Learning 3D Scene Priors with 2D Supervision

Yinyu Nie[1]   Angela Dai[1]   Xiaoguang Han[2]   Matthias Nießner[1]

[1]Technical University of Munich   [2]The Chinese University of Hong Kong (Shenzhen)

Abstract

Holistic 3D scene understanding entails estimation of both layout configuration and object geometry in a 3D environment. Recent works have shown advances in 3D scene estimation from various input modalities (e.g., images, 3D scans), by leveraging 3D supervision (e.g., 3D bounding boxes or CAD models), for which collection at scale is expensive and often intractable. To address this shortcoming, we propose a new method to learn 3D scene priors of layout and shape without requiring any 3D ground truth. Instead, we rely on 2D supervision from multi-view RGB images. Our method represents a 3D scene as a latent vector, from which we can progressively decode to a sequence of objects characterized by their class categories, 3D bounding boxes, and meshes. With our trained autoregressive decoder representing the scene prior, our method facilitates many downstream applications, including scene synthesis, interpolation, and single-view reconstruction. Experiments on 3D-FRONT and ScanNet show that our method outperforms state of the art in single-view reconstruction, and achieves state-of-the-art results in scene synthesis against baselines which require for 3D supervision. Project page: https://yinyunie.github.io/sceneprior-page/

1. Introduction

Understanding the geometric structures in real world scenes takes significant meaning in 3D computer vision. In recent years, many prominent approaches have been proposed to understand 3D scenes for different applications, e.g., scene synthesis [24, 28, 36, 48, 49, 52] and semantic reconstruction [6, 15, 16, 18–21, 26, 30, 34, 35, 56]. These methods have shown initial promise in estimating object layouts and shapes in 3D scenes, from RGB or RGB-D input data.

However, such methods have focused on leveraging full 3D supervision for layout and shape learning, in task-specific fashion. Such ground truth 3D scene geometry with semantic decomposition to objects and structures is difficult and expensive to acquire, requiring significant artist modeling efforts [13] or expert annotations [1] of CAD modeling. In contrast, 2D image and video acquisition is significantly
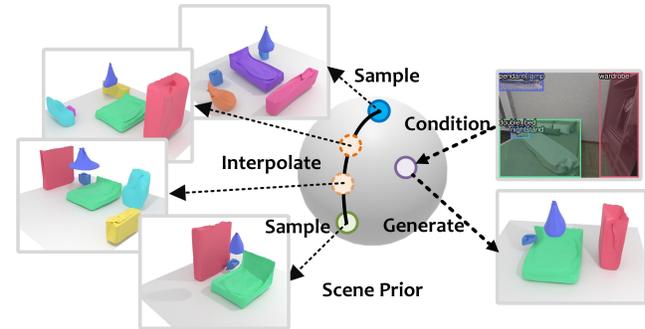

Figure 1. We learn 3D scene priors with 2D supervision. We model a latent hypersphere surface to represent a manifold of 3D scenes, characterizing the semantic and geometric distribution of objects in 3D scenes. This supports many downstream applications, including scene synthesis, interpolation and single-view reconstruction.

more accessible. We thus propose to instead learn a general 3D scene prior from only multi-view 2D information. This approach takes a unified perspective to 3D scene understanding: our 3D scene prior is learned in a generalizable fashion from only multi-view 2D information, enabling application to multiple downstream tasks, including scene synthesis and single-view reconstruction.

In order to understand geometric 3D scene structure from only multi-view RGB images, we must bridge the domain gap between 3D geometry and RGB colors. To this end, we propose to leverage the 2D semantic domain: we learn our 3D scene prior by ensuring that our learned 3D scene structures and objects, when differentiably rendered, match the 2D semantic distribution of estimated 2D instance segmentations from the RGB images. This enables supervision of 3D object and layout structures guided by 2D information.

Our 3D scene prior is learned by optimizing for a mapping from a latent space (parameterized as a hypersphere surface) to a manifold of 3D scenes, as shown in Fig. 1. Each random vector sampled from the latent space is decoded into a sequence of objects autoregressively using our permutation-invariant transformer. Each output object is characterized by their class category, 3D bounding box and mesh. By training this decoder, we encode the scene prior

into a latent space, sampling on which enables us to synthesize plausible 3D scenes. With such formulation, our method supports many other downstream tasks on different conditions, e.g., scene synthesis, interpolation and single-view reconstruction.

In summary, we present our contributions as follows:

- We present a new perspective on general 3D scene prior learning with only 2D supervision. We learn the prior distribution of both semantic object instance layouts and shapes via mapping from a latent space to the 3D scene space, by training from 2D multi-view data.
- We formulate scene prior learning as an autoregressive sequence decoding problem, and propose a novel permutation-invariant transformer corresponding with a 2D view loss to map a latent vector from a hypersphere surface to a sequence of semantic instances characterized by their class categories, 3D bounding boxes and meshes.
- Our learned 3D scene prior supports many downstream tasks: single-view scene reconstruction, scene synthesis and interpolation. Experiments on 3D-FRONT and ScanNet show that our method matches state-of-the-art performance in scene synthesis against methods requiring 3D supervision, and outperforms state of the art in single-view reconstruction.

2. Related Work

3D Semantic Scene Synthesis 3D semantic scene synthesis has received extensive exploration in recent decades, particularly following the popularity of 3D indoor scene datasets and 3D deep learning. Given a collection of synthesized or real 3D scenes, many methods have been proposed to learn the distribution of object layouts to synthesize unseen but plausible 3D scene arrangements.

Traditional methods for scene synthesis usually formulate the task as data-driven object placement [57], where a scene with object instances is represented as a graph [3, 4, 39, 59], or is connected to human activities to build a human-centric representation [12, 14, 31, 39]. Scene priors are explicitly hard-coded based on design guidelines [32, 54], represented by frequency statistics (e.g., objects co-occurrence map) [3, 4, 10], inferred by affordance map from human activities [12, 14, 22], or learned from arrangement examples in 3D scenes [11, 14]. Given an initial scene [14, 32, 54], a 2D sketch [51] or a text [3, 4], several methods synthesize object placements under the guidance of scene priors with different optimization strategies, e.g., manual interactions [4, 32, 42], iterative methods [12, 14, 29, 50] or non-linear optimization [3, 4, 11, 39, 51, 54, 55], to maximize a plausibility score or a joint probability.

