SemLayoutDiff: Semantic Layout Generation with Diffusion Model for Indoor Scene Synthesis

Xiaohao Sun[1] Divyam Goel[2] Angel X. Chang[1,3]
[1]Simon Fraser University [2]CMU [3]Alberta Machine Intelligence Institute (Amii)

https://3dlg-hcvc.github.io/SemLayoutDiff/

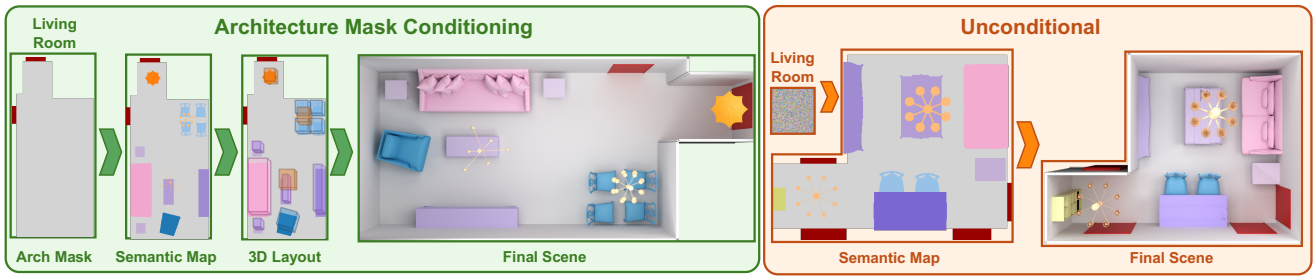

Figure 1. SemLayoutDiff generates 3D scenes conditioned on an architectural map or unconditionally. Left (full pipeline): With architectural conditioning and room type label, SemLayoutDiff synthesizes a 2D semantic layout map, predicts 3D attributes to form bounding box layouts, and retrieves objects to construct a final scene. Right: In the unconditional setting, SemLayoutDiff generates the architecture map and the semantic layout map from noise and room type label before synthesizing the scene (3D layout stage not shown).

Abstract

We present SemLayoutDiff, a unified model for synthesizing diverse 3D indoor scenes across multiple room types. The model introduces a scene layout representation combining a top-down semantic map and attributes for each object. Unlike prior approaches, which cannot condition on architectural constraints, SemLayoutDiff employs a categorical diffusion model capable of conditioning scene synthesis explicitly on room masks. It first generates a coherent semantic map, followed by a cross-attention-based network to predict furniture placements that respect the synthesized layout. Our method also accounts for architectural elements such as doors and windows, ensuring that generated furniture arrangements remain practical and unobstructed. Experiments on the 3D-FRONT dataset show that SemLayoutDiff produces spatially coherent, realistic, and varied scenes, outperforming previous methods.

1. Introduction

Automatic 3D indoor environment generation has many applications such as assisting content designers for AR/VR, video games, and serving as training data for computer vision models [31] and embodied AI agents [7]. Various approaches have been proposed for generating 3D indoor scenes. Typically, these methods separate the task into two steps: 1) generating a coarse layout specifying the semantic categories and positions of objects, and 2) retrieving and placing suitable objects based on the generated layout.

Early attempts used design guidelines [27] to determine the object arrangement. These approaches used handcrafted rules, limiting generalization to diverse types of scenes. Fisher et al. [11] introduced data-driven object placement. Since then, various deep learning techniques have been used for layout generation, including autoregressive models [29, 44, 46], graph neural networks [45], and diffusion over graphs [42]. More recently, LLMs have been used for open-vocabulary scene generation [1].

While LLMs are useful for providing priors on what objects are present and semantic relations between objects, they struggle to precisely place objects. Thus, researchers still actively investigate what can be learned via training on 3D scenes [19, 42]. Recent works trained on 3D scenes share common limitations: 1) room architecture is not handled, 2) interpenetration of objects, 3) lack of unified model that can be conditioned with different inputs (separate models typically trained for each room type).

To address the first two issues, we propose the use of a 2D top-down semantic map to represent the layout. In this semantic map, each cell represents one object category. By including architectural elements such as floors, doors, and windows, the representation also accommodates *generation of the room* as well as *layout of objects* in the room. The representation naturally ensures that objects do not overlap and floorplan constraints are properly maintained as shown in related work [37, 53]. Specifically, we use a categori-

cal diffusion model to generate the top-down semantic map. From the semantic map, we extract the object instances and their attributes (e.g., semantic category, size, orientation).

We also tackle training a unified model across room types that can be conditioned on the architecture and room type. We demonstrate that we can train a unified model that handles different room types and generates more plausible and realistic layouts. In summary, our contributions are:

- A novel scene representation using semantic layout maps with instance attributes, which enables simultaneous object placement and better captures spatial relationships
- A categorical diffusion model for scene synthesis by generating semantic maps, enabling a unified approach that efficiently handles diverse layouts across room types, captures architectural constraints and furniture relationships, and reduces out-of-bound and object intersection issues.
- Our model incorporates architectural elements by generating the room together with the objects, and conditioning on the architecture.

2. Related work

Rule-based and statistical prior indoor scene synthesis. Early work relied on placement constraints [48] and rules based on interior design principles [27] to place objects. Following these works, Fisher et al. [11] learned object arrangements from a 3D scene database by modeling the co-occurrence and spatial relationship of pairs of objects and constructing scenes hierarchically based on support.

Deep-learning based scene synthesis. Wang et al. [44], introduced auto-regressive scene generation and used CNNs for scene synthesis by representing scenes as 2D top-down images, with multiple channels encoding information such as floor layout, object semantic mask, etc. Followup work improved generation efficiency [32], and used transformers to decode the objects [28, 29, 40, 46]. Scenes were also modeled using graphs: a scene-hierarchy [15, 22], a relationship graph [25, 45, 51, 54], or a hybrid representation that combines scene-graphs with top-down image based representation [53]. Recent work [23, 39, 42] trains denoising diffusion model to generate the scene graph, with some work incorporating a floor plan as a conditioning signal [19, 26] and loss terms to avoid collisions [49].

We advocate the use of 2D top-down images as in Wang et al. [44]. Instead of autoregressively adding one object at a time, we use diffusion to layout all objects together. By working directly in image space, the model can ensure that the position of the objects is within the floor plan. Concurrent to our work, Su et al. [37] used diffusion-based models to generate a semantic layout conditioned on floor plan, but uses continuous instead of discrete categorical outputs.

