>7.90</u>	13.1	9.40	**5.90**
FID ↓	49.4	49.8	<u>36.1</u>	43.3	40.5	**35.2**
CAS (%)	83.7	80.1	<u>68.1</u>	76.4	71.7	**68.0**

Table 4: Ablation study: comparison of KL, FID and CAS of different order settings in living room scenario. The second-best score is underscored.

Superiority of forest ordering. We observe that random permutation and fixed order [64] using frequency-based arrangement perform worse than tree-guided order. This indicates that an optimal ordering could be beneficial for

scene modeling. The further improvement in KL from the tree to forest demonstrates the benefit of our introducing scene forest, which explicitly model the flexible objects. When comparing breadth-first (BFS) and depth-first (DFS) traversal methods for scene tree sequences, we find that BFS yields superior results due to the greater consistency within sets of BFS-derived sequences. Additionally, we examine the ordering strategy that is utilized in ATISS [47] that shuffles the sequence before fed into the network. This approach, representing a scene as multiple random sequences, shows improvement over the single random case, attributed to data augmentation and the resulting high diversity. However, its performance is constrained by sequence inconsistency (9.54) within the set. Supported by qualitative evidence in Figure 5, these findings underscore the effectiveness of our proposed forest ordering in generating more realistic and diverse object arrangements. Notably, the placement of primary furniture is less accurate in the DFS scenario compared to the BFS scenario.

Fig. 5: Ablation study: visual comparison of different orderings. While random order and fixed order can not provide appropriate prior, our tree-guided order benefits the scene synthesis, generating plausible scenes. The forest representation further enhances the scene diversity and realism.

On the methods for forest formation/order construction. The goal of forest formation is to reconstruct the scene hierarchy from sthe given object representations. Several studies [57,60,71] have explored learning tree structures or scene graphs from natural images. Consequently, we evaluate the scene parsing efficacy of pure Euclidean Distance Clustering (EDC), Modified Euclidean Distance Clustering (MEDC), and a learning-based baseline, VCTree [60] that was originally designed for parsing natural images. We assess reconstruction accuracy using Average Hierarchical Distance (AHD), which calculates similarity by comparing sets formed at each tree depth between the reconstructed and human-annotated trees. AHD averages these set-based similarity scores across all depths for a concise measure of accuracy. As indicated in Table 5a, while the proposed MEDC method is straightforward, it effectively captures underlying semantic relationships and offers computational efficiency advantages.

Methods	AHD	Inference time(ms)
MEDC	**0.84**	0.827
EDC	0.53	0.777
VCTree [60]	0.77	56.1

(a) On different scene parsing methods.

Positional Encoding type	KL	FID	CAS (%)
w/o	11.1	41.2	74.3
w. absolute [61]	**5.9**	**35.2**	**68.0**
w. relative [56]	6.2	36.7	70.1

(b) On the type of positional encoding.

Mask/noise rate	0	0.05	0.1	0.15	0.2
KL ↓	6.4	**5.9**	6.1	8.8	12.2
FID ↓	35.6	35.2	**34.4**	39.6	42.5
CAS (%)	69.2	68.0	**66.7**	73.1	74.5

(c) On the mask/noise rate in denoising training scheme.

Table 5: Ablation study.

Attention visualization. Figure 6 presents a visualization of the attention heatmaps trained under multiple random ordering and scene forest ordering. It is evident that with our scene forest ordering, pivotal objects receive heightened attention scores in the next object token prediction. This inherent bias of underlying model confirms that the order prior affects the model, also explains why the object embeddings are predicted more accurately. Conversely, in the unordered scenario, the distribution of attention scores is notably uniform across the object sequence without attention efficacy.

