

# Roof-GAN: Learning to Generate Roof Geometry and Relations for Residential Houses

Yiming Qian    Hao Zhang    Yasutaka Furukawa  
Simon Fraser University  
{yimingq, haoz, furukawa}@sfu.ca



Figure 1. The paper proposes a novel generative model, producing realistic residential roof models. An aerial shot of the real houses are at the left. Our generated samples are at the right. In the middle, we overlay a real house and our generation with similar structure.

## Abstract

*This paper presents Roof-GAN, a novel generative adversarial network that generates structured geometry of residential roof structures as a set of roof primitives and their relationships. Given the number of primitives, the generator produces a structured roof model as a graph, which consists of 1) primitive geometry as raster images at each node, encoding facet segmentation and angles; 2) inter-primitive colinear/coplanar relationships at each edge; and 3) primitive geometry in a vector format at each node, generated by a novel differentiable vectorizer while enforcing the relationships. The discriminator is trained to assess the primitive raster geometry, the primitive relationships, and the primitive vector geometry in a fully end-to-end architecture. Qualitative and quantitative evaluations demonstrate the effectiveness of our approach in generating diverse and realistic roof models over the competing methods with a novel metric proposed in this paper for the task of structured geometry generation. Code and data are available at <https://github.com/yi-ming-qian/roofgan>.*

## 1. Introduction

Residential roof structure exhibits intricate structural details and regularities. An observation reveals that a complex

polygonal surface structure emerges from a careful combination of a few primitive shapes under incident geometric relationships such as colinearity or coplanarity. Automated generation of high quality residential house models and any man-made structures beyond would have tremendous impact on broader disciplines such as construction, manufacturing, urban planning, and visual effects.

With the emergence of deep learning, automated generation of CAD-style 3D models has seen a breakthrough. Early methods focus on generating geometry without their incident relationships as a part assembly [17, 26, 35]. Auto-encoder based methods learn to generate both geometry and relationships in a form of a binary tree [18], a hierarchical N-ary tree [23], or deformable mesh models [10]. These techniques use fully connected layers with 1D feature vectors to produce CAD geometries in a vector format.

The paper takes the CAD-geometry generation research to the next level, while making the following distinctions from the existing methods: 1) Adversarial training is the foundation of our architecture, providing real generative power over auto-encoder based methods; 2) Convolution with raster-geometry representation enables effective spatial part arrangements and incident relationship generation; and 3) A novel differentiable vectorization module generates vector-geometry in an end-to-end architecture.

Concretely, we propose a novel generative adversarial

network, dubbed Roof-GAN. Given the number of primitives, the generator produces a structured geometry model as a graph. A node contains primitive geometry information as a 4-channel image (*i.e.*, roof facet segmentation and roof angles). An edge contains incident primitive relationships as one-hot vectors (*i.e.*, colinearity of footprint boundaries and parallelism of facets, where colinear and parallel implies coplanar). A node also contains geometry information in a vector format from a differentiable vectorizer, which is further capable of enforcing incident relationships. Roof-GAN employs two discriminators, one for assessing holistic geometry composition and the other for examining relationship labels together with geometry.

We have evaluated the proposed approach against the current state-of-the-art, while creating a new database of CAD-style roof geometry with incident relationships, consisting of 502 residential houses. Qualitative and quantitative evaluations demonstrate the effectiveness of Roof-GAN against competing methods in generating diverse and realistic set of roof models with a novel metric proposed in this paper for the task of structured geometry generation. Code and data are available at <https://github.com/yi-ming-qian/roofgan>.

## 2. Related Work

We review related techniques in three domains: architectural reconstruction, generative models for architectural structures, and assembly-based modeling.

**Architectural reconstruction:** Reconstruction of architectural elements such as lines, planes, room layouts, and 3D buildings has a long history in vision research. Traditional methods are either rule-based, *e.g.*, built on shape grammars [6, 20, 16], or use optimization with ad-hoc objectives, typically requiring depth or multiple view information to infer planes [9], room layouts [31], or CAD-quality objects [25]. With the surge of deep neural networks (DNNs), data-driven methods enable single-image reconstruction of depthmaps [21, 29], room layouts [41], or wire-frame building models [42]. Shape grammars are also utilized for roof reconstruction by DNNs, which classify grammar branches and estimate geometric parameters [38]. Recently, ConvMPN [39] learns to reconstruct vector-graphics building models, without shape grammars, by using a relational neural architecture, which is the backbone of our network. In contrast to reconstruction, our RoofGAN is a fully generative model, striving for plausibility and diversity.

**Generative models for architecture:** Procedural models with hand-crafted rules can provide production-level solutions to virtual building generation [24]. Various techniques have been leveraged to improve the generation quality such as Markov Chain Monte Carlo [32], discrete optimization [13] and, probabilistic graphical model [22].

Inspired by the success in deep image generation, sev-

eral deep generative networks have been proposed for 2D or 3D structured data [3]. Recursive neural networks have been applied to learn object placements and relations for indoor scene generation [19], while graph CNNs have been trained to generate room layouts [33] or floorplans [15, 36] by providing building outlines as input. Closer to our work, House-GAN [27] employs a generative adversarial network for floorplan generation, while requiring room adjacency relations as input. Roof-GAN also follows adversarial training, but does not input building outlines or adjacency relations. Instead, the relations are generated by the network.