With deep learning techniques, many methods instead learn scene priors with neural networks. They keep the problem formulation and use recurrent networks [29, 36, 41, 47–49], feed-forward models [33, 58], GANs [53], or VAEs [38, 52] to synthesize object poses recursively or in one stage from an initial scene or random noise, and usually require 3D information for training. In contrast, our method directly learns the joint semantic and geometric distribution of instances in scenes with 2D supervision, which enables a general scene priors that supports semantic scene synthesis from a latent random vector.

Single-view Scene Reconstruction Single image reconstruction methods have also been designed around various task-specific, data-driven scene priors [6, 15, 16, 18, 20, 21, 25–27, 30, 34, 37, 45, 56]. Izadinia et al. [21] propose a deep learning-based method to retrieve and align CAD models to an input image. Further methods were developed with a similar top-down approach to reconstruct a scene by detection and retrieval, in which [20, 21, 25] use a render-and-compare strategy. [18, 26, 27] improve the offline shape retrieval with an end-to-end 2D-3D shape matching with pose alignment. Other works focus on shape reconstruction instead of retrieval, where object shapes are learned as meshes [15, 16, 34], volume grids [37, 45], or implicit functions [30, 30, 56] with the supervision of paired 3D object shapes or multi-view images. However, modeling scenes with a top-down manner usually focuses on per-object perception without any global scene reasoning. In contrast, we focus on learning prior distribution of 3D scenes, to generate plausible object layouts conditioned on an input image.

3. Methodology

Our work aims to learn the prior distribution of 3D semantic scenes. We formulate this problem by learning a mapping f from a parameterized latent space $\mathcal{Z}$ to the scene space $\mathcal{S}$, and hope each sample $z \in \mathcal{Z}$ corresponds to a reasonable 3D scene $s \in \mathcal{S}$ as in

$$s = f(z); \ z \in \mathcal{Z}, \ s \in \mathcal{S}. \tag{1}$$

In Eq. 1, there are four key components: the parameterization of latent space $\mathcal{Z}$ and scene space $\mathcal{S}$, the architecture of f, and the losses to train f. We illustrate our approach in Fig. 2, the latent space $\mathcal{Z}$ is represented as a unit hypersphere surface, each latent vector z sampled from which conditions the generation of an object sequence with our permutation-invariant transformer. We formulate a 3D scene $s \in \mathcal{S}$ as a set of objects with their class categories, positions and shapes, thus each object from the transformer is decoded into a category label, a 3D bounding box and a mesh with a completeness score indicating whether the generated scene is complete or not. Training our network only needs 2D supervision. We design a 2D view loss between the renderings of generated scenes and the ground-truth instance masks under respective camera views.
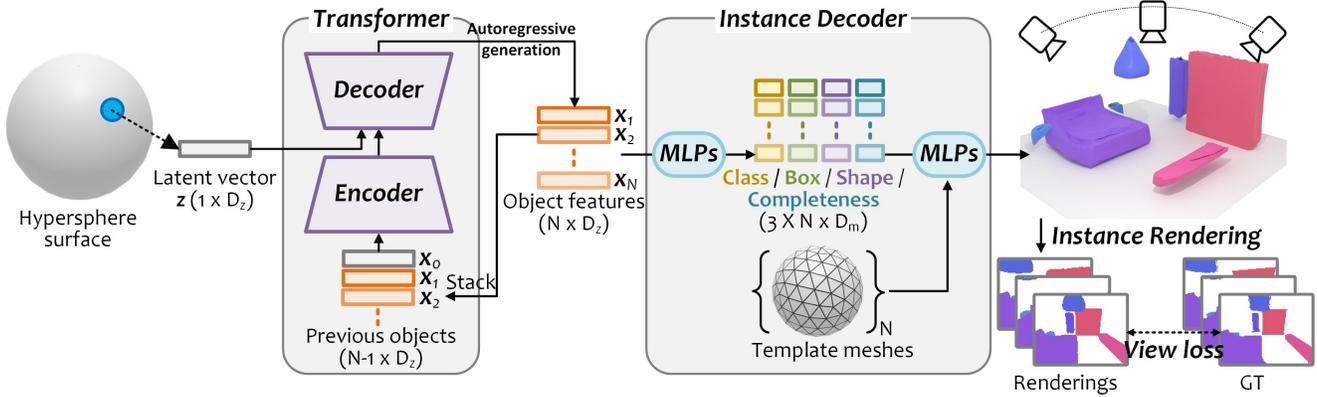
Figure 2. Overview of our approach. Our method represents each scene as a learnable latent vector z on a hypersphere. To build mapping from the latent surface to the semantic scene space, we propose a permutation-invariant transformer to generate objects $\{x_k\}$ autoregressively conditioned on previous objects and z. For each object feature, we regress its object class label, 3D bounding box, shape feature and completeness score, which respectively represent the object layout, geometry, and the completeness of the output scene. We do not require 3D supervision, and instead leverage differentiable rendering with multi-view 2D instance masks supervision.

3.1. Latent Space Parameterization

We parameterize the latent space $\mathcal{Z}$ as a unit hypersphere surface. We choose a spherical surface as the latent representation because it does not have strict boundaries (e.g, uniform distribution), and sampling on it will not produce outliers (e.g., gaussian distribution) [8], which guarantees the continuity when moving from one latent vector to another. We formulate the sampling process from $\mathcal{Z}$ as

$$z = \frac{\sum_{i=1}^{M} w_i \cdot \phi_i}{|| \sum_{i=1}^{M} w_i \cdot \phi_i ||_2}; \ \sum_{i=1}^{M} w_i = 1, w_i > 0 \quad (2)$$

where $\{\phi_i | \phi_i \in \mathbb{R}^{D_z}, ||\phi_i||_2 = 1\}$ is a set of M fixed vectors with unit length, which uniformly spreads on the sphere surface. D_z is the dimension of the hypersphere. Therefore, by sampling a set of weights $\{w_i\}$, we can obtain a latent vector $z \in \mathbb{R}^{D_z}$ for scene generation. Note that both $\{w_i\}$ and our network f are learnable during training.

3.2. Permutation-invariant Transformer

We generate a sequence of objects in a scene, conditioned on the latent vector z, in an autoregressive fashion. New objects are generated conditioned on the previous ones, which can be formulated as to estimate the distribution of

$$P_\theta\left(x_1, ..., x_N | z\right) = P_\theta\left(x_1 | z\right) \prod_{k=2}^{N} P_\theta\left(x_k | x_1, ..., x_{k-1}; z\right), \quad (3)$$

where x_k denotes the k-th object feature. N is the maximal number of objects to be generated. θ is the trainable network. Every new object x_k is generated conditioning on $x_1, ..., x_{k-1}$ and z.