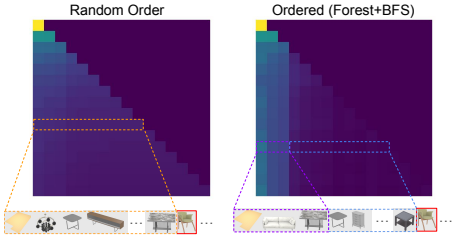

Fig. 6: Attention map for different ordering. The dash line boxes indicates the previous objects that are used to predict the next objects that are annotated with red solid boxes.

4.4 More ablation study

On the positional encoding. A core component of Transformers is the positional encoding mechanism that represents the order of input sequence [32]. We compare the results of transformer decoder variant without positional encoding and those incorporating relative [56] and absolute [61] positional encoding. In Table 5b, the results clearly demonstrate that incorporating positional encoding facilitates learning of order, with the model variant employing absolute positional encoding achieving marginally better generation outcomes.

On denoising strategy. Table 5c shows the effectiveness of our denoising strategy. By introducing a small rate (0.05 or 0.1) of masked and random tokens, we effectively prevent overfitting and enhance model generalization. Conversely, at a higher mask/noise rate (0.2), we find that achieving convergence of training loss becomes challenging.

4.5 Application in downstream tasks

With the sequential generation framework, our method is easily applicable to various downstream tasks, such as scene completion and rearrangement.

Scene completion. We retain the ground truth of the first N objects, predicting subsequent tokens auto-regressively util the sequence concludes. In Figure 7, we

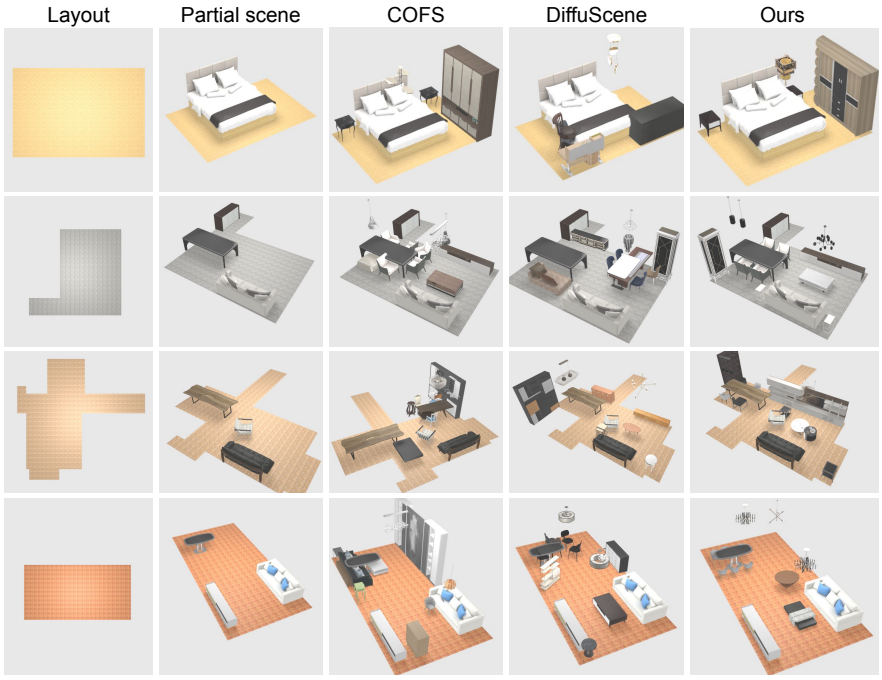

Fig. 7: Scene completion – Comparison with COFS and DiffuScene for bedroom (1st row) and living room (the bottom three rows).