LLM/VLM-based scene generation. As LLMs and VLMs capture common-sense knowledge about object placements in rooms, researchers developed frameworks that leverage LLMs to generate more open-world scenes based on text [2, 4, 13, 20, 50] and/or visual information [38]. While the use of LLMs/VLMs is a promising direction, we investigate whether we can train a unified model that generates reasonable semantic layouts using diffusion.

Combined layout and object generation. With advances in mesh generation, some works generate an entire scene mesh based on a input layout [9, 33] with Fang et al. [9] being able to generate the room layout via a separate layout stage. Other work generate individual objects for a specified layout in a compositional setting [6, 24, 30]. Our model can be used to specify the initial layout used by these approaches. Some works generate both the layout and the objects [14, 43, 51, 52], typically with a graph-based representation for the layout and then a shape generator for each object. Some [51, 52] still learn priors on the placement of objects from 3D datasets such as 3D-FRONT [12], while others [14] use an LLM to convert text to semantic scene-graphs, which are then used to generate relative positions of objects. We show in Fig. 8 that layouts generated by our model can also be combined with generated objects.

Diffusion models for layout generation. Diffusion has also been applied to 2D layout synthesis [17, 34–36]. Chai et al. [5] explore graphic layout generation with a newly designed transformer-based denoiser. Inoue et al. [21] formulate a discrete diffusion model for layout generation, which can solve diverse tasks via a single model using complex layout constraints. Another work [18] shows that using a multinomial diffusion model over categorical data, it is possible to generate 2D semantic maps. We take inspiration from diffusion for 2D layouts, and use diffusion to generate a 2D top-down semantic map that captures the spatial relationship between all objects.

3. Semantic Layout Representation

In recent scene generation work, the scene is represented as a sequence of objects [29] with corresponding attributes (category, location, etc.) or a scene graph [42] where nodes indicate objects and edges indicate object relations. Both approaches face the problem of overlapping objects and do not necessarily respect the floor boundary.

To tackle this problem, we represent the scene as a semantic map with exact instance annotation, shown in Fig. 2. The semantic map, denoted as $\mathcal{S} \in \mathbb{R}^{H \times W}$, represents the semantic segmentation of a top-down projection of a room, encoding both location and object category. Each pixel corresponds to fixed physical dimensions. We use a scale of $s = 0.01$ meters (i.e., a pixel is 0.01 meters). We denote the instance-level annotations as $\mathcal{I} = \{O_i\}$ where $i = 1, ..., N$, where N is the total number of objects within the room. Each instance O_i is defined by $O_i = \{c_i, s_i, p_i, r_i\}$, representing an object's category, size, position, and orientation. The category $c_i \in \{0, 1\}^K$ is a categorical variable over

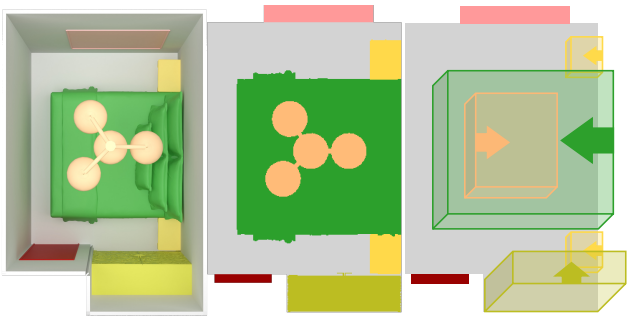

Figure 2. Semantic map representation example. We represent the scene in two parts: the 2D top-down semantic map with a fixed physical unit per pixel (middle) and the 3D bounding boxes with orientations for object-level attributes (right). Note that the left image is the corresponding rendered scene.

the total number of semantic categories K in the dataset, where $K = C + 4$. Here, C is the number of object types, and the additional categories represent architectural elements: `void` (outside room boundaries), `floor`, `door`, and `window`. Including floors, doors, and windows enables our model to generate the room together with the objects, while also handling architectural constraints. Our framework can be simplified to scenarios without doors and windows by reducing the semantic categories to $K = C + 2$, representing only `floor` and `void`. This simplified setting is similar to conditioning by floor as in prior work [19, 29]. However, our method simultaneously generates the room and object layouts, even under unconditional generation.

For each object instance O_i, in addition to the semantic category c_i, we also specify the bounding box size $s_i \in \mathbb{R}^3$, position $p_i \in \mathbb{R}^3$, and orientation r_i. For the orientation, we only consider the rotation of objects about the up (vertical) axis. From analyzing the orientation of objects in 3D-FRONT, we observed that over 97% of orientations are aligned with the coordinate axes. Thus we restrict our problem to predicting four distinct orientation classes and use a categorical variable $r_i \in \{0, 1, 2, 3\}$ to indicate the object's front direction corresponding to $0°, 90°, 180°, 270°$.

We process the scene data using BlenderProc [8] to render top-down views of the rooms, producing both semantic map and instance-level annotations. The camera parameters ensure consistent pixel-to-meter scaling. Images are padded to a fixed size (1200×1200, representing $12\text{m} \times 12\text{m}$). Finally, we apply the filtering strategy from ATISS [29] to ensure data quality. More details are in **??**.

4. Method

Unlike previous methods [19, 29, 42], our unified model generates all room types with a single model. It generates both the architecture and furniture objects at the same time in an unconditional way. Furthermore, it takes a *room mask* as input condition, and generates layouts conditioned on just the floormask or the full architecture mask (archmask for short). Our model has two stages: 1) a semantic layout diffusion model (Fig. 3a) for predicting the layout, and 2) an attribute prediction model (Fig. 3b). The two stages are trained separately and combined at the inference stage.

4.1. Semantic Layout Generation

The first stage is based on the multinomial diffusion model [18], shown in Figure 3 (a). The multinomial diffusion model is a categorical discrete diffusion model designed for categorical data perfectly suited for our data representation. We modify the multinomial diffusion model to make it a unified model conditioned on possible room masks for all room types.

During training, the input is a 2D segmentation map $\mathcal{S} \in \mathbb{R}^{H \times W \times K}$, where pixels indicate semantic ID. Each pixel value is a one-hot vector $\mathbf{x} \in \{0, 1\}^K$, where K is the number of semantic categories. We set $K = 38$ with 34 for object types, 3 for architecture elements (`floor`, `door`, `window`), and 1 for `void`. For conditioning, we specify the room type c_{room} and the room mask $\mathcal{A} \in \mathbb{R}^{H \times W}$ where $\mathcal{A}_{i,j} \in \{0, 1, 2, 3\}$ denotes `void`, `floor`, `door`, and `window`. Note that we can also condition with the floor (i.e., the room mask without doors or windows), which can be expressed as a binary mask, or do unconditional generation of both the room architecture and the objects.