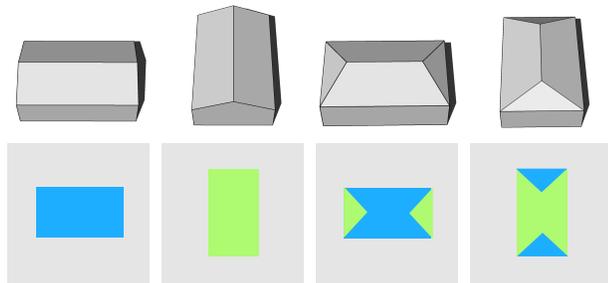
**Assembly-based modeling:** Many methods have been proposed for 3D shape generation using different representations: voxels [34, 11], point clouds [2, 8], meshes [4], and implicit functions [5, 28]. Generative models for 3D shape structures [37, 3] are typically trained to learn models of part variations and assembly. Most representative works resort to part and structure autoencoders, that are built on different shape structure representations, including hierarchical trees in GRASS [18], graphs in StructureNet [23], part assembly sequences in PQ-Net [35], and more general graphs in other works [30, 7, 10]. In contrast, our network is designed with generative capability and diversity in mind, relying on adversarial training which is shown to be effective even with a relative small training set.

## 3. Structured Roof Geometry Representation

A combination of a few number of rectangular roof primitives explains the majority of residential roof structure. We represent a roof as a graph, where a node encodes primitive geometry information and an edge encodes pairwise incident relationships. Structured geometry representation is at the heart of this research, enabling 1) effective spatial reasoning of primitive arrangement; and 2) differentiable vectorization in an end-to-end architecture.

**Geometry representation at nodes:** We assume that each primitive is an axis-aligned rectangle. The top covering is either horizontal-gable, vertical-gable, horizontal-hip, or vertical-hip (See Fig. 2). A 4-channel image is our representation, where the first three channels constitute one-hot encoding over the three facet *orientations*: 1) left/right facing; 2) top/bottom facing; and 3) background. The fourth channel is the facet *angle* (*i.e.*, the angle between the normal and the inverse gravity), which is set to 0 for background.

Differentiable vectorization (Sect. 4.2) converts a 4-channel image into a 6d vector. The six numbers are the four boundary coordinates of the rectangle and the two roof angles (one for left/right facets and one for top/bottom facets). Note that we are further assuming that 1) Primitives are symmetric, where the angles of the left/right (*resp.* the top/bottom) facets are equal. 2) The height of the wall is a constant (not generated). 3) When primitives overlap, we keep the highest facet and ignore invisible portions.



(a) Horizontal gable (b) Vertical gable (c) Horizontal hip (d) Vertical hip

Figure 2. Our model handles four types of roof primitives, whose sample 3D models are shown at the top. The bottom shows a part of our raster geometry representation, where a 3-dimensional one-hot encoding represents the roof facet type per pixel: top/bottom facing (blue), left/right facing (green), or background (grey). The representation also has a roof-angle image, which stores the angle between the facet normal and the inverse gravity per pixel.

**Relationship representation at edges:** For each pair of primitives, 6d one-hot encoding represents colinearity of rectangular boundaries and parallelism of polygonal facets. Concretely, the first dimension encodes if the left rectangular boundaries are colinear. The next three dimensions are for the colinearity of the right/top/bottom boundaries. We do not model a colinearity of the left boundary and the right boundary, which rarely happens in our data because primitives overlap instead. The remaining two dimensions encode the parallelism of the primitive pair (one for the left/right facets and one for top/bottom facets), where facets are coplanar when they are parallel and the corresponding boundaries are colinear.

## 4. Roof-GAN

Relational GAN for floorplan generation is our backbone [27]. Given the number of primitives, Roof-GAN initializes a complete graph of noise vectors and generates a roof geometry (See Fig. 3). We focus on the differences from the prior work, where the architectural details are in the supplementary document and Section 6 explains how we pick the number of primitives at training and testing.

### 4.1. Generator

Each node is initialized with a 128d noise vector sampled from the normal distribution. Note that there are no one-hot node-type vectors as in the prior work [27], because node/edge properties are to be generated instead of given in our work. The architecture is the same till the output layers, where the last feature volume per node has the dimension of  $16 \times 32 \times 32$ . Two 3-layer CNNs are used to produce a  $3 \times 32 \times 32$  facet orientation image and a  $1 \times 32 \times 32$  facet angle image. A softmax function is applied for the orientation image and a sigmoid function is applied for the angle

image as the cosine value of the angle. For each node pair, we concatenate node features into a  $32 \times 32 \times 32$  volume, and apply a 5-layer CNN to downsample it into a  $512 \times 1 \times 1$  vector. Finally, we use a linear layer and a sigmoid function to convert to a 6d relationship vector. Note that the node feature concatenation order is arbitrary.

### 4.2. Differentiable vectorization

A novel differentiable vectorization converts the raster geometry into the vector format, serving two purposes. First, this eradicates the post-processing heuristics for vectorization. Second, raster-geometry may not follow the relationships. The vector parameterization allows us to generate relationship-enforced raster-geometry, which can be passed to the discriminator for assessment. There are two vectorization modules, one for rectangle boundary coordinates and the other for roof angles/type. We here describe the former and refer the latter to the supplementary document, which is rather straightforward with a sequence of reasonable heuristics. Note that both modules are fixed algebraic machinery without learnable weights.