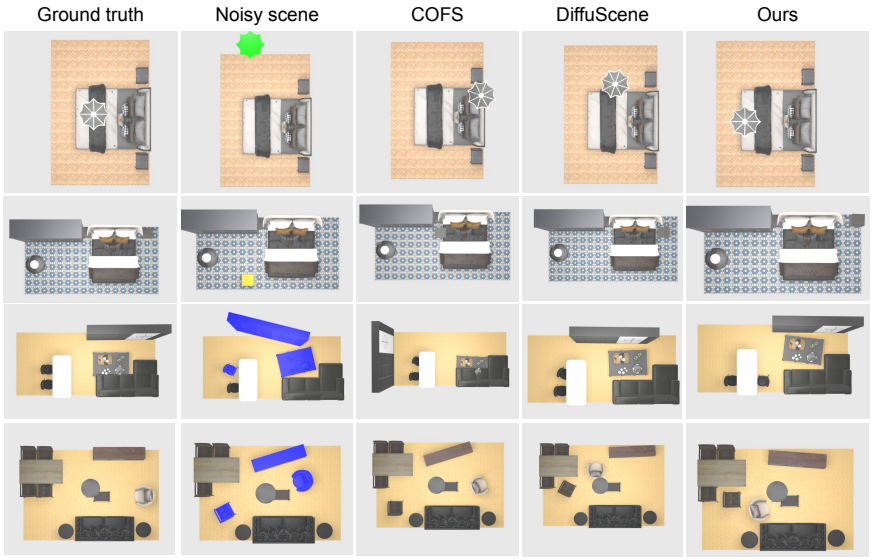

Fig. 8: Scene rearrangement – Comparison with COFS and DiffuScene for living room (the top two rows) and dining room (the bottom two rows).

compare against the state-of-the-art method, DiffuScene [59] and COFS [46], on the task. Unlike DiffuScene and COFS, which omits essential items like lights (1st row) and introduces misplaced elements such as dining tables and cabinets (2nd row), our approach (3nd column) yields clean and coherent scenes due to the awareness of the key furniture placement.

Rearangement. Our method corrects one or multiple failure cases by resampling the position of an object considering prior inputs. Figure 8 illustrates our success in adjusting the location of a night stand (2nd row) and optimally placing the bookshelf and chairs (the last row), an improvement over the inability of DiffuScene to correct these placements.

5 Conclusion, Limitations and Future Work

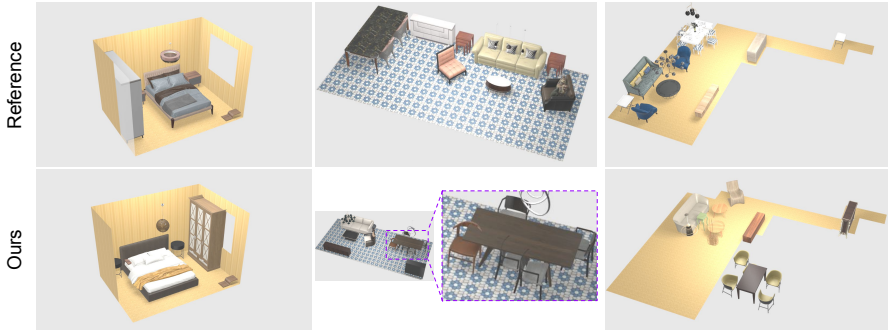

Fig. 9: Failure cases – Neglecting window placement (left); overlapping furniture arrangements (mid); objects placed out of boundary in non-standard layouts (right).

We have introduced FOREST2SEQ, a novel framework to synthesize indoor scenes via sequential modeling. In contrast with previous works that neglect ordering, we leverage a parsing tree/forest and breadth-first search (BFS) to explore the implicit scene ordering, train the underlying network in an order-aware manner, and validate that *order matters* in scene modeling through extensive experiments including scene synthesis, completion and rearrangement.

Figure 9 highlights three primary limitations of our method. The left column indicates that the cabinet is blocking the window, creating undesirable scenes. This occurs since our model currently not factoring doors and windows as additional condition. In the mid column, we observe occasional instance of object overlaps due to a lack of spatial constraints on relative positioning. As demonstrated by the right, the model struggles with generating plausible results for highly complex layouts, partially due to the limited diversity of training data.