We observe that objects in a scene are an unordered set, where each permutation of an object sequence should be equivalent to each other, e.g., *[chair, table, lamp]* should be equivalent to *[lamp, chair, table]* given the three objects. Thus, the generation of the next object x_k should be independent to the order of previous objects $\{x_1, ..., x_{k-1}\}$, which can be formulated as

$$P_\theta\left(x_k | x_{p_1}, ..., x_{p_{k-1}}; z\right) = P_\theta\left(x_k | x_{q_1}, ..., x_{q_{k-1}}; z\right)$$
$$1 \leq p_i, q_i \leq k - 1 \quad \text{and} \quad p_i \neq p_j, q_i \neq q_j, \ \forall \ i \neq j. \quad (4)$$

This requires our generation network to be permutation-invariant to the input objects. We model this step using a transformer backbone [23], which is illustrated in Fig. 3. Inspired by [36, 46] where removing the positional encoding in a transformer encoder makes the network agnostic to the object order, we consider this design and encode permutation-invariant scene context $F_k \in \mathbb{R}^{k \times D_z}$ from previous objects $\{x_1, ..., x_{k-1}\}$. Note that x_0 in Fig. 2 and 3 is a learnable start token which indicates an empty room.

F_k contains attention signals between object pairs and the scene context feature. We use F_k as the keys and values, and the latent vector z as the query to infer the next object based on the scene context. We adopt a transformer decoder for object inference, where we also do not use positional encoding. Afterwards, it outputs the next possible object $x_k \in \mathbb{R}^{D_z}$. We forward this step autoregressively to obtain an object feature sequence $\{x_1, ..., x_N\}$.

3.3. Layout and Shape Decoder

In this section, we decode object features $\{x_k\}$ into a set of object categories, 3D bounding boxes and shapes.

Scene Parameterization We focus on indoor scenarios, and represent a 3D scene in a world coordinate system with the floor center located at the origin. Each object inside
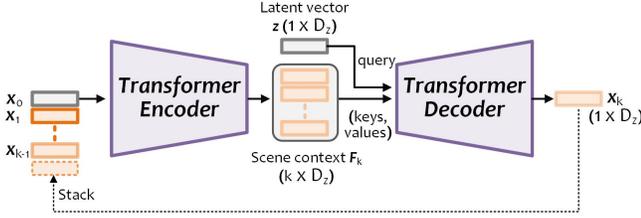
Figure 3. Transformer for autoregressive object generation. From the start token $\boldsymbol{x}_0$, we design a permutation-invariant transformer to autoregressively generate a new object feature $\boldsymbol{x}_k$ by querying the encoded scene context $\boldsymbol{F}_k$ with the latent vector $\boldsymbol{z}$.

is characterized by its class label $\boldsymbol{l} \in \mathbb{L}$, axis-aligned 3D bounding box with size $\boldsymbol{s} \in \mathbb{R}^3$ and center $\boldsymbol{c} \in \mathbb{R}^3$, and 3D mesh $\boldsymbol{M} \in \mathbb{M}$ in the canonical system, where $\mathbb{L}$ denotes all class categories and $\mathbb{M}$ is the shape space.

Layout Decoder We adopt a layout decoder to estimate the class category $\boldsymbol{l}_k$, box center $\boldsymbol{c}_k$, size $\boldsymbol{s}_k$ and completeness score $\boldsymbol{p}_k$ from each object feature $\boldsymbol{x}_k$, which is implemented by two MLPs respectively:

$$\{\boldsymbol{l}_k, \boldsymbol{c}_k, \boldsymbol{s}_k, \boldsymbol{p}_k\} = \text{MLP}_2(\text{MLP}_1(\boldsymbol{x}_k)), \ k = 1, ..., N. \quad (5)$$

The completeness $\boldsymbol{p}_k$ is a binary score of whether $\boldsymbol{x}_k$ is the last object or not, indicating whether we should continue object generation.

Shape Decoder We estimate object shapes by deforming a template sphere from $\boldsymbol{x}_k$. Inspired by [17], we use MLPs to learn the offsets from vertices $\boldsymbol{V} \in \mathbb{R}^{|\boldsymbol{V}| \times 3}$ on the template sphere to the target surface by

$$\Delta \boldsymbol{v}_k = \text{MLP}_3(\text{Concat}[\text{MLP}_1(\boldsymbol{x}_k), \boldsymbol{v}]), \boldsymbol{v} \in \boldsymbol{V}, \quad (6)$$

where $\boldsymbol{V}$ denotes the set of vertices on the template; MLP_3 is an point-wise MLP, and we share MLP_1 with Eq. 5. Afterwards, the vertices of the k-th object mesh $\boldsymbol{M}_k$ is predicted as $\boldsymbol{S}_k = \{\boldsymbol{v} + \Delta \boldsymbol{v}_k\}$. Note that $\boldsymbol{M}_k$ is in a canonical system, which is transformed to the world system with the bounding boxes in Eq. 5 by $\boldsymbol{o}_k = \boldsymbol{s}_k \cdot \boldsymbol{M}_i + \boldsymbol{c}_k$. Finally, we obtain $\{\boldsymbol{o}_k\}$ as the set of output objects in a scene.

3.4. View Loss

Instead of using 3D data, our method leverages multi-view posed instance masks with tracks to supervise layout and shape generation. We consider T views in a 3D scene with n tracked objects $\{\boldsymbol{o}_j^{gt}\}$, $j=1,...,n$. Each object $\boldsymbol{o}_j^{gt}$ is observable in T_j views, where $T_j \subseteq T$. We denote $\boldsymbol{m}_j^p$ as the mask of object $\boldsymbol{o}_j^{gt}$ observed in the view $p \in T_j$. For each mask $\boldsymbol{m}_j^p$, there are a class label $\boldsymbol{l}_j^{gt}$ and camera parameters cam^p.

We train our model with a set of view losses between the ground-truth masks and the renderings of generated objects. We illustrate our training strategy in Fig. 4, which
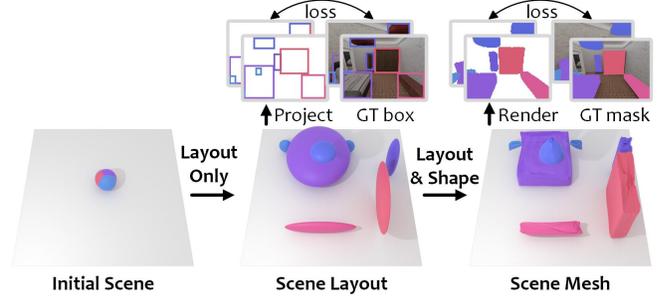

Figure 4. Optimization strategy for layout and shape training. We train layout learning and mesh deformation end-to-end with two stages to guarantee that objects are deformed after alignment.

is trained end-to-end with two stages: **1) layout pretraining** and **2) layout+shape joint learning**. We pretrain the layout first because 1) shape deformation from 2D requires an ideal alignment between its renderings and ground-truth masks; 2) differentiable rendering is not required in the first stage thus the efficiency is largely improved.