Zhang *et al.* proposed a neat vectorization trick for corner detection, which takes a corner probability image from CNN and computes their weighted mean coordinate [40]. We extend the idea to rectangle boundary coordinate vectorization. Given a facet orientation image (*i.e.*, a probability image over left/right, top/bottom, or background classes), we obtain a primitive mask probability by one minus the background probability. Let us use the left-boundary coordinate as an example. We compute the x-derivative of the mask by finite difference, apply ReLU to keep only non-negative responses, then take the weighted mean coordinate [40]. The non-negative responses should concentrate on the left boundary of the mask and this algebraic formula computes the left boundary coordinate. Exactly the same algebraic operations apply to the other three boundaries.

### 4.3. Differentiable relationship enforcement

Given rectangle boundary coordinates, we 1) enforce their colinearity relationships and obtain adjusted boundary coordinates probabilistically; 2) solve for the non-uniform scale and translation that maps the original rectangle to the adjusted one; and 3) warp the facet orientation/angle images based on the transformation to obtain relationship-enforced geometry-images. Note that roof angle generation is more stable, and we do not enforce the parallelism relationships with the roof angle image.

Without loss of generality, the adjusted left coordinate of one primitive is calculated as the weighted average of the left coordinates of all the primitives. The weight is 1 for itself and the colinear relationship probabilities for the others.<sup>1</sup> Given the original and the adjusted coordinates,

<sup>1</sup>Low-probability relationship should not influence, and the weight for-

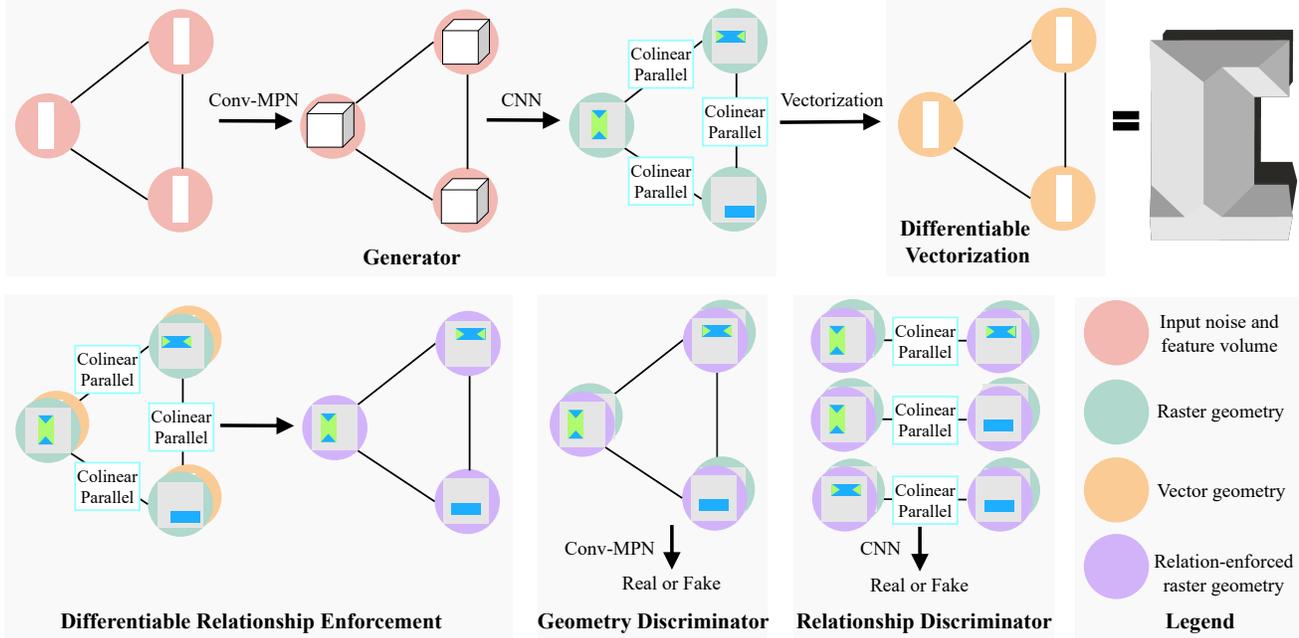


Figure 3. Roof-GAN architecture. The generator takes a graph of noise vectors as input and utilize Conv-MPN [39] to obtain 3D feature volumnes. Several convolutional layers are used to output per-node raster-geometry (here we draw orientation images only for illustration) and pairwise relationships. A differentiable vectorization module converts the raster-geometry to the vector-geometry, followed by a differentiable relationship enforcement. The geometry discriminator takes the raster-geometry before and after the relationship-enforcement for all the nodes. The relationship discriminator acts on a pair of nodes while also taking the relationship information as the input.

we compute non-uniform scaling and translation parameters and apply image-warping to the facet orientation/angle images by “grid\_sample” built-in function in PyTorch.

#### 4.4. Discriminators

Roof-GAN has two discriminators, whose loss functions are added with equal weights. The first discriminator focuses on geometry without relationships. As in the generator, we form a complete relational graph. At each node, we concatenate the orientation image and the relationship-enforced orientation image into a  $6 \times 32 \times 32$  volume. A 3-layer CNN converts to a  $16 \times 32 \times 32$  volume. The same architecture converts the angle images into a  $16 \times 32 \times 32$  volume. We further concatenate these two volumes to obtain an input to the Conv-MPN architecture, which is the same as the prior work [27]. For ground-truth samples, the relationship-enforced images are identical to the original.