In addition to addressing the difficulties suggested by the failure cases, the future work should explore towards the optimal order for sequential generation. For example, this could involve integrating learning-to-ordering module into an end-to-end network by joint optimizing the ordering and the likelihood objective.

Acknowledgements: The authors would like to thank Nathan Yan (Cornell) and Prof. Jing Liao (CityU) for useful comments and discussions. This work is supported by National Natural Science Foundation of China under Contract 62021001 and the Youth Innovation Promotion Association CAS. It was also supported by GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC, and the Supercomputing Center of the USTC.

References

1. https://www.ikea.com/pt/en/customer-service/services/planning-consultation
2. Bengio, Y., Ducharme, R., Vincent, P.: A neural probabilistic language model. In: NeurIPS (2000)
3. Bundy, A., Wallen, L.: Breadth-first search. Catalogue of Artificial Intelligence Tools (1984)
4. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. 3DV (2017)
5. Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M.H., Murphy, K., Freeman, W.T., Rubinstein, M., et al.: Muse: Text-to-image generation via masked generative transformers. arXiv preprint arXiv:2301.00704 (2023)
6. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: MaskGiT: Masked generative image transformer. In: CVPR (2022)
7. Chaudhuri, S., Kalogerakis, E., Giguere, S., Funkhouser, T.: Attribit: content creation with semantic attributes. In: UIST (2013)
8. Currey, A., Heafield, K.: Unsupervised source hierarchies for low-resource neural machine translation. In: ACL (2018)
9. Deitke, M., VanderBilt, E., Herrasti, A., Weihs, L., Salvador, J., Ehsani, K., Han, W., Kolve, E., Farhadi, A., Kembhavi, A., Mottaghi, R.: ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. In: NeurIPS (2022)
10. Devaranjan, J., Kar, A., Fidler, S.: Meta-sim2: Unsupervised learning of scene structure for synthetic data generation. In: ECCV (2020)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
13. Eriguchi, A., Hashimoto, K., Tsuruoka, Y.: Tree-to-sequence attentional neural machine translation. In: ACL (2016)
14. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR (2021)
15. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD (1996)
16. Feng, W., Zhu, W., Fu, T.j., Jampani, V., Akula, A., He, X., Basu, S., Wang, X.E., Wang, W.Y.: LayoutGPT: Compositional visual planning and generation with large language models. arXiv preprint arXiv:2305.15393 (2023)
17. Fisher, M., Ritchie, D., Savva, M., Funkhouser, T., Hanrahan, P.: Example-based synthesis of 3D object arrangements. ACM TOG (2012)