Layout Loss Our network is initialized with all objects $\{\boldsymbol{o}_k\}$ at the floor center. In the first stage, we learn the object layout without shape decoding (i.e., $\Delta \boldsymbol{v}_k = 0$ in Eq. 6). Only object categories, centers, sizes and completeness scores are optimized ($\{\boldsymbol{l}_k, \boldsymbol{c}_k, \boldsymbol{s}_k, \boldsymbol{p}_k\}$). Therefore, we define the layout loss $\mathcal{L}_{layout}$ as the combination of 1) classification loss $\mathcal{L}_l$; 2) 2D bounding box loss $\mathcal{L}_{box}$; 3) completeness loss $\mathcal{L}_p$, and 4) frustum loss $\mathcal{L}_f$.

For an object prediction $\boldsymbol{o}_k$, we project the vertices on its mesh to all views in T and calculate the 2D bounding boxes $\boldsymbol{B}_k = \{\boldsymbol{b}^1, ..., \boldsymbol{b}^T\}_k$. For each ground-truth object $\boldsymbol{o}_j^{gt}$, we also have its 2D bounding boxes $\boldsymbol{B}_j^{gt} = \{\boldsymbol{b}^1, ..., \boldsymbol{b}^{T_j}\}_j^{gt}$ in T_j views. We let $\boldsymbol{l}_k$ as the predicted class label of $\boldsymbol{o}_k$, and $\boldsymbol{l}_j^{gt}$ as the ground-truth label of $\boldsymbol{o}_j^{gt}$ respectively. For each prediction pair $(\boldsymbol{B}_k, \boldsymbol{l}_k)$, we adopt the Hungarian algorithm [2, 43] to find its optimal bipartite matching $(\boldsymbol{B}_{\sigma(k)}^{gt}, \boldsymbol{l}_{\sigma(k)}^{gt})$ from the ground truth. Therefore we can obtain the matches from each prediction $\boldsymbol{o}_k$ to a ground-truth $\boldsymbol{o}_{\sigma(k)}^{gt}$, which is the basis of our loss functions. For details of Hungarian matching, we refer to the supplemental.

Based on Hungarian matching, our layout loss can be formulated as

$$\mathcal{L}_L = \lambda_l \mathcal{L}_l + \lambda_{box} \mathcal{L}_{box} + \lambda_p \mathcal{L}_p + \lambda_f \mathcal{L}_f, \quad (7)$$

where $\mathcal{L}_l$ is the cross-entropy loss between $\boldsymbol{l}_k$ and $\boldsymbol{l}_{\sigma(k)}^{gt}$ for object classification; $\mathcal{L}_{box}$ is the average L_1 distance between 2D bounding boxes in $\boldsymbol{B}_k$ and $\boldsymbol{B}_{\sigma(k)}^{gt}$ on each view of $T_{\sigma(k)}$; $\mathcal{L}_p$ is the binary cross entropy loss to judge whether $\boldsymbol{o}_k$ is the last object or not. For objects whose initial position is not located in a view frustum of their matched ground-truth, we design a frustum loss $\mathcal{L}_f$ to enforce their 3D center $\boldsymbol{c}$ moving into the target frustum. Further details can be found in the supplemental.

Shape Loss After the layout loss converges, we switch on the shape decoder in Eq. 6 and use 2D instance masks $\{\boldsymbol{m}_j^p\}$ for mesh deformation. In addition to 2D bounding boxes, we also render the soft silhouettes of each $\boldsymbol{o}_k$ on all views T, by a soft silhouette shader from [40]. The shape loss of $\boldsymbol{o}_k$ is defined as a multi-view mask loss as

$$\mathcal{L}_S = \sum_{p \in T_{\sigma(k)}} \text{BCE}(\boldsymbol{r}_k^p, \boldsymbol{m}_{\sigma(k)}^p) \mathbb{1}[\text{IoU}(\boldsymbol{r}_k^p, \boldsymbol{m}_{\sigma(k)}^p) > 0.5], \tag{8}$$

where $\boldsymbol{r}_k^p$ is the rendering of $\boldsymbol{o}_k$ in view $p \in T_{\sigma(k)}$; $\boldsymbol{m}_{\sigma(k)}^p$ is the corresponding mask of $\boldsymbol{o}_{\sigma(k)}^{gt}$. For pairs whose IoU larger than 0.5, we use binary cross entropy to calculate the mask loss, which is inspired by [16] that we only use those well-aligned masks for supervision. We optimize $\mathcal{L}_S$ to improve object mesh deformation.

Finally, we train the our network end-to-end with $\mathcal{L} = \mathcal{L}_L + \lambda_S \mathcal{L}_S$. All losses above are averaged across all objects. $\{\lambda\}$ are weights that balance the losses.

4. Experiment Setup

Datasets and Tasks We train and test our method on two downstream tasks: 3D scene synthesis and single-view scene reconstruction.

For scene synthesis, we use the 3D-FRONT [13] dataset for training and evaluation as in [36, 52]. 3D-FRONT is a synthetic indoor scene dataset with 18,968 rooms populated with 3D furniture objects, wherein we choose bedrooms and living rooms for our training and testing. We use [9] to sample virtual camera poses in each room and render the 2D instance masks for objects under different views, which produces 102,268 views in 3,922 bedrooms ($\approx$5.53 objects in each room), and 80,997 views in 1,496 living rooms ($\approx$9.82 objects in each room). For each room type, the train/test split ratio is 9:1 over rooms. We provide additional rendering details in the supplemental.

For single-view scene reconstruction, we use the Scan-Net dataset [7] for benchmarking. ScanNet is a real indoor scene dataset with 1,513 rooms in 21 room types. It provides 3D scans with semantic instance labels, as well as 2D scanning data where each room is captured into an RGB video sequence, and each frame contains a calibrated camera pose and 2D instance masks with tracking IDs. We uniformly sample up to 100 frames on each video and obtains 108,059 valid frames from all scenes. We use the official train/val split for our training and evaluation.