We denote the noise pixel value at time step t as $\mathbf{x}_t \in \{0, 1\}^K$. If $\mathbf{x_t}$ belongs to category k, then $\mathbf{x}_{tk} = 1$ and $\mathbf{x}_{tj} = 0$ for $j \neq k$. The probability of $\mathbf{x_t}$ given $\mathbf{x_0}$ is

$$q(\mathbf{x_t}|\mathbf{x_0}) = \mathcal{C}(\mathbf{x_t}|\bar{\alpha}_t \mathbf{x_0} + (1 - \bar{\alpha}_t / K))$$

where $\mathcal{C}$ is the categorical distribution with parameters $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{\tau=1}^{t} \alpha_\tau$.

The objective of the multinomial diffusion model is to minimize the KL divergence of the categorical distribution between the generated data and ground truth, which is

$$L_{\text{MDM}} = \mathbb{E}_{\mathbf{x},t} \left[\text{KL} \left(q(\mathbf{x}_{t-1}|\mathbf{x_t}, \mathbf{x}_0), p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathcal{A}, c_{\text{room}}) \right) \right]$$

where $q(\mathbf{x}_{t-1}|\mathbf{x_t}, \mathbf{x}_0)$ is the ground-truth categorical posterior and $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathcal{A}, c_{\text{room}})$ is the predicted distribution at time $t-1$ given previous time t distribution, room mask and room type condition. See Hoogeboom et al. [18] for details of the multinomial diffusion model.

Figure 3 shows how the room mask $\mathcal{A}$ is passed through an embedding layer and an MLP to get the room mask embedding, then added to the $\mathbf{x_t}$ embedding to control denoising so the generated layout can respect the input mask. Furthermore, the room type c_{room} embedding is added to the timestep embedding to allow generating the desired room type. The embedding size for both the room mask and room type is 64. These two conditions allow a unified model for all room types that generates the room semantic layout with different room types and masks as input. By adding an-

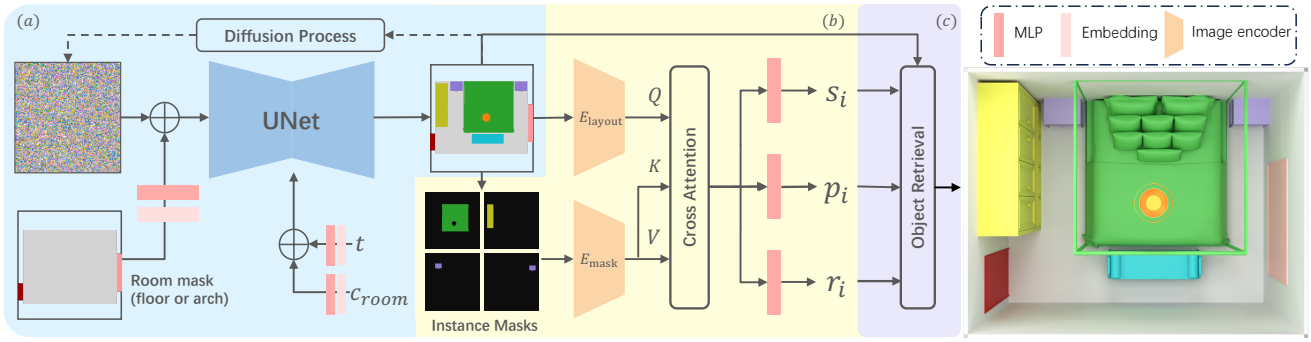

Figure 3. **SemLayoutDiff overview**. From left: (a) is the unified diffusion model that is conditioned on the room mask, and room type c_{room}. During the denoising process, the archmask or floormask embedding is added to the noise input embedding. The room type embedding is added to the timestep embedding. (b) is the object attribute prediction model with a semantic layout map as input. s_i, p_i, r_i indicate the ith instance's size, position, and orientation. At training time, we use ground-truth instance masks. During inference the semantic layout map is split into instance masks by using connected component analysis. The layout feature and the mask feature are passed to a cross-attention layer to get the final object instance feature, which is used to predict attributes. (c) During inference, objects are retrieved to match the category c_i and size s_i, and arranged using the position p_i and orientation r_i.

other control for the type of room mask used for conditioning (e.g., none, floor, arch), we can train a single unified model for all room and mask types (see **??**).

4.2. Attribute Prediction

The generated semantic layout only gives potential object instances and their 2D location on the projected plane. Thus, we design an attribute prediction model (APM) to predict attributes (s, p, r) for each object as shown in Figure 3 (b). Based on each instance mask, we obtain the 2D position and 2D size (width and length). The APM network then predicts the vertical size and position.

The input of the APM is the semantic map and the extracted instance masks, while the output is the instance-level object attributes, including vertical size $s_{y_i} \in \mathbb{R}^1$, vertical position $p_{y_i} \in \mathbb{R}^1$, and orientation $r_i \in \{0, 1, 2, 3\}$. We pass the semantic layout map and instance mask to the encoder E_{layout} and E_{mask} respectively to get the layout feature $f_{\text{layout}} \in \mathbb{R}^{128\times32\times32}$ and the instance mask feature $f_{\text{mask}} \in \mathbb{R}^{128\times32\times32}$. We treat f_{layout} as the query and f_{mask} as the key and value to perform cross-attention to obtain the final object instance feature $f_{\text{inst}} \in \mathbb{R}^{128\times32\times32}$. Finally, we use different prediction heads, consisting of a shared 2-layer MLP followed by a 1-layer MLP for each attribute, to predict attributes using f_{inst}. During training, we use the ground-truth instance masks. We use MSE loss for size and position heads, defined as L_s, L_p, and cross-entropy loss for orientation loss L_r. The total loss for the APM is a sum of the three losses: $L_{\text{APM}} = L_s + L_p + L_r$.

4.3. Inference

At inference, we follow the pipeline in Fig. 3 to generate the final scene (see **??** for details). **Semantic layout sampling.** Given the room mask $\mathcal{A}$ and room type c_{room} as conditions, we sample a semantic layout map $\mathcal{S}$ using the semantic layout diffusion model. **Instance extraction.** Based on the generated semantic map $\mathcal{S}$, we extract instances using connected component analysis. As there may be noisy pixels, we define category-specific size thresholds to filter out object instances that are too small. We determine the category-specific thresholds by calculating the minimum ratio of object pixels to total room pixels per object type. If an object's pixel ratio falls below the type-specific threshold, it is deemed invalid and excluded from subsequent attribute prediction and object retrieval. **Attribute prediction.** After we have both the semantic map and instance masks, we pass them to the attribute prediction model to predict attributes for each instance. **Room construction and object retrieval.** Lastly, we use the attributes to retrieve and place the objects. For unconditioned generation, we also construct the room based on the generated architecture mask.