18. Fisher, M., Savva, M., Li, Y., Hanrahan, P., Nießner, M.: Activity-centric scene synthesis for functional 3D scene modeling. ACM TOG (2015)
19. Fu, H., Cai, B., Gao, L., Zhang, L.X., Wang, J., Li, C., Zeng, Q., Sun, C., Jia, R., Zhao, B., et al.: 3D-FRONT: 3D furnished rooms with layouts and semantics. In: CVPR (2021)
20. Fu, H., Jia, R., Gao, L., Gong, M., Zhao, B., Maybank, S., Tao, D.: 3D-FUTURE: 3D furniture shape with texture. IJCV (2021)
21. Fu, Z., Lam, W., Yu, Q., So, A.M.C., Hu, S., Liu, Z., Collier, N.: Decoder-only or encoder-decoder? Interpreting language model as a regularized encoder-decoder. arXiv preprint arXiv:2304.04052 (2023)
22. Gani, H., Naseer, M., Yaqub, M.: How to train vision transformer on small-scale datasets? In: BMVC (2022)
23. Gao, L., Sun, J.M., Mo, K., Lai, Y.K., Guibas, L.J., Yang, J.: SceneHGN: Hierarchical graph networks for 3D indoor scene generation with fine-grained geometry. IEEE TPAMI (2023)
24. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. In: CVPR (2022)
25. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR (2022)
26. Henderson, P., Subr, K., Ferrari, V.: Automatic generation of constrained furniture layouts. arXiv preprint arXiv:1711.10939 (2017)
27. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS (2017)
28. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation (1997)
29. Iyer, S., Konstas, I., Cheung, A., Zettlemoyer, L.: Summarizing source code using a neural attention model. In: ACL (2016)
30. Jiang, Y., Lim, M., Saxena, A.: Learning object arrangements in 3D scenes using human context. arXiv preprint arXiv:1206.6462 (2012)
31. Kar, A., Prakash, A., Liu, M.Y., Cameracci, E., Yuan, J., Rusiniak, M., Acuna, D., Torralba, A., Fidler, S.: Meta-sim: Learning to generate synthetic datasets. In: ICCV (2019)
32. Ke, G., He, D., Liu, T.Y.: Rethinking positional encoding in language pre-training. arXiv preprint arXiv:2006.15595 (2020)
33. Koh, J.Y., Agrawal, H., Batra, D., Tucker, R., Waters, A., Lee, H., Yang, Y., Baldridge, J., Anderson, P.: Simple and effective synthesis of indoor 3d scenes. In: AAAI (2023)
34. Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., Teh, Y.W.: Set transformer: A framework for attention-based permutation-invariant neural networks. In: ICML (2019)
35. Leimer, K., Guerrero, P., Weiss, T., Musialski, P.: Layoutenhancer: Generating good indoor layouts from imperfect data. In: SIGGRAPH Asia (2022)
36. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: ACL (2020)
37. Li, M., Patil, A.G., Xu, K., Chaudhuri, S., Khan, O., Shamir, A., Tu, C., Chen, B., Cohen-Or, D., Zhang, H.: GRAINS: Generative recursive autoencoders for indoor scenes. ACM TOG (2019)

38. Li, R., Allal, L.B., Zi, Y., Muennighoff, N., Kocetkov, D., Mou, C., Marone, M., Akiki, C., Li, J., Chim, J., et al.: StarCoder: May the source be with you! arXiv preprint arXiv:2305.06161 (2023)
39. Liu, J., et al.: CLIP-Layout: Style-consistent indoor scene synthesis with semantic furniture embedding. arXiv preprint arXiv:2303.03565 (2023)
40. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
41. Ma, R., Li, H., Zou, C., Liao, Z., Tong, X., Zhang, H.: Action-driven 3D indoor scene evolution. ACM TOG (2016)
42. Ma, R., Patil, A.G., Fisher, M., Li, M., Pirk, S., Hua, B.S., Yeung, S.K., Tong, X., Guibas, L., Zhang, H.: Language-driven synthesis of 3D scenes from scene databases. ACM TOG (2018)
43. Merrell, P., Schkufza, E., Li, Z., Agrawala, M., Koltun, V.: Interactive furniture layout using interior design guidelines. ACM TOG (2011)
44. OpenAI: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
45. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. In: NeurIPS (2022)
46. Para, W.R., Guerrero, P., Mitra, N., Wonka, P.: COFS: Controllable furniture layout synthesis. In: SIGGRAPH (2023)
47. Paschalidou, D., Kar, A., Shugrina, M., Kreis, K., Geiger, A., Fidler, S.: ATISS: Autoregressive transformers for indoor scene synthesis. NeurIPS (2021)
48. Patil, A.G., Patil, S.G., Li, M., Fisher, M., Savva, M., Zhang, H.: Advances in data-driven analysis and synthesis of 3D indoor scenes. Comput. Graph. Forum (2023)
49. Qi, S., Zhu, Y., Huang, S., Jiang, C., Zhu, S.C.: Human-centric indoor scene synthesis using stochastic grammar. In: CVPR (2018)
50. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
51. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. In: ICML (2019)
52. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: CVPR (2019)
53. Ritchie, D., Wang, K., Lin, Y.a.: Fast and flexible indoor scene synthesis via deep convolutional generative models. In: CVPR (2019)
54. Rubenstein, P.K., Asawaroengchai, C., Nguyen, D.D., Bapna, A., Borsos, Z., Quitry, F.d.C., Chen, P., Badawy, D.E., Han, W., Kharitonov, E., et al.: AudioPaLM: A large language model that can speak and listen. arXiv preprint arXiv:2306.12925 (2023)
55. Savva, M., Chang, A.X., Hanrahan, P., Fisher, M., Nießner, M.: PiGraphs: learning interaction snapshots from observations. ACM TOG (2016)
56. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. arXiv preprint arXiv:1803.02155 (2018)
57. Socher, R., Lin, C.C.Y., Ng, A.Y., Manning, C.D.: Parsing natural scenes and natural language with recursive neural networks. In: ICML (2011)
58. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., et al.: The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)