Implementation For each task, we train our network on 4 NVIDIA 2080 Ti GPUs with batch size at 16. Each sample in a batch contains a scene with 20 randomly sampled views. We augment each scene in 3D-FRONT and ScanNet with axis-aligned rotations by 90, 180, 270 degrees. As in Sec. 3.4, we train our network end-to-end with two stages. We first train the layout learning in first 800 epochs, followed by the shape learning with 500 epochs. We use Adam

as the optimizer with initial learning rate 1e-4, which is decayed by $0.1\times$ after 300 epochs, in both of the two stages. The losses are weighted by $\lambda_l = \lambda_p = \lambda_f = 1$, $\lambda_{box} = 5$, $\lambda_S = 2$ to balance loss values. For layer and differentiable rendering specifications, we refer to the supplemental.

For scene synthesis, we randomly sample latent vectors by Eq. 2 to synthesize diverse scenes. For single-view reconstruction, we freeze the generation network and take the single-view instance masks as the ground-truth to optimize the latent vector with our loss function. After it converges, we generate the 3D scene with the learned latent vector. We provide additional details about this optimization step in the supplemental.

Evaluation Metrics We evaluate our method on the tasks of scene synthesis and single-view scene reconstruction using 3D-FRONT and ScanNet respectively. For scene synthesis, we generate 1000 scenes for each method and use FID score (FID), scene classification accuracy (SCA) and category KL divergence (KL) from [36] to measure the generation plausibility against the ground-truth in the test set. We refer readers to [36] for their definitions. For FID and SCA calculation, we render the generated and the ground-truth scenes into 512^2 images with a top-down orthographic projection, where each object is textured with a unique category-specific color. We unify this setting among all baselines for a fair comparison.

For single-view reconstruction, we use the metrics of 3D box IoU and chamfer distance between generated objects and the ground-truth. 3D box IoU measures the layout accuracy by comparing the 3D object bounding box between each ground-truth object and the prediction. Chamfer distance measures the accuracy of both layout and shape prediction. It calculates the point-wise L_2 distance between generated object surface points and the ground-truth.

5. Results and Analysis

We evaluate our method qualitatively and quantitatively on the tasks of scene synthesis and single-view reconstruction in comparisons with state-of-the-art baselines, as well as scene interpolation and an ablation analysis on different network configurations.

5.1. Scene Synthesis

Baselines Since there are no previous works that synthesize 3D semantic instance scenes by learning from only 2D supervision, for a fair comparison, we use several state-of-the-art baselines while train them under 2D supervision. Additionally, we also compare with the state-of-the-art methods which use 3D training data, to explore the boundary of scene synthesis by learning with only 2D supervision. We consider the following baselines: **1)** ATISS [36], a transformer encoder-based method that learns the distri-
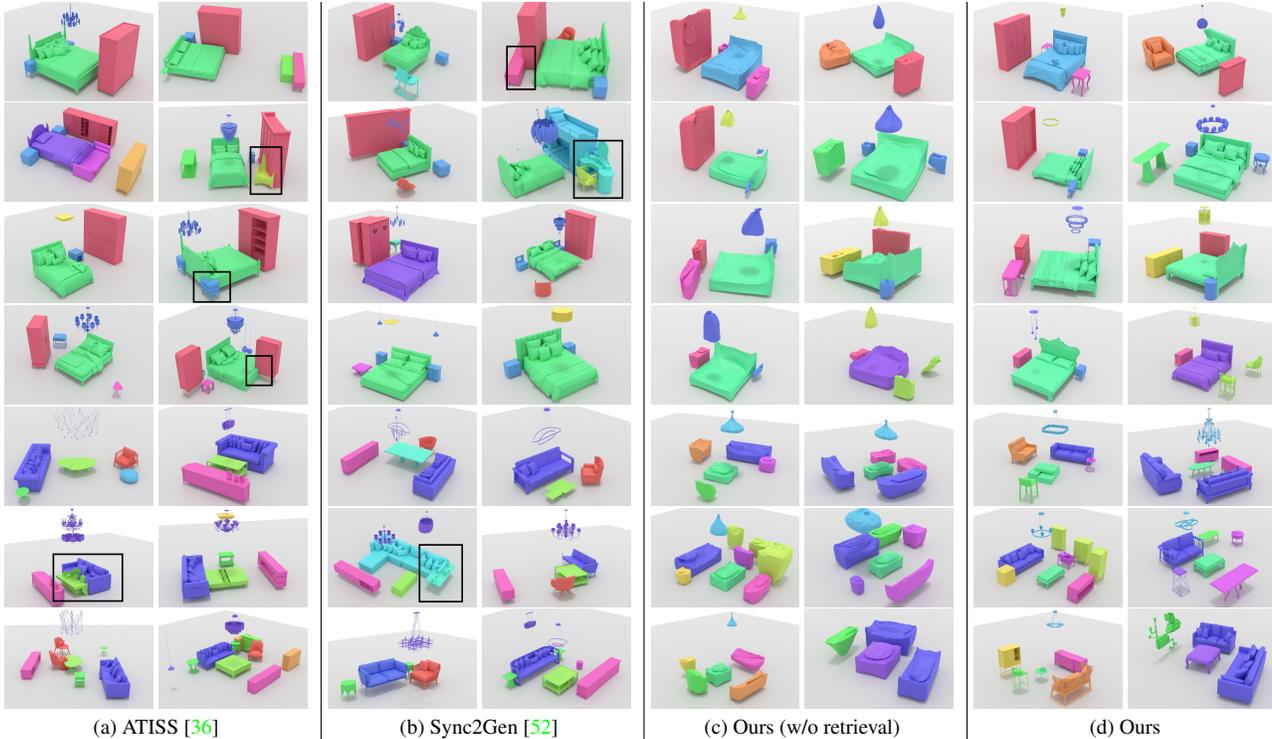
Figure 5. Qualitative comparisons with the state-of-the-art on scene synthesis. Note that both ATISS [36] and Sync2Gen [52] use 3D data for training while our method learns with only 2D supervision.

bution of 3D object bounding box properties. By progressively sampling from each property distribution, it generates a sequence of object boxes as the output. **2) Sync2Gen [52]**, a VAE-based method that maps a scene arrangement (parameterized with a sequence of 3D object attributes) to itself. After training, a 3D scene can be generated by sampling from the bottleneck latent distribution. **3) ATISS-2D.** We use the transformer in ATISS [36] as the backbone and adapt it to be trainable under 2D supervision with our view loss. **4) GAN.** We customize the generative adversary network for semantic scene synthesis from [53] to make it trainable with only 2D supervision. **5) LSTM.** We replace our transformer with an LSTM network, while keeping other modules unchanged. Detailed architecture specifications are given in the supplemental. We train/test all methods on the same split for fair comparisons.