5. Experiments

We compare our SemLayoutDiff with two recent diffusion methods (DiffuScene, MiDiffusion). See **??** for additional experimental and training details.

Experimental protocol. We use the experimental setup from prior work [29, 42], but we revisit the evaluation protocol and show that rendering choices can greatly influence common evaluation metrics for scene generation (**??**). We advocate for a specific set of choices (rendering using a unified color palette that groups semantically close objects together, with a floormask, and shows more object details).

Dataset. We train our model using the 3D-Front [12] training split from ATISS [29] with 4616 rooms containing bedrooms, living rooms, and dining rooms. We use 38 semantic classes (34 object types, 3 arch-element types, and `void`).

Baselines. We select two recent diffusion-based indoor scene synthesis methods [19, 42] with training code for comparison. We do not compare against CHoRD [37] as there is no code available, and report results for a pre-

Figure 4. Example textured synthesized scenes with unconditional generation. We generate the room architecture together with the placement of the objects.

trained PhyScene [49] model in **??**. Note that MIDIFFUSION was originally designed to support floormask conditioning, while DIFFUSCENE do not. In addition, both works trained separate models for each room type. For fair comparison, we add room-type conditioning to prior methods (see **??** for implementation details), and consider three generation modes for room mask conditioning: no room mask (*none*), conditioned on the *floor*, and *architecture*.

5.1. Examples of generated scenes

To demonstrate our model's ability to generate plausible room layouts, we show examples of generated rooms (objects and the room itself) in Fig. 4. We also compare generated scenes from our SEMLAYOUTDIFF to those generated by DIFFUSCENE and MIDIFFUSION with the three generation modes: no room mask (Fig. 5), conditioned on floor (appendix **??**), and on architecture (Fig. 6). For the comparison, we use semantically colored renderings with orthographic projection to clearly show the layout (see **??**). These are the same renderings used in the quantitative evaluation and user studies. We provide more qualitative results and renderings in **??**.

SEMLAYOUTDIFF can generate the architecture mask without conditioning. In Fig. 5, we show examples of different rooms generated by our SEMLAYOUTDIFF and other methods. As prior methods (DIFFUSCENE, MIDIFFUSION) cannot generate the architecture mask, we show the scenes with a square floor. In contrast, our proposed SEMLAYOUTDIFF effectively generates diverse and realistic arch masks (including floors, doors, and windows) even without explicit architectural conditioning. The scenes produced by SEMLAYOUTDIFF exhibit more plausible and varied room shapes and furniture placements, highlighting its capability to create coherent indoor scenes.

SEMLAYOUTDIFF respects the architecture mask better. From Fig. 6, we observe that DIFFUSCENE and MIDIFFUSION struggle to place furniture with respect to architectural elements, often resulting in objects blocking doors, windows, or extending beyond room boundaries. In contrast, our proposed SEMLAYOUTDIFF consistently generates coherent, organized, and realistic furniture arrangements, effectively respecting architectural constraints such as doors, windows, and floor boundaries.

Table 1. **Distribution match to ground-truth scenes.** We report the FID↓ , KID↓ , SCA% , and CKL↓ with different levels of architecture conditioning: unconditioned (None), conditioned on the floor (Floor), and the architecture map (Arch). **Bold** indicates best results. Our SEMLAYOUTDIFF outperforms other methods.

Condition	Method	FID↓	KID↓	SCA%	CKL↓
None	DIFFUSCENE [42]	125.46	88.09	99.75	24.70
	MIDIFFUSION [19]	100.54	41.27	97.35	28.31
	SEMLAYOUTDIFF (Ours)	**93.93**	**10.72**	**96.76**	**17.21**
Floor	DIFFUSCENE [42]	90.82	30.13	94.43	23.93
	MIDIFFUSION [19]	91.79	26.23	97.27	23.64
	SEMLAYOUTDIFF (Ours)	**81.79**	**14.52**	**89.83**	**5.99**
Arch	DIFFUSCENE [42]	88.47	30.22	95.02	20.42
	MIDIFFUSION [19]	93.51	31.06	95.28	36.97
	SEMLAYOUTDIFF (Ours)	**71.06**	**8.65**	**86.96**	**4.78**

Table 2. **Physical plausibility of generated scenes.** We compare out-of-bounds (OOB) ratios at the scene ($OOB_S\downarrow$) and object level ($OOB_O\downarrow$), as well as the collision rate (COL↓), and navigability (NAV↑). We do not include DIFFUSCENE and MIDIFFUSION for the `none` condition as they do not generate any room architecture, so most of the metrics are irrelevant.

Condition	Method	$OOB_S\downarrow$	$OOB_O\downarrow$	COL↓	NAV↑
None	SEMLAYOUTDIFF (Ours)	2.04	0.46	17.70	95.30
Floor	DIFFUSCENE [42]	69.93	28.92	37.68	94.99
	MIDIFFUSION [19]	70.17	31.83	35.92	96.02
	SEMLAYOUTDIFF (Ours)	**24.60**	**6.93**	**19.61**	**96.69**
Arch	DIFFUSCENE [42]	66.47	20.03	40.97	94.34
	MIDIFFUSION [19]	60.68	34.66	51.62	96.19
	SEMLAYOUTDIFF (Ours)	**16.63**	**6.62**	**16.13**	**96.36**

5.2. Quantitative evaluation

We evaluate the generated scenes by comparing against the distribution of real scenes, and evaluating the overall plausibility based on object collisions and how well the room mask is respected. For each condition, we generate 1000 scenes for evaluation. **Metrics.** For comparison of synthesized and real scenes, we follow ATISS [29] and report the KL divergence (CKL) between object category distributions, and Fréchet inception distance (FID) [16] and Kernel inception distance (KID) [3] against the test set. Following prior work, we report the CKL as CKL $\times 10^2$ and the KID as KID $\times 10^3$. We also report the scene classification accuracy (SCA), which scores how well a trained classifier can distinguish synthesized scenes from real-world scenes.