The second discriminator takes a pair of nodes and an edge. A 6d relationship vector at the edge is converted to a 4096d vector with a linear layer, and reshaped to a  $4 \times 32 \times 32$  volume. We concatenate with the facet orientation/angle images before and after the relationship enforcement (*i.e.*, eight tensors), then use a 5-layer CNN to down-sample to  $128 \times 1 \times 1$ , followed by a linear layer to output a scalar. The average over all node-pairs is the discrimination score.

formula is in fact  $\text{ReLU}(2p - 1)$ , where  $p$  is the original probability.

## 5. Dataset and Metrics

Zeng *et al.* introduced a database of residential houses in England, containing height maps and surface normal maps in the aerial view [38], which we borrow to construct our database of structured roof geometry with relationships. Quantitative evaluation of vector-graphics geometry is a non-trivial task especially for the generative models. We propose a new metric, dubbed Recursive Minimum Matching Distance, which assess the realism and diversity of the generated samples as in the FID metric, while respecting the vector structure of the geometry representation.

### 5.1. Dataset

Given a surface normal image of a house in a Nadir view, we use the annotation tool “Colabeler” [1] to annotate the bounding box of each rectangle primitive and its corresponding roof type. A total of 502 houses are annotated, out of which (152, 256, 68, 26) houses have (2, 3, 4, 5) rectangles, respectively. The colinearity relationships are obtained by simply checking the equality of the annotated bounding box coordinates with an error tolerance of 1 pixel to account for rare human errors. Facet angles are calculated from depthmaps by the pre-processing algorithm of the prior work [38]. Parallelism is automatically determined by checking the equality of the angles with an error tolerance of  $18^\circ$ . When detecting the colinearity (resp.

parallelism), we enforce the relationship by snapping one coordinate (resp. angle) to the other.

The next step is to merge connected coplanar (i.e., co-linear and parallel) rectangular facets into a single polygonal facet. For each house, we rasterize all the primitives from their parameters into a common facet orientation image (instance-aware), facet angle image, and a height map, while discarding invisible facet information per pixel based on the height values. We use the rasterized images to identify connected coplanar facets and merge them into a single polygon. The data pre-processing algorithms are all heuristics and their details are referred to the supplementary. Finally, we perform standard 8x data augmentation by 90° rotation and mirroring, yielding a total of 4,016 houses.

## 5.2. Metrics

For quantitative evaluation of generative models for structured geometry, we propose a novel distance metric for a pair of sets of polygonal 3D models, dubbed Recursive Minimum Matching Distance (RMMD), which measures the realism and the diversity of the generation.<sup>2</sup> RMMD is recursively defined for a pair  $(s_1, s_2)$ , each of which is a set of polygonal surface models  $(\{m_i\})$ , each of which is a set of facets  $(\{f_i\})$ , each of which is a set of vertices  $(\{v_i\})$ :

$$D_S(s_1, s_2) = \sum_{m_1 \in s_1} \min_{m_2 \in s_2}^* \frac{D_M(m_1, m_2) + D_M(m_2, m_1)}{2|s_1|},$$

$$D_M(m_1, m_2) = \sum_{f_1 \in m_1} \min_{f_2 \in m_2}^* \frac{D_F(f_1, f_2) + D_F(f_2, f_1)}{2|m_1|},$$

$$D_F(f_1, f_2) = \sum_{v_1 \in f_1} \min_{v_2 \in f_2}^* \frac{D_V(v_1, v_2) + D_V(v_2, v_1)}{2|f_1|},$$

$$D_V(v_1, v_2) = |v_1 - v_2|.$$

At the bottom of the recursion, the distance  $D_V(v_1, v_2)$  between two vertices is their Euclidean distance. The distance  $D_F(f_1, f_2)$  between two facets is computed as follows. For each vertex in  $f_1$ , we find the closest vertex from  $f_2$  based on the vertex-distance. The average over all the vertices in  $f_1$  is the facet-distance, except that we enforce mutual exclusiveness in the matching, that is, each vertex in  $f_2$  is matched at most once. The mutual exclusiveness implies that when  $f_1$  has more vertices than  $f_2$  (topologically inconsistent), some vertices do not have matches and the largest possible distance (i.e., diagonal of the square image) is used for the average calculation ( $\star$  denotes a special min operation with abuse of notation). In practice, after computing all pairwise vertex distances, we use a greedy algorithm to find the minimum distance matching. The model  $D_M$  and the set  $D_S$  distances are defined in the same way recursively.

<sup>2</sup>Since our roof 3D model can be represented as a 2D polygon in a Nadir view, we present the metric in a case of 2D polygons but the formula applies to general 3D models.

RMMD measures geometric and topological differences into a single number in the unit of the vertex Euclidean distance. Note that each distance function is asymmetric and we make it symmetric inside each summation. At the top level, the number of ground-truth samples (512 augmented from 64) is less than the number of samples in a generation (1000), and an asymmetric distance  $D(g_1, g_2)$  is used as our metric, where  $g_1$  is the ground-truth set. RMMD measures the realism and the diversity, because there must be a similar model for every ground-truth sample.

Lastly, RMMD is sensitive to scale and translation differences. We normalize each model by taking its axis aligned bounding box and apply uniform scaling and translation so that the bounding box is center-aligned and tightly fits inside a square image, which is assumed to be  $16m \times 16m$ .