Qualitative Comparisons Fig. 5 visualizes the qualitative comparisons with ATISS [36] and Sync2Gen [52], which are the state-of-the-art scene synthesis methods using 3D object data for training. While our method learns with only 2D supervision. Both the two methods add a shape retrieval post-processing to obtain object CAD models for evaluation. For fair comparison, we adopt shape retrieval by using Chamfer distance to measure the similarity between our reconstructed shape and CAD models in 3D-FRONT. We provide the details in the supplemental. From

Fig. 5, we observe that ATISS is vulnerable to the penetration issues between object bounding boxes (row 2,3,4,6). Sync2Gen addresses this by involving a Bayesian optimization post-processing to refine object placement based on a 3D object arrangement prior, but it is still vulnerable to penetration issues (row 1,2,6). Our method learns object layout and shape from only 2D data, where we learn each object conditioned on the existence of previous generated objects under the multi-view constraint, which helps to avoid object intersections and produces plausible scene arrangements.

Quantitative Comparisons Tab. 1 presents the quantitative comparisons between methods under different supervision sources. We observe that our method achieves better FID, SCA and KL scores against these methods using the same 2D supervision, which indicates that we produce better generation plausibility and more realistic object category distribution. To explore the boundary of our method, we also compare with prior arts which use 3D supervision. The results show that our method still presents better visual plausibility (FID and SCA scores) than ATISS, and shows comparable performance against Sync2Gen.

5.2. Single-view Reconstruction

Baselines We evaluate our method on single-view instance reconstruction from real images, and compare with the prior arts which use 3D supervision: Total3D [34] and

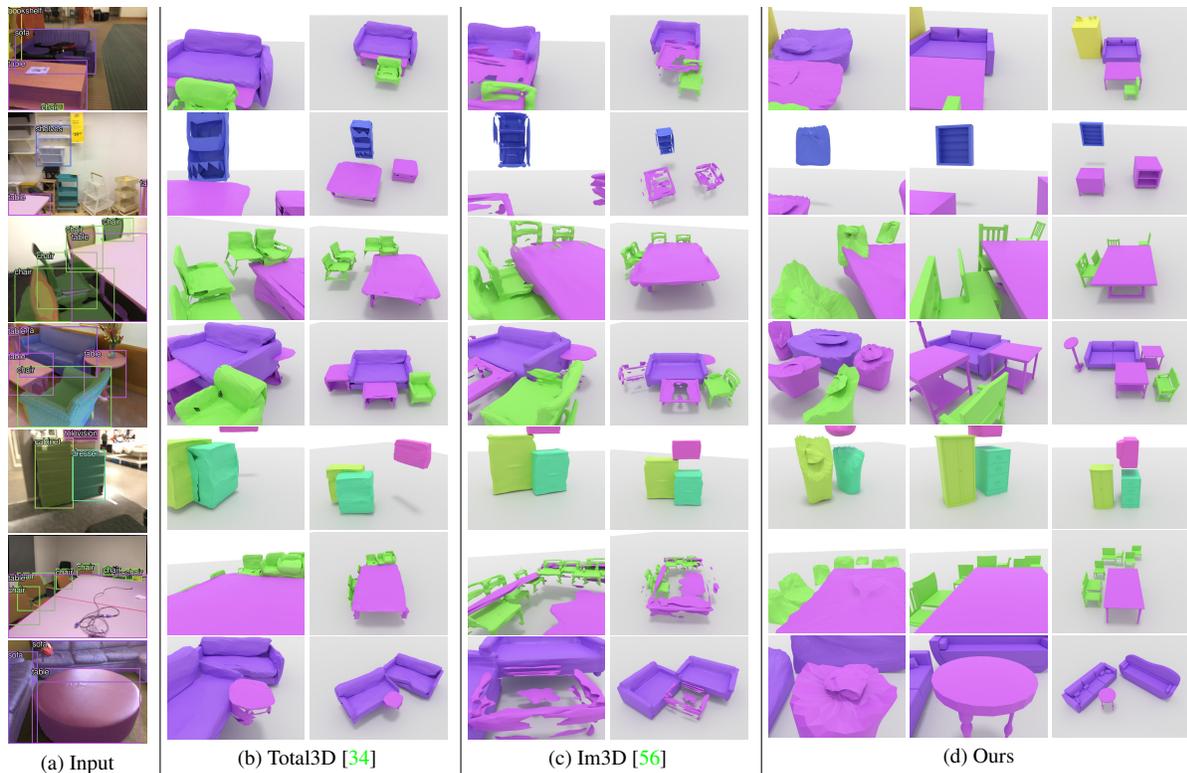| | (a) Input | (b) Total3D [34] | (c) Im3D [56] | (d) Ours |

Figure 6. Qualitative comparisons with the state-of-the-art on single view reconstruction. Note that both Total3D [36] and Im3D [52] use 3D data for training while our method learns from only 2D supervisions. For each method we present the reconstruction result from both the image view and a global view. For our method, we also provide the reconstruction result without shape retrieval on the left.

		Bedroom			Living room		
	Supervision	FID ($\downarrow$)	SCA	KL ($\downarrow$)	FID ($\downarrow$)	SCA	KL ($\downarrow$)
ATISS [36]	3D	26.82	0.97	<u>0.01</u>	50.02	0.97	0.02
Sync2Gen* [52]	3D	25.24	<u>0.80</u>	0.03	<u>35.34</u>	0.88	<u>0.02</u>
Sync2Gen [52]	3D	<u>23.86</u>	0.83	0.02	35.60	<u>0.86</u>	0.08
ATISS-2D [36]	2D	38.13	0.96	0.10	65.07	0.98	0.36
GAN [53]	2D	24.42	0.97	0.05	44.76	0.99	0.08
LSTM	2D	67.97	1.00	0.30	86.77	1.00	0.26
Ours	2D	**21.59**	**0.85**	**0.03**	**40.47**	**0.96**	**0.02**

Table 1. Quantitative comparisons on scene synthesis. Note that SCA score closer to 0.5 is better. Sync2Gen* is the version of [52] without Bayesian optimization. ATISS-2D is customized from ATISS [36] while using our 2D view loss. GAN is customized from [53] with using only 2D supervision.