For SCA, the closer the score is to 50%, the more difficult it is for the classifier to distinguish between generated and real scenes. We find that in our experiments, the selection of objects is severely limited, causing the SCA to easily identify generated scenes (see **??**). For the view-based met-

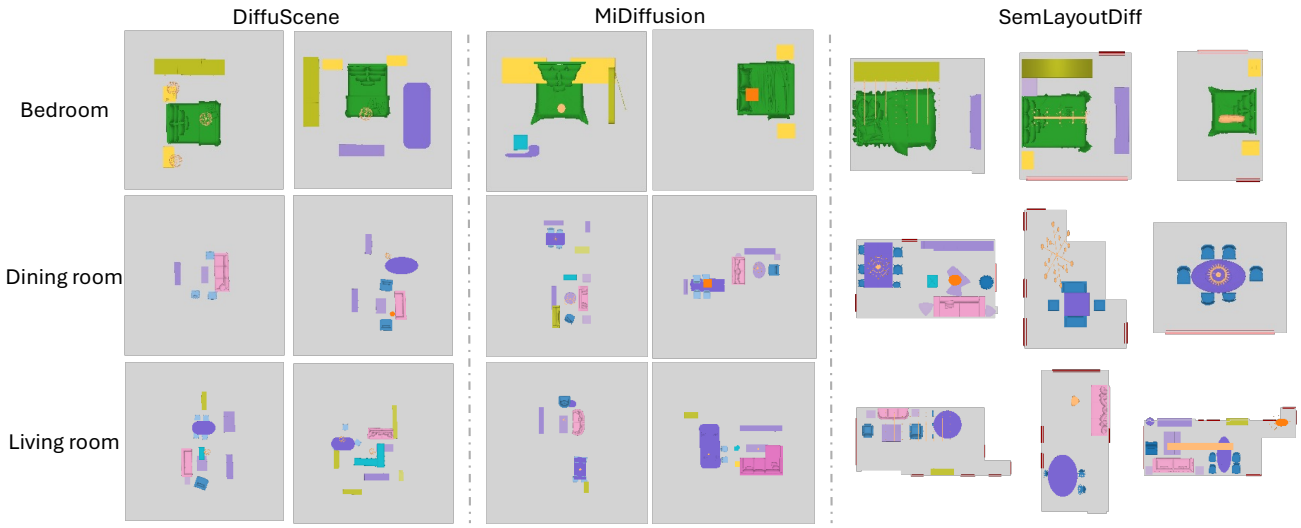

Figure 5. Comparison of scenes generated by prior methods and SEMLAYOUTDIFF *without room mask (i.e., no floor or arch) conditioning*. Since prior methods (left) *cannot* generate architectural elements, a square floor is used by default. Our method (right) generates feasible furniture placements aligned with its generated architectural layouts, leaving doors and windows unobstructed.

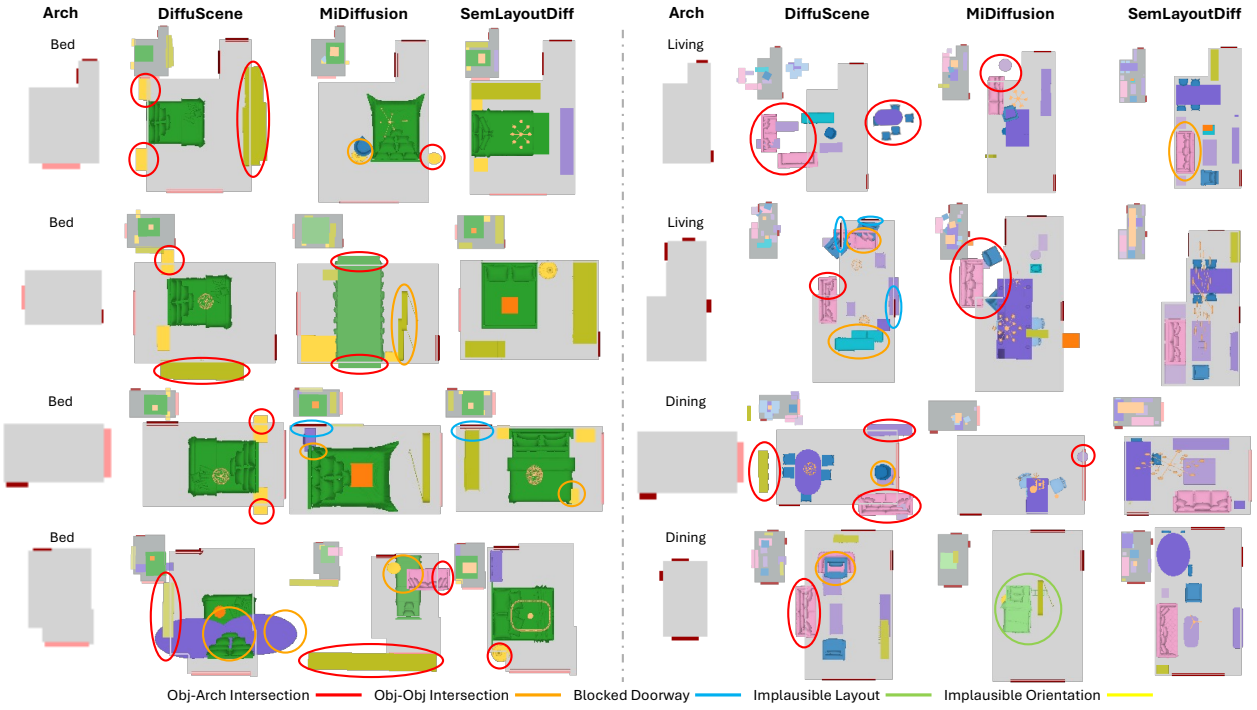

Figure 6. Comparison of generated scenes using different methods with *arch mask conditioning*. For each scene, we show the bounding box layout before object retrieval, inset on the upper left. Errors are indicated with color-coded circles (see Sec. 5.3 for error types). Our SEMLAYOUTDIFF generates scenes with fewer errors and respects architectural constraints by keeping furniture within room boundaries and maintaining clear spaces around doors (red) and windows (pink), whereas DIFFUSCENE and MIDIFFUSION do not.

rics (KID, FID, SCA), we use a top-down semantic-colored rendering that clearly shows the objects and includes the architectural elements (see **??**).

To assess scene plausibility, we report the Out-of-Bounds (OOB) ratio [10] (ratio of objects placed outside room boundaries). We report both the percentage of scenes (OOB_S) and the percentage of objects (OOB_O) with OOB issues. Finally, we report object collision rate (COL) and scene navigability (NAV) following SceneEval [41].