We also use a standard FID metric [14]. Each roof model is represented as a surface normal image where the 3D surface normal vector is used as a RGB value and the background is set to white. Again, 64 ground-truth samples and 1,000 generated samples are used for the metric calculation.

## 6. Experiments

We have implemented Roof-GAN in PyTorch and trained the models with WGAN-GP [12] and the Adam optimizer. The learning rate is 0.0001, the batch size is 16, the number of critics is 1, and the weight of gradient penalty is 10. The training takes about 17 hours on an NVIDIA GTX 1080 Ti GPU with 11GB of RAM for 200k iterations.

### 6.1. Competing methods

We have compared against three competing methods and two Roof-GAN variants for an ablation study.

- **PQ-Net** [35] learns a latent geometry representation with a Seq2Seq-based auto-encoder, followed by latent GAN [2] for code generation. We keep the original auto-encoder architecture and modify the input/output as a sequence of roof primitives to fit our problem. Concretely, the input/output vector is a concatenation of boundary coordinates, facet angles, one-hot encoded primitive types, and the number of primitives. The training requires a pre-defined sequential order of primitives, for which we start from the largest rectangle, and then sort in a decreasing order of the distances to the largest one. We make no changes to the latent GAN.
- **PQ-Net-Relation** is a novel variant of PQ-Net, where pairwise relationships are serialized into a one-hot vector with a fixed order and concatenated to the representation. Note that the original PQ-Net did not encode relationships.
- **House-GAN** is a state-of-the-art floorplan generative model [27]. We make two modifications: (1) At the end of the generator, we add two 4-layer CNNs (each followed by a linear layer) to output the facet angle vectors and the one-hot vector encoding the primitive types; (2) At the be-

Table 1. Quantitative evaluations. Three variants of the proposed approach (Roof-GAN) are compared against the three competing methods on the two metrics. Recursive minimum matching distance (RMMD) is a new metric, while FID is a standard one. The smaller the better for both metrics. “GAN” column indicates if a method is GAN-based or not. “Rela.” column indicates if a method generates relationships or not, as opposed to threshold-based snapping. The colors cyan and blue represent the best and the second best methods.

Method	GAN	Rela.	FID	RMMD
PQ-Net			12.0	7.80
PQ-Net-Relation		✓	13.5	6.97
House-GAN	✓		18.4	7.43
Roof-GAN (w/o rela.)	✓		10.6	6.61
Roof-GAN (w/o diff.)	✓	✓	11.9	6.48
Roof-GAN	✓	✓	9.8	6.20

gining of the discriminator, we add two linear layers to transform the facet angle vector and the primitive type vector to 512d, which is reshaped to  $2 \times 16 \times 16$  and concatenated with the feature tensor of the primitive mask.

- **Roof-GAN (w/o rela.)** is a variant of our architecture without the relationship generation. The relationship discriminator, the differentiable vectorization, and the differentiable relationship enforcement are also removed.
- **Roof-GAN (w/o diff.)** is a variant without the differentiable vectorization and the differentiable enforcement modules during training. These modules do not have learnable weights and are utilized during testing.

PQ-Net, House-GAN, and Roof-GAN (w/o rela.) do not generate relationship labels, and a simple snapping heuristic with a threshold is used. We vary the thresholds, but do not see much differences in the results. Therefore, we use an error tolerance of 1 pixel for colinearity and  $18^\circ$  for parallelism, the same threshold setting used in the ground-truth preparation. House-GAN and Roof-GAN requires the relational graph, in particular, the number of primitives for a model generation. Both at training and testing time, we randomly pick the number of primitives by following the statistics of samples in our database, that is, (30%, 51%, 14%, 5%) for houses with (2, 3, 4, 5) primitives. PQ-Net and PQ-Net-Relation sequentially produces primitives until the termination probability reaches 0.5. We have 502 house samples before augmentation and randomly split them into 438 training and 64 testing samples. Each system generates 1,000 samples for the evaluation.

## 6.2. Quantitative evaluations

Table 1 shows the main quantitative evaluations, where Roof-GAN makes clear improvements over all the competing methods on both metrics. PQ-Net is one of the state-of-the-art methods but has the largest error with the RMMD metric. PQ-Net-Relation is a novel baseline proposed in

Table 2. Cross-validation results. To prevent a method from simply copying and pasting, we split the training and testing sets based on the number of primitives. See the text for the details.

Method	3 Primitives		4 Primitives	
	FID	RMMD	FID	RMMD
PQ-Net	13.0	10.4	14.6	12.9
House-GAN	27.5	8.5	27.2	12.5
Roof-GAN	11.1	7.5	13.8	10.9

this paper, which encodes relationships in addition to the geometric parameters. PQ-Net-Relation reduces the error by roughly 11%, which verifies the importance of relationship encoding, a key contribution of this paper. House-GAN is another existing state-of-the-art and better than PQ-Net by about 5% in RMMD, but inferior to all our variants.

The last three rows of Table 1 form the ablation study over the two technical contributions, relationship encoding and differentiable modules. Roof-GAN (w/o rela.) does not use relationship encoding and the differentiable modules, dropping the performance by about 6.6% compared to Roof-GAN (all). The comparison between Roof-GAN (w/o rela.) and House-GAN sheds a light on our geometry representation (Sect. 3), which is in fact the only difference between the two systems. House-GAN uses vectorized representation for facet angles and primitive types while we use the new raster representation in Sect. 3. Roof-GAN (w/o diff.) drops the performance about 4.5%, validating the contributions of the differentiable modules.