Im3D [56]. Both of them learn 3D scenes with a fully supervised manner, i.e., they require ground-truth object meshes, 2D detections, 3D posed object bounding boxes, and camera poses for training, while our method learns from multi-view 2D instance masks with camera poses. We also compare with USL [16] which is an unsupervised method without using 3D supervision. Similar with ours, they require multi-view tracked instance masks for learning. We train and test all methods using the official split on ScanNet. For Total3D [34] and Im3D [56], we use their pretrained model and finetune their 3D detector on ScanNet. For finetuning, we uniformly sample up to 20 views to cover each scene,

with each view contains at least three objects, which produces 11,395 valid views. For USL [16], we reproduce their network due to the unavailability of an official implementation, and train it on the same data with ours. For fair comparison, we use ground-truth 2D object detections and camera poses as the input for training and testing all baselines.

Qualitative Comparisons Fig. 6 presents the qualitative comparisons with Total3D [34] and Im3D [56], which are the state-of-the-art single-view scene reconstruction methods. Both Total3D [34] and Im3D [56] pretrain their shape generator on a 3D shape dataset [44]. To produce better object geometry, our method keeps the shape retrieval in Sec. 5.1 while using the ShapeNet [5] dataset to retrieve CAD models given object positions and meshes. We observe that Total3D presents better mesh quality than Im3D. However, since objects in ScanNet images are usually occluded due to the high scene complexity, it struggles to estimate 3D object poses from 2D detections and results in erroneous object placement (row 1,3,5,6). Im3D addresses this by involving a refinement stage to improve object layout. However, for those severely occluded or partially visible objects, it is vulnerable to object penetration issues (row 1,3,7). Instead of detecting 3D objects from 2D detections (as in Total3D and Im3D), our method focuses on the prior learning of object arrangements, i.e., to understand what a

	Supervision	3D Box IoU ($\uparrow$)	CD ($\downarrow$)
Total3D [34]	3D	0.15	1.36
Im3D [56]	3D	0.23	0.84
USL* [16]	2D	0.10	1.73
Ours (w/o retrieval)	2D	0.25	1.01
Ours	2D	**0.27**	**0.79**

Table 2. Quantitative comparisons on single-view reconstruction. Note that USL* is reproduced from [16] due to the unavailability of the official implementation.

reasonable object layout looks like in multiple views, which helps to produce better scene configurations.

Quantitative Comparisons Tab. 2 presents the quantitative comparisons, where we observe that Total3D [34] and USL [16] struggles to detect 3D object bounding boxes from 2D detections (lower 3D Box IoU) because both of them predict object centers by depth estimation, which is still a challenging problem due to the ambiguity. Our method attempts to reconstruct a plausible scene from the learned scene prior conditioned on the input instance masks, which presents more reasonable 3D object bounding boxes. Since training and testing our method requires instance masks for supervision, the mask quality has a direct influence to our reconstruction results. However, mask artifacts are inevitable in real images (row 1-5 in Fig. 6 (a)). By learning shapes from multi-view observations with a naive shape retrieval post-processing, our method presents convincing object layouts and geometries.

5.3. Discussion

Scene Interpolation Fig. 7 presents the interpolation results between two scenes. Given the latent vectors of two scenes, we interpolate intermediate vectors by sampling from the geodesic curve between them on the hypersphere surface (see Fig. 1). By decoding them, we can obtain the interpolated scenes. The transition between intermediates results show that our method interpolate not only the object pose and geometry but also the semantics and scene context, where the interpolated scenes are constrained with a reasonable scene layout.

Ablations In Tab. 3 we investigate the effects of each submodule in scene synthesis, i.e., hyper-spherical latent space, permutation-invariant transformer, and stage-wise layout and shape learning, by replacing them with uniform latent space parameterization (c_1), LSTM (c_2), and one-stage layout and shape training (c_3), respectively. From the results, we observe that our transformer plays the most important role. LSTM encodes object order information. However, objects in a scene is an unordered set, and our transformer learns the permutation-invariant context to predict the next object recursively, which largely improves our performance. The hypersphere parameterization and stage-

	Bedroom			Living room		
	FID ($\downarrow$)	SCA	KL ($\downarrow$)	FID ($\downarrow$)	SCA	KL ($\downarrow$)
c_1	30.61	0.96	0.24	95.51	1.00	0.38
c_2	67.97	1.00	0.30	86.77	1.00	0.26
c_3	24.24	0.90	0.04	43.43	0.98	0.03
Ours	**21.59**	**0.85**	**0.03**	**40.47**	**0.96**	**0.02**

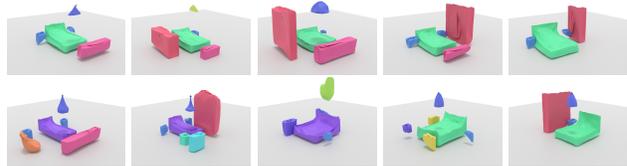Table 3. Ablation analysis on different configurations.

Figure 7. Scene interpolation between two samples (the leftmost and the rightmost).

wise training further increases the output plausibility and stabilizes the learning process, and combining them delivers the best performance.

Limitations Since training our network requires 2D posed images with instance masks, erroneous camera poses or mask labels would mislead the learning process. We believe an interesting direction for future work is to incorporate an uncertainty model to learn stable scene priors from noisy masks and camera poses.

6. Conclusion

We have presented a novel approach to learn the prior distribution of object layouts and shapes in 3D scenes with only 2D supervision. Our proposed method represents a 3D scene as a latent vector. By decoding it autoregressively using a permutation-invariant transformer, we recover a sequence of object instances characterized by their class categories, 3D bounding boxes and meshes. Since collecting 3D data at scale is expensive and intractable, we design a 2D view loss to train object generation using only 2D supervision. We demonstrate that such prior learning approach enables various downstream tasks, i.e., scene synthesis, interpolation and single-view reconstruction, and produces convincing scene arrangements. We hope that this establishes a step towards a universal self-supervised scene estimator that mimics human vision to understand, emulate and create 3D surroundings from only image observations.