We report the performance of the different methods in Tabs. 1 and 2, where we see that our SEMLAYOUTDIFF consistently outperforms other unified methods.

SEMLAYOUTDIFF has a better distribution match to the ground-truth scenes. Compared to prior methods, SEM-

Table 3. Comparison of different training strategies for SEMLAYOUTDIFF: single condition (per-masktype) vs mixed conditions.

Condition	Training	Distribution				Plausibility			
		FID↓	KID↓	SCA%	CKL↓	OOB_S↓	OOB_O↓	COL↓	NAV↑
None	per-masktype	93.93	**10.72**	96.76	17.21	2.04	0.46	17.70	95.30
	mixed	**72.51**	11.27	**92.81**	**5.37**	**0.70**	**0.18**	**14.32**	**96.84**
Floor	per-masktype	81.79	14.52	89.83	5.99	24.60	6.93	19.61	**96.69**
	mixed	**76.24**	**10.60**	**86.04**	**4.38**	**21.50**	**5.07**	**14.67**	96.63
Arch	per-roomtype	107.99	42.99	96.08	16.57	26.97	8.89	23.11	95.11
	per-masktype	**71.06**	**8.65**	**86.96**	4.78	**16.63**	6.62	16.13	96.36
	mixed	73.61	9.85	92.19	**4.27**	21.53	**4.98**	**14.42**	**96.91**

Table 4. User rankings for the three methods on 40 samples.

Method	AR↓	Rank Distribution		
		1st%	2nd%	3rd%
DIFFUSCENE [42]	2.21	11.11	57.22	31.67
MIDIFFUSION [19]	2.55	8.47	28.33	63.19
SEMLAYOUTDIFF (Ours)	**1.25**	**80.42**	**14.44**	**5.14**

Table 5. Fine-grained user evaluation. Users were asked to assess errors found in generated scenes.

Method	Intersection		Blocked	Implausible	
	Arch↓	Obj↓	Door↓	Layout↓	Orientation↓
DIFFUSCENE [42]	89.38	38.12	41.25	45.00	25.62
MIDIFFUSION [19]	80.62	60.00	27.50	78.12	55.62
SEMLAYOUTDIFF (Ours)	**20.00**	**25.00**	**22.50**	**11.88**	**24.38**

LAYOUTDIFF maintains closer visual and object distribution (see lower FID, KID, and CKL in Tab. 1). The other methods have unrealistic object distributions, such as placing beds in living rooms or dining tables in bedrooms (e.g., in Fig. 6, the MIDIFFUSION-generated dining room at the bottom right has a bed in the middle; the DIFFUSCENE-generated bedroom at the bottom left has two dining tables). This issue arises because these models are trained on a mixed object distribution across room types, which can confuse learning. In contrast, our model treats scene layout generation as a semantic image synthesis problem, conditioned specifically by room type, thus facilitating more effective learning of distinct object distributions.

Conditioning on floor and arch mask improves performance. Table 1 shows that adding conditioning on the floor or arch mask significantly improves scene synthesis. The overall FID for SEMLAYOUTDIFF drops from 93.93 for no room mask to 81.79 with floor conditioning and 71.06 for arch conditioning. The drop in FID, KID, and CKL shows that architecture information enables more realistic layouts matching the ground-truth distribution.

SEMLAYOUTDIFF better respects architectural constraints and generates more plausible layouts. Table 2 shows that our SEMLAYOUTDIFF consistently achieves significantly lower scene-level (OOB_S) and object-level (OOB_O) out-of-boundary ratios compared to prior methods, across all room types. For instance, in bedrooms, SEMLAYOUTDIFF reduces the OOB_S from approximately 55% (DIFFUSCENE) and 60% (MIDIFFUSION) to 13.8%, and the OOB_O from over 20% to only 4%. These results demonstrate our method's superior ability to place furniture objects accurately within architectural boundaries.

SEMLAYOUTDIFF also consistently has the lowest object-object collisions and maintains high navigability scores. For navigability, the only slight drop appears in bedrooms, where our navigability metric treats lights positioned below 2 meters as obstacles; such fixtures are common in practice, so this penalty does not reflect a real limitation of the layout. Overall, the results show that our method generates navigable layouts with less collisions.

Mixed-condition training. In our experiments so far, we have trained a separate model for each room mask type (none, floor, arch). We can also train one single unified model that can handle conditioning both on room type as well as different room mask types by training a single mixed condition model (see **??** for details).

In Tab. 3, we show that with the mixed training, we generate scenes with slightly better distribution match and plausibility when conditioned on no room mask or just the floor mask. When conditioning on arch mask, the per-masktype model has slightly better distribution match. We also compare to training of a separate per-roomtype model. Notably, we find that per-roomtype training results in generated scenes that are further from the ground-truth distribution (high FID / KID / CKL) and poor plausibility (more out-of-bounds and collisions). The per-roomtype training suffers from both poor performance as well as the need for multiple models. By including all room types and mixing the conditioning allows the models to be trained on more data and learn the overall distribution better. To handle the different conditioning types, the per-roomtype training (as common in prior work) would need $M \times R$ separate models where R is the number of room-types and M is the number of mask types, vs M models for the per-masktype training, and 1 model for our proposed mixed condition training.

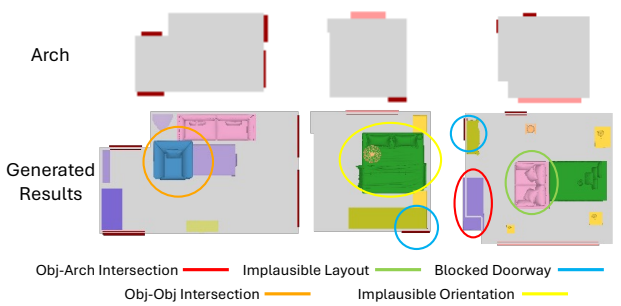

Figure 7. Examples of error cases for user study from SEMLAYOUTDIFF-generated scenes with arch mask conditioning.

5.3. User study

To further compare the quality of the generated results, we conducted two user studies. As prior methods cannot generate the architecture, for fairer comparison we focus on generation conditioned on the arch mask. The first study asked participants to rank three generated scenes conditioned on the same architectural mask with different models according to plan adherence and overall layout plausibility. In Tab. 4, we report average ranking (AR) and the frequency of 1st, 2nd, and 3rd places. The second study examined five specific error types: object-architecture intersection or out-of-bounds, object-object intersection, blocked doorway, implausible layout, and implausible orientation (Fig. 7). We report the percentage of scenes in which each error appears (Tab. 5). See **??** for more details on the user studies.