The contributions of our innovations become even more significant in the cross validation experiments in Table 2. In order to prevent a method from copying and pasting models, we split the training and testing sets based on the number of primitives. Concretely, for the left columns (3 Primitives) of the table, we train the networks on houses with 2, 4, and 5 primitives and test on houses with 3 primitives. The performance gap further increases, where our RMMD score is better by 39% than PQ-Net and by 13% than House-GAN for the case of (3 Primitives). Note that this evaluation may appear unfair for PQ-Net, which does not use the number of primitives in the test set. However, PQ-Net does not allow an external control and rarely generate such samples (*i.e.*, around 3%), incapable of exploring new compositions.

Roof-GAN outperforms all the other methods in the FID metric consistently. However, it is not clear why and how. For example, House-GAN is the worst in FID, despite reasonable RMMD scores. RMMD has an intuitive physical meaning with a physical unit, and can also provide a distance measure for a particular pair of samples, which can be used for a model retrieval evaluation, all of which will be demonstrated in the next section.

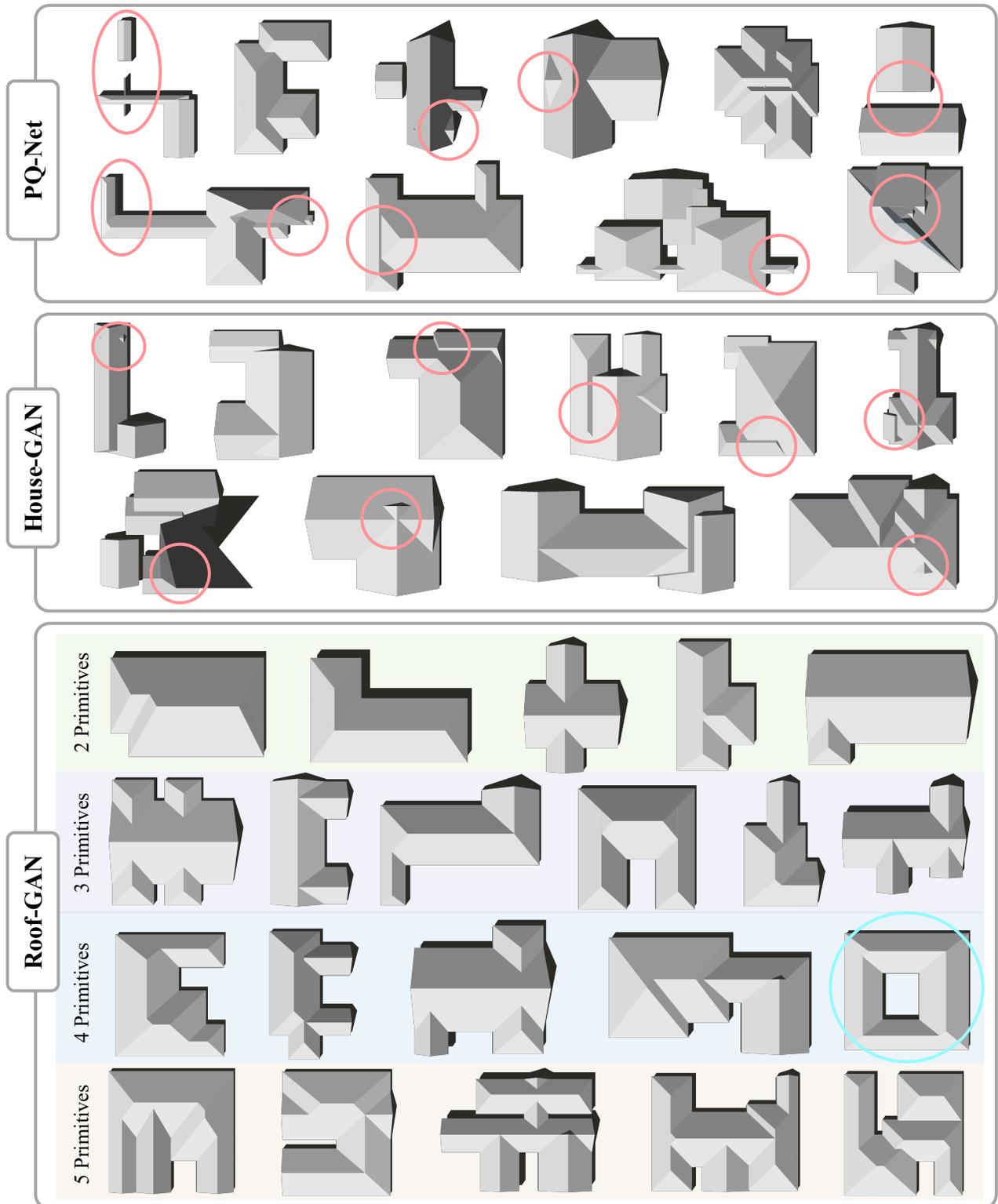


Figure 4. Qualitative evaluations among PQ-Net [35], House-GAN [27], and Roof-GAN (ours). Red ovals indicate non-realistic roof structures by the competing methods. Roof-GAN results are grouped row-by-row based on the number of primitives. Roof-GAN is capable of producing a new realistic roof structure that does not exist in our dataset, for example, an O-shaped building highlighted by the cyan oval.