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 2614–2623, 2019. 1

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 4

[3] Angel Chang, Manolis Savva, and Christopher D Manning. Learning spatial knowledge for text to 3d scene generation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 2028–2038, 2014. 2

[4] Angel X Chang, Mihail Eric, Manolis Savva, and Christopher D Manning. Sceneseer: 3d scene design with natural language. *arXiv preprint arXiv:1703.00050*, 2017. 2

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 7

[6] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. *Advances in Neural Information Processing Systems*, 34:8282–8293, 2021. 1, 2

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 5

[8] Andrey Davydov, Anastasia Remizova, Victor Constantin, Sina Honari, Mathieu Salzmann, and Pascal Fua. Adversarial parametric pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10997–11005, 2022. 3

[9] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019. 5

[10] Matthew Fisher and Pat Hanrahan. Context-based search for 3d models. In *ACM SIGGRAPH Asia 2010 papers*, pages 1–10. 2010. 2

[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. 2

[12] Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. Activity-centric scene synthesis for functional 3d scene modeling. *ACM Transactions on Graphics (TOG)*, 34(6):1–13, 2015. 2

[13] 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. 1, 5

[14] Qiang Fu, Xiaowu Chen, Xiaotian Wang, Sijia Wen, Bin Zhou, and Hongbo Fu. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017. 2

[15] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 2

[16] Georgia Gkioxari, Nikhila Ravi, and Justin Johnson. Learning 3d object shape and layout without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1695–1704, 2022. 1, 2, 5, 7, 8

[17] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. 4

[18] Can Gümeli, Angela Dai, and Matthias Nießner. Roca: Robust cad model retrieval and alignment from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4022–4031, 2022. 1, 2

[19] Ji Hou, Angela Dai, and Matthias Nießner. Revealnet: Seeing behind objects in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2098–2107, 2020. 1

[20] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *Proceedings of the European conference on computer vision (ECCV)*, pages 187–203, 2018. 1, 2

[21] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5134–5143, 2017. 1, 2

[22] Yun Jiang, Marcus Lim, and Ashutosh Saxena. Learning object arrangements in 3d scenes using human context. *arXiv preprint arXiv:1206.6462*, 2012. 2

[23] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. 3

[24] Mohammad Keshavarzi, Aakash Parikh, Xiyu Zhai, Melody Mao, Luisa Caldas, and Allen Y Yang. Scenegen: Generative contextual scene augmentation using scene graph priors. *arXiv preprint arXiv:2009.12395*, 2020. 1

[25] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3559–3568, 2018. 2

[26] Weicheng Kuo, Anelia Angelova, Tsung-Yi Lin, and Angela Dai. Mask2cad: 3d shape prediction by learning to segment and retrieve. In *European Conference on Computer Vision*, pages 260–277. Springer, 2020. 1, 2

[27] Weicheng Kuo, Anelia Angelova, Tsung-Yi Lin, and Angela Dai. Patch2cad: Patchwise embedding learning for in-the-wild shape retrieval from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12589–12599, 2021. 2

[28] Kurt Leimer, Paul Guerrero, Tomer Weiss, and Przemyslaw Musialski. Atek: Augmenting transformers with expert knowledge for indoor layout synthesis. *arXiv preprint arXiv:2202.00185*, 2022. 1

[29] 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

[30] Haolin Liu, Yujian Zheng, Guanying Chen, Shuguang Cui, and Xiaoguang Han. Towards high-fidelity single-view holistic reconstruction of indoor scenes. In *European Conference on Computer Vision*, pages 429–446. Springer, 2022. 1, 2

[31] Rui Ma, Honghua Li, Changqing Zou, Zicheng Liao, Xin Tong, and Hao Zhang. Action-driven 3d indoor scene evolution. *ACM Trans. Graph.*, 35(6):173–1, 2016. 2

[32] 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. 2

[33] Yinyu Nie, Angela Dai, Xiaoguang Han, and Matthias Nießner. Pose2room: Understanding 3d scenes from human activities. In *European Conference on Computer Vision*, pages 425–443. Springer, 2022. 2

[34] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 55–64, 2020. 1, 2, 6, 7, 8

[35] Yinyu Nie, Ji Hou, Xiaoguang Han, and Matthias Niessner. Rfd-net: Point scene understanding by semantic instance reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4608–4618, June 2021. 1

[36] 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, 5, 6, 7

[37] Stefan Popov, Pablo Bauszat, and Vittorio Ferrari. Corenet: Coherent 3d scene reconstruction from a single rgb image. In *European Conference on Computer Vision*, pages 366–383. Springer, 2020. 2

[38] Pulak Purkait, Christopher Zach, and Ian Reid. Sg-vae: Scene grammar variational autoencoder to generate new indoor scenes. In *European Conference on Computer Vision*, pages 155–171. Springer, 2020. 2

[39] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5899–5908, 2018. 2

[40] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 5

[41] 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

[42] Manolis Savva, Angel X Chang, and Maneesh Agrawala. Scenesuggest: Context-driven 3d scene design. *arXiv preprint arXiv:1703.00061*, 2017. 2

[43] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016. 4

[44] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7

[45] Shubham Tulsiani, Saurabh Gupta, David F Fouhey, Alexei A Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 302–310, 2018. 2

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[47] 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. 2

[48] 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

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

[50] Hualiang Xie, Wenzhuo Xu, and Bin Wang. Reshuffle-based interior scene synthesis. In *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, pages 191–198, 2013. 2

[51] Kun Xu, Kang Chen, Hongbo Fu, Wei-Lun Sun, and Shi-Min Hu. Sketch2scene: Sketch-based co-retrieval and co-placement of 3d models. *ACM Transactions on Graphics (TOG)*, 32(4):1–15, 2013. 2

[52] Haitao Yang, Zaiwei Zhang, Siming Yan, Haibin Huang, Chongyang Ma, Yi Zheng, Chandrajit Bajaj, and Qixing Huang. Scene synthesis via uncertainty-driven attribute synchronization. In *Proceedings of the IEEE/CVF International*

Conference on Computer Vision, pages 5630–5640, 2021. 1, 2, 5, 6, 7

[53] Ming-Jia Yang, Yu-Xiao Guo, Bin Zhou, and Xin Tong. Indoor scene generation from a collection of semantic-segmented depth images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15203–15212, 2021. 2, 6, 7

[54] Yi-Ting Yeh, Lingfeng Yang, Matthew Watson, Noah D Goodman, and Pat Hanrahan. Synthesizing open worlds with constraints using locally annealed reversible jump mcmc. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012. 2

[55] Lap Fai Yu, Sai Kit Yeung, Chi Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley J Osher. Make it home: automatic optimization of furniture arrangement. *ACM Transactions on Graphics (TOG)-Proceedings of ACM SIGGRAPH 2011, v. 30,(4), July 2011, article no. 86*, 30(4), 2011. 2

[56] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3d scene understanding from a single image with implicit representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8833–8842, 2021. 1, 2, 7, 8

[57] Song-Hai Zhang, Shao-Kui Zhang, Yuan Liang, and Peter Hall. A survey of 3d indoor scene synthesis. *Journal of Computer Science and Technology*, 34(3):594–608, 2019. 2

[58] 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. 2

[59] Ji-Zhao Zhu, Yan-Tao Jia, Jun Xu, Jian-Zhong Qiao, and Xue-Qi Cheng. Modeling the correlations of relations for knowledge graph embedding. *Journal of Computer Science and Technology*, 33(2):323–334, 2018. 2