SEMLAYOUTDIFF has the best performance. As shown in Tab. 4, SEMLAYOUTDIFF achieved the best average ranking (AR=1.25), with users selecting it as the best 80% of the time, and only 5% ranking it last. In the fine-grained evaluation (Tab. 5), SEMLAYOUTDIFF also outperform DIFFUSCENE and MIDIFFUSION with users reporting lower issue rates in all 5 cases. Both user studies reinforces the findings from the automatic metrics: SEMLAYOUTDIFF *generates the most plausible scenes as judged by people*.

5.4. Discussion

Scene synthesis with generated object. In Fig. 8, we show that layouts from SEMLAYOUTDIFF can be converted into 3-D scenes by pairing them with a 3D object generation model. Specifically, we use TRELLIS [47], prompting it with "A `<category>`" to create a 3D object for each category in the layout. Then each mesh is scaled, translated, and rotated according to the predicted attributes.

Limitations. Despite the strong performance of our SEMLAYOUTDIFF in unified indoor scene synthesis, it has several limitations. First, our method does not leverage object shape or contextual information, which could improve attribute predictions, such as orientation, and enhance object consistency. Second, SEMLAYOUTDIFF currently does not support conditioning on text or a partial scene, limiting the level of user control. Third, relying on the small-scale 3D-FRONT dataset, which includes fewer than 5000 valid training rooms, constrains our model's potential; incorporating more diverse and extensive 2D and 3D scene data could enable the generation of more complex, multi-room layouts. Our two-stage design can also produce layout errors that are irrecoverable during attribute prediction. Finally, our top-down representation lacks explicit vertical information. Using 3D voxel grids or multi-height semantic maps, can allow for placement of objects at different heights.

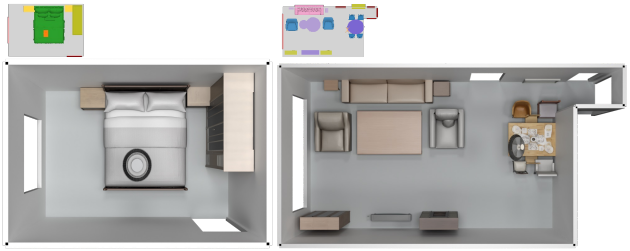

Figure 8. Example of our synthesized scenes with TRELLIS-generated objects. Top left is the top-down rendering with 3D-FUTURE retrieved objects using our STK rendering, and bottom is the TRELLIS-based rendering with Blender.

6. Conclusion

In this work, we represent the indoor scene as the 2D semantic layout map with instance attributes to better capture the spatial relationship between objects and generalize to different room types. With this representation, we propose a novel indoor scene synthesis model called SEMLAYOUTDIFF by leveraging the discrete denoising diffusion probabilistic model. Unlike prior work that trains models separately for different room types, we present a unified model that is trained across different room types, and can handle different generation modes. It can operate without any room mask, with floor mask conditioning, or with full architectural mask conditioning. The ability to incorporate architectural elements (doors and windows) beyond just the floor mask is particularly valuable for indoor scene synthesis, as these elements impose important constraints on furniture placement. Our approach also outperforms prior work in placing objects without intersection, and can better fit the objects within a input room mask.

Acknowledgments

This work was funded in part by a CIFAR AI Chair and NSERC Discovery Grants, and enabled by support from the Digital Research Alliance of Canada and a CFI/BCKDF JELF. We thank Ivan Tam for help with running SceneEval; Yiming Zhang and Jiayi Liu for suggestions on figures; Derek Pun, Dongchen Yang, Xingguang Yan, and Manolis Savva for discussions, proofreading, and paper suggestions. We also thank the anonymous reviewers for their feedback.

References

[1] Rio Aguina-Kang, Maxim Gumin, Do Heon Han, Stewart Morris, Seung Jean Yoo, Aditya Ganeshan, R Kenny Jones, Qiuhong Anna Wei, Kailiang Fu, and Daniel Ritchie. Open-universe indoor scene generation using LLM program synthesis and uncurated object databases. *arXiv preprint arXiv:2403.09675*, 2024. 1

[2] Tongyuan Bai, Wangyuanfan Bai, Dong Chen, Tieru Wu, Manyi Li, and Rui Ma. FreeScene: Mixed graph diffusion for 3D scene synthesis from free prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5893–5903, 2025. 2

[3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 5

[4] Ata Çelen, Guo Han, Konrad Schindler, Luc Van Gool, Iro Armeni, Anton Obukhov, and Xi Wang. I-design: Personalized LLM interior designer. *arXiv preprint arXiv:2404.02838*, 2024. 2

[5] Shang Chai, Liansheng Zhuang, and Fengying Yan. LayoutDM: Transformer-based diffusion model for layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18349–18358, 2023. 2

[6] Dana Cohen-Bar, Elad Richardson, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Set-the-scene: Global-local training for generating controllable NeRF scenes. *arXiv preprint arXiv:2303.13450*, 2023. 2

[7] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-scale embodied AI using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022. 1

[8] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Wendelin Knauer, Klaus H Strobl, Matthias Humt, and Rudolph Triebel. BlenderProc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. 3

[9] Chuan Fang, Yuan Dong, Kunming Luo, Xiaotao Hu, Rakesh Shrestha, and Ping Tan. Ctrl-Room: Controllable text-to-3D room meshes generation with layout constraints. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 692–701. IEEE, 2025. 2

[10] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. LayoutGPT: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 2023. 6

[11] Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. Example-based synthesis of 3D object arrangements. *ACM Transactions on Graphics (TOG)*, 31(6):1–11, 2012. 1, 2

[12] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3D-Front: 3D furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021. 2, 4

[13] Rao Fu, Zehao Wen, Zichen Liu, and Srinath Sridhar. AnyHome: Open-vocabulary generation of structured and textured 3D homes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 2

[14] Gege Gao, Weiyang Liu, Anpei Chen, Andreas Geiger, and Bernhard Schölkopf. GraphDreamer: Compositional 3D scene synthesis from scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[15] Lin Gao, Jia-Mu Sun, Kaichun Mo, Yu-Kun Lai, Leonidas J Guibas, and Jie Yang. SceneHGN: Hierarchical graph networks for 3D indoor scene generation with fine-grained geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8902–8919, 2023. 2