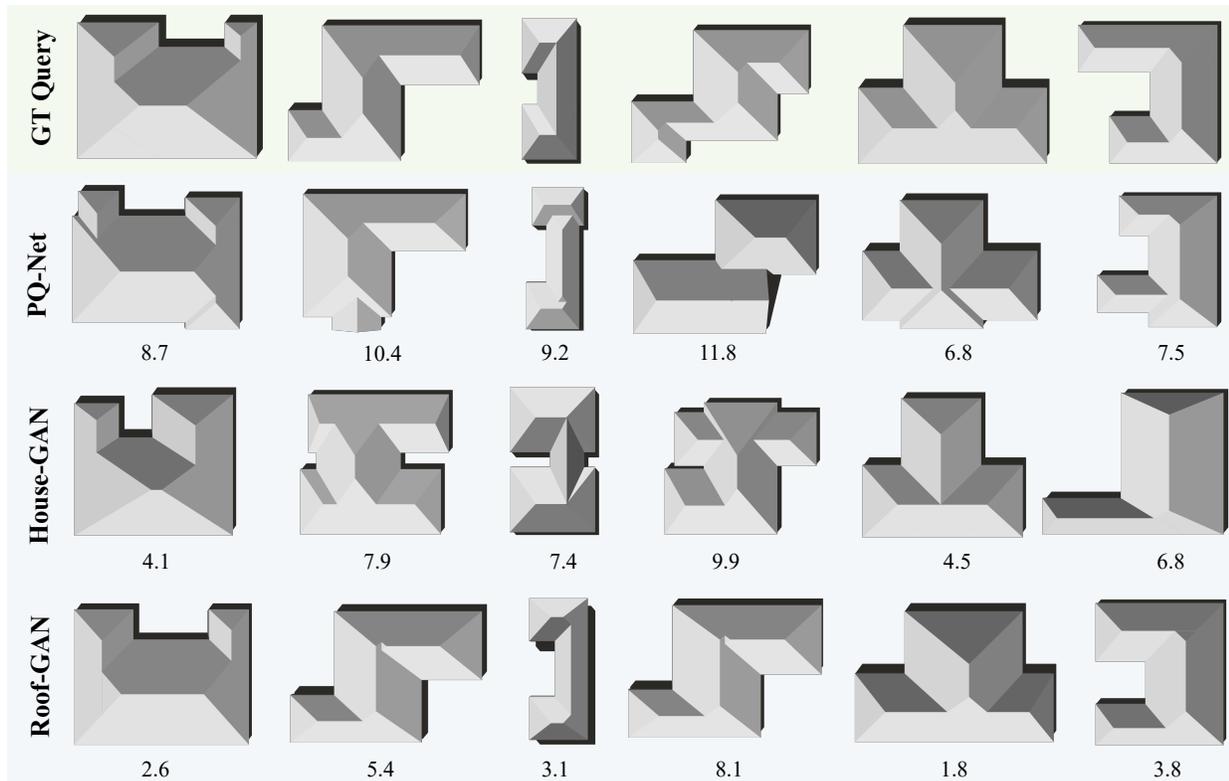


Figure 5. RMMD-based model retrieval evaluations. In each column, given a query ground-truth model at the top, we use the RMMD-metric to find the closest sample among 1,000 generations by each method. Roof-GAN models are visually similar to the queries, which is also supported by the RMMD metric listed at the bottom of each model.

### 6.3. Qualitative evaluations

Figure 4 compares generated models by the three methods. Both PQ-Net and House-GAN generate unrealistic roof models. For example, PQ-Net produces isolated, too long, or too thin components. Individual primitive shapes look much better in House-GAN, but poor incident relationships lead to unrealistic polygonal shapes and topology due to the failures of threshold based snapping. Roof-GAN, on the other hand, learns to generate incident relationships, producing complex and realistic combination of roof primitives. The last four rows of the figure show the roof models with 2, 3, 4, and 5 primitives, respectively. Diversity of our generation is apparent in each row, especially in the last two rows. Compositional capability is another strength of Roof-GAN, which generates a new realistic roof structure such as an O-shaped building (highlighted by the cyan oval in Fig. 4), which does not exist in our database.

Lastly, Figure 5 demonstrates the model retrieval evaluations based on the RMMD metric. Each column shows a reference GT model at the top, and the closest sample by each method (from 1,000 generations). Roof-GAN is able to produce visually similar roof structures consistently, which is also supported by the RMMD metric. For exam-

ple in the second column from the right, Roof-GAN is the only sample that has the same topology as the ground-truth. House-GAN sample is reasonable but misses a triangular facet in the middle and is penalized severely by the metric. We provide more retrieval evaluations and the visualization of reconstructed models in the supplementary.

## 7. Conclusion

This paper proposes a novel generative adversarial network that generates structured geometry of residential roof structures. For a roof represented as a graph, the generator learns to generate primitive geometries at the nodes and incident geometric relationships at the edges. Qualitative and quantitative evaluations demonstrate the effectiveness of our approach in generating diverse and realistic roof models over all the competing methods. Our future work includes the handling of higher-order primitive relationships such as symmetries and more diverse and complex primitive types such as dormers and chimneys.

**Acknowledgements:** This research is partially supported by NSERC Discovery Grants (No. 611714 and No. 611370), an NSERC Discovery Accelerator Supplement, and an DND/NSERC Discovery Grant Supplement.