59. Tang, J., Nie, Y., Markhasin, L., Dai, A., Thies, J., Nießner, M.: DiffuScene: Scene graph denoising diffusion probabilistic model for generative indoor scene synthesis. arXiv preprint arXiv:2303.14207 (2023)
60. Tang, K., Zhang, H., Wu, B., Luo, W., Liu, W.: Learning to compose dynamic tree structures for visual contexts. In: CVPR (2019)
61. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS (2017)
62. Vinyals, O., Bengio, S., Kudlur, M.: Order matters: Sequence to sequence for sets. In: ICLR (2016)
63. Wang, K., Lin, Y.A., Weissmann, B., Savva, M., Chang, A.X., Ritchie, D.: PlanIT: Planning and instantiating indoor scenes with relation graph and spatial prior networks. ACM TOG (2019)
64. Wang, X., Yeshwanth, C., Nießner, M.: SceneFormer: Indoor scene generation with transformers. In: 3DV (2021)
65. Wei, Q.A., Ding, S., Park, J.J., Sajnani, R., Poulenard, A., Sridhar, S., Guibas, L.: LEGO-Net: Learning regular rearrangements of objects in rooms. In: CVPR (2023)
66. Weiss, T., Litteneker, A., Duncan, N., Nakada, M., Jiang, C., Yu, L.F., Terzopoulos, D.: Fast and scalable position-based layout synthesis. IEEE TVCG (2018)
67. Xu, K., Wu, L., Wang, Z., Feng, Y., Witbrock, M., Sheinin, V.: Graph2seq: Graph to sequence learning with attention-based neural networks. arXiv preprint arXiv:1804.00823 (2018)
68. Yu, L.F., Yeung, S.K., Tang, C.K., Terzopoulos, D., Chan, T.F., Osher, S.J.: Make It Home: automatic optimization of furniture arrangement. ACM TOG (2011)
69. Yu, L.F., Yeung, S.K., Terzopoulos, D.: The ClutterPalette: An interactive tool for detailing indoor scenes. IEEE TVCG (2015)
70. Zaheer, M., Kottur, S., Ravanbhakhsh, S., Póczos, B., Salakhutdinov, R., Smola, A.J.: Deep sets. In: NeurIPS (2017)
71. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. In: CVPR (2018)
72. Zhai, G., Örnek, E.P., Wu, S.C., Di, Y., Tombari, F., Navab, N., Busam, B.: CommonScenes: Generating commonsense 3D indoor scenes with scene graphs. arXiv preprint arXiv:2305.16283 (2023)
73. Zhao, X., Hu, R., Guerrero, P., Mitra, N., Komura, T.: Relationship templates for creating scene variations. ACM TOG (2016)
74. Zhao, Y., Lin, K., Jia, Z., Gao, Q.Q., Thattai, G., Thomason, J., Sukhatme, G.: Luminous: Indoor scene generation for embodied ai challenges. In: NeurIPSW (2021)
75. Zhou, Y., While, Z., Kalogerakis, E.: SceneGraphNet: Neural message passing for 3D indoor scene augmentation. In: CVPR (2019)