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[18] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021. 2, 3

[19] Siyi Hu, Diego Martin Arroyo, Stephanie Debats, Fabian Manhardt, Luca Carlone, and Federico Tombari. Mixed diffusion for 3D indoor scene synthesis. *arXiv preprint arXiv:2405.21066*, 2024. 1, 2, 3, 4, 5, 7

[20] Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A Ross, Cordelia Schmid, and Alireza Fathi. SceneCraft: An LLM agent for synthesizing 3D scene as Blender code. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024. 2

[21] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. LayoutDM: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10167–10176, 2023. 2

[22] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. GRAINS: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics (TOG)*, 38(2):1–16, 2019. 2

[23] Chenguo Lin and Yadong Mu. InstructScene: Instruction-driven 3D indoor scene synthesis with semantic graph prior. In *International Conference on Learning Representations (ICLR)*, 2024. 2

[24] Yiqi Lin, Haotian Bai, Sijia Li, Haonan Lu, Xiaodong Lin, Hui Xiong, and Lin Wang. CompoNeRF: Text-guided multi-object compositional NeRF with editable 3D scene layout. *arXiv preprint arXiv:2303.13843*, 2023. 2

[25] Andrew Luo, Zhoutong Zhang, Jiajun Wu, and Joshua B Tenenbaum. End-to-end optimization of scene layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3754–3763, 2020. 2

[26] Léopold Maillard, Nicolas Sereyjol-Garros, Tom Durand, and Maks Ovsjanikov. DeBaRA: Denoising-based 3D room arrangement generation. *Advances in Neural Information Processing Systems*, 2024. 2

[27] Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. Interactive furniture layout using interior design guidelines. *ACM transactions on graphics (TOG)*, 30(4):1–10, 2011. 1, 2

[28] Wamiq Reyaz Para, Paul Guerrero, Niloy Mitra, and Peter Wonka. COFS: Controllable furniture layout synthesis. In *ACM SIGGRAPH Conference Proceedings*, 2023. 2

[29] Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. ATISS: Autoregressive transformers for indoor scene synthesis. *Advances in Neural Information Processing Systems*, 34:12013–12026, 2021. 1, 2, 3, 4, 5

[30] Ryan Po and Gordon Wetzstein. Compositional 3D scene generation using locally conditioned diffusion. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2024. 2

[31] Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite photorealistic worlds using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12630–12641, 2023. 1

[32] Daniel Ritchie, Kai Wang, and Yu-an Lin. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6182–6190, 2019. 2

[33] Jonas Schult, Sam Tsai, Lukas Höllein, Bichen Wu, Jialiang Wang, Chih-Yao Ma, Kunpeng Li, Xiaofang Wang, Felix Wimbauer, Zijian He, et al. ControlRoom3D: Room generation using semantic proxy rooms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6201–6210, 2024. 2

[34] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2

[35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[36] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. 2

[37] Chong Su, Yingbin Fu, Zheyuan Hu, Jing Yang, Param Hanji, Shaojun Wang, Xuan Zhao, Cengiz Öztireli, and Fangcheng Zhong. CHOrD: Generation of collision-free, house-scale, and organized digital twins for 3D indoor scenes with controllable floor plans and optimal layouts. *arXiv preprint arXiv:2503.11958*, 2025. 1, 2, 4

[38] Fan-Yun Sun, Weiyu Liu, Siyi Gu, Dylan Lim, Goutam Bhat, Federico Tombari, Manling Li, Nick Haber, and Jiajun Wu. LayoutVLM: Differentiable optimization of 3D layout via vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2

[39] Kaifan Sun, Bingchen Yang, Peter Wonka, Jun Xiao, and Haiyong Jiang. RelTriple: Learning plausible indoor layouts by integrating relationship triples into the diffusion process. *arXiv preprint arXiv:2503.20289*, 2025. 2

[40] Qi Sun, Hang Zhou, Wengang Zhou, Li Li, and Houqiang Li. Forest2Seq: Revitalizing order prior for sequential indoor scene synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 2

[41] Hou In Ivan Tam, Hou In Derek Pun, Austin T Wang, Angel X Chang, and Manolis Savva. SceneEval: Evaluating semantic coherence in text-conditioned 3D indoor scene synthesis. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 2026. 6

[42] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. DiffuScene: Scene graph denoising diffusion probabilistic model for generative indoor scene synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 3, 4, 5, 7

[43] Alexander Vilesov, Pradyumna Chari, and Achuta Kadambi. CG3D: Compositional generation for text-to-3D via gaussian splatting. *arXiv preprint arXiv:2311.17907*, 2023. 2

[44] Kai Wang, Manolis Savva, Angel X Chang, and Daniel Ritchie. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 1, 2

[45] Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. PlanIT: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019. 1, 2

[46] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 106–115. IEEE, 2021. 1, 2

[47] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3D latents for scalable and versatile 3D generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 8

[48] Ken Xu, James Stewart, and Eugene Fiume. Constraint-based automatic placement for scene composition. In *Graphics Interface*, pages 25–34, 2002. 2

[49] Yandan Yang, Baoxiong Jia, Peiyuan Zhi, and Siyuan Huang. PhyScene: Physically interactable 3D scene synthesis for embodied AI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16262–16272, 2024. 2, 5

[50] Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3D embodied AI environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20–25. IEEE/CVF, 2024. 2

[51] Guangyao Zhai, Evin Pinar Örnek, Shun-Cheng Wu, Yan Di, Federico Tombari, Nassir Navab, and Benjamin Busam. CommonScenes: Generating commonsense 3D indoor scenes with scene graphs. *Advances in Neural Information Processing Systems*, 2023. 2

[52] Guangyao Zhai, Evin Pınar Örnek, Dave Zhenyu Chen, Ruotong Liao, Yan Di, Nassir Navab, Federico Tombari, and Benjamin Busam. EchoScene: Indoor scene generation via information echo over scene graph diffusion. In *European Conference on Computer Vision*, pages 167–184. Springer, 2024. 2

[53] Zaiwei Zhang, Zhenpei Yang, Chongyang Ma, Linjie Luo, Alexander Huth, Etienne Vouga, and Qixing Huang. Deep generative modeling for scene synthesis via hybrid representations. *ACM Transactions on Graphics (TOG)*, 39(2):1–21, 2020. 1, 2

[54] Yang Zhou, Zachary While, and Evangelos Kalogerakis. SceneGraphNet: Neural message passing for 3D indoor scene augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7384–7392, 2019. 2