## References

- [1] Colabeler. <http://www.colabeler.com/>. 4
- [2] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 2, 5
- [3] Siddhartha Chaudhuri, Daniel Ritchie, Jiajun Wu, Kai Xu, and Hao Zhang. Learning generative models of 3d structures. *Computer Graphics Forum (Eurographics STAR)*, 2020. 2
- [4] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 45–54, 2020. 2
- [5] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2
- [6] Anthony R Dick, Philip HS Torr, and Roberto Cipolla. A bayesian estimation of building shape using mcmc. In *European Conference on Computer Vision*, pages 852–866. Springer, 2002. 2
- [7] Anastasia Dubrovina, Fei Xia, Panos Achlioptas, Mira Shahlah, and Leonidas J. Guibas. Composite shape modeling via latent space factorization. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2019. 2
- [8] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 2
- [9] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Manhattan-world stereo. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1422–1429. IEEE, 2009. 2
- [10] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yukun Lai, and Hao Zhang. Sdm-net: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019. 1, 2
- [11] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016. 2
- [12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017. 5
- [13] Mark Hendrikk, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1):1–22, 2013. 2
- [14] 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. In *Advances in neural information processing systems*, pages 6626–6637, 2017. 5
- [15] Ruizhen Hu, Zeyu Huang, Yuhan Tang, Oliver Van Kaick, Hao Zhang, and Hui Huang. Graph2plan: Learning floor plan generation from layout graphs. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2020)*, 39(4):118:1–118:14, 2020. 2
- [16] Hai Huang, Claus Brenner, and Monika Sester. A generative statistical approach to automatic 3d building roof reconstruction from laser scanning data. *ISPRS Journal of photogrammetry and remote sensing*, 79:29–43, 2013. 2
- [17] Haibin Huang, Evangelos Kalogerakis, and Benjamin Marlin. Analysis and synthesis of 3d shape families via deep-learned generative models of surfaces. In *Computer Graphics Forum*, volume 34, pages 25–38. Wiley Online Library, 2015. 1
- [18] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. GRASS: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 1, 2
- [19] 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*, 38(2):Article 12, 2019. 2
- [20] Hui Lin, Jizhou Gao, Yu Zhou, Guiliang Lu, Mao Ye, Chenxi Zhang, Ligang Liu, and Ruigang Yang. Semantic decomposition and reconstruction of residential scenes from lidar data. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013. 2
- [21] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019. 2
- [22] Paul Merrell, Eric Schkufza, and Vladlen Koltun. Computer-generated residential building layouts. In *ACM SIGGRAPH Asia 2010 papers*, pages 1–12. 2010. 2
- [23] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics (TOG), Siggraph Asia 2019*, 38(6):Article 242, 2019. 1, 2
- [24] Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool. Procedural modeling of buildings. In *ACM SIGGRAPH 2006 Papers*, pages 614–623. 2006. 2
- [25] Liangliang Nan and Peter Wonka. Polyfit: Polygonal surface reconstruction from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2353–2361, 2017. 2
- [26] Charlie Nash and Christopher KI Williams. The shape variational autoencoder: A deep generative model of part-segmented 3d objects. In *Computer Graphics Forum*, volume 36, pages 1–12. Wiley Online Library, 2017. 1
- [27] Nelson Nauata, Kai-Hung Chang, Chin-Yi Cheng, Greg Mori, and Yasutaka Furukawa. House-gan: Relational generative adversarial networks for graph-constrained house layout generation. *European Conference on Computer Vision*, 2020. 2, 3, 4, 5, 7
- [28] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation.

- In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 2
- [29] Yiming Qian and Yasutaka Furukawa. Learning pairwise inter-plane relations for piecewise planar reconstruction. In *European Conference on Computer Vision*, pages 330–345. Springer, 2020. 2
- [30] Nadav Schor, Oren Katzier, Hao Zhang, and Daniel Cohen-Or. Learning to generate the “unseen” via part synthesis and composition. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2019. 2
- [31] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 2
- [32] Jerry O Talton, Yu Lou, Steve Lesser, Jared Duke, Radomír Měch, and Vladlen Koltun. Metropolis procedural modeling. *ACM Transactions on Graphics (TOG)*, 30(2):1–14, 2011. 2
- [33] 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
- [34] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems*, pages 82–90, 2016. 2
- [35] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 829–838, 2020. 1, 2, 5, 7
- [36] Wenming Wu, Xiao-Ming Fu, Rui Tang, Yuhan Wang, Yu-Hao Qi, and Ligang Liu. Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics (TOG)*, 38(6):1–12, 2019. 2
- [37] Kai Xu, Vladimir G Kim, Qixing Huang, Niloy Mitra, and Evangelos Kalogerakis. Data-driven shape analysis and processing. In *SIGGRAPH ASIA 2016 Courses*, pages 1–38. 2016. 2
- [38] Huayi Zeng, Jiaye Wu, and Yasutaka Furukawa. Neural procedural reconstruction for residential buildings. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 737–753, 2018. 2, 4
- [39] Fuyang Zhang, Nelson Nauata, and Yasutaka Furukawa. Conv-mpn: Convolutional message passing neural network for structured outdoor architecture reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2798–2807, 2020. 2, 4
- [40] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2694–2703, 2018. 3
- [41] Yichao Zhou, Haozhi Qi, Yuexiang Zhai, Qi Sun, Zhili Chen, Li-Yi Wei, and Yi Ma. Learning to reconstruct 3d manhattan wireframes from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7698–7707, 2019. 2
- [42] Chuhan Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2059, 2018. 2