DepthGAN: GAN-based Depth Generation from Semantic Layouts

Yidi Li
University of Chinese Academy of Sciences
Beijing, China
liyidi19@mails.ucas.ac.cn

Jun Xiao*
University of Chinese Academy of Sciences
Beijing, China
xiaojun@.ucas.ac.cn

Yiqun Wang*
Chongqing University
Chongqing, China
yiqun.wang@cqu.edu.cn

Zhengda Lu
University of Chinese Academy of Sciences
Beijing, China
luzhengda@.ucas.ac.cn

Abstract

Existing GAN-based generative methods are typically used for semantic image synthesis. We pose the question of whether GAN-based architectures can generate plausible depth maps and find that existing methods have difficulty in generating a reasonable depth map to represent the 3D scene structure due to the lack of global geometric correlation. To this end, we propose the DepthGAN, a novel method of generating a decent depth map with a semantic layout as input for further constructing and manipulating well-structured 3D scene point clouds. Specifically, we first build a feature generation model with a cascade of semantic-aware transformer blocks to obtain depth features with global structure information. In our semantic-aware transformer block, we propose the mixed attention module and semantic-aware layer normalization module to better exploit the semantic consistency for depth features generation. Moreover, we present a novel semantic weighted depth synthesis module, which generates adaptive depth interval for the current scene. Then we generate the final depth map by using a weighted combination of semantic-aware depth weights in different depth ranges. In this manner, we obtain a more accurate depth map. Finally, we conduct extensive experiments on indoor and outdoor datasets and demonstrate that the proposed DepthGAN achieves superior performance both on quantitative and visual effects in the depth generation task.

Keywords: Depth Generation; generative model; transformer;scene generation.

*Corresponding author

1. Introduction

With the rapid development of technologies of computer vision and computer graphics, 3D scene generation has become important in a variety of downstream applications, such as virtual scene construction, AR and VR, etc.

However, existing 3D generation methods mainly focus on generating a single object with the representation of point clouds [58, 33], voxels [63], meshes [56, 55] and implicit representations [39, 20], or optimizing the scene layout of retrieved 3D models for the scene construction [36, 11]. The limited fitting capability of 3D generation models and the complexity of object relations in 3D scenes make it extremely challenging to generate 3D representations of scenes containing diverse objects directly. Moreover, optimizing existing 3D models is computationally friendly while lacking flexibility. Hence, the solution to generating complex 3D scenes still remains an open problem. Compared with manually building 3D scenes with multiple objects, visual designers typically prefer controllable and simple input, such as 2D semantic layouts [25, 40, 38] or sketches [9, 21, 6]. However, due to the insufficient input information, it is impractical to straightly construct 3D scenes from the simplified 2D constraints above. Inspired by the works of depth estimation task [59, 60], the depth map is a viable 2.5D medium that measures the distance between the objects and the camera in stereoscopic space, and it can be regarded as the transition from 2D images to 3D scenes.

Therefore, we focus on a new task of generating an accurate and reasonable depth map with a simple semantic layout as input to further construct the 3D scene for visual designers. To the best of our knowledge, this is the first work that explores depth generation that only uses a semantic layout as input for constructing 3D scenes. With given camera parameters, the 3D scene can be precisely constructed once the depth map is reasonably generated, as shown in

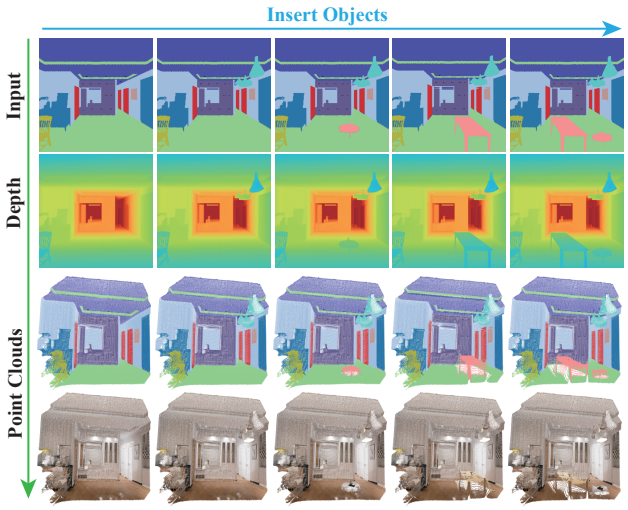


Figure 1. We generate different depth maps (row 2) by manipulating the input semantic layout (row 1). Given a fixed camera in the center of the room and the corresponding appearance of the scene and edited objects, we further construct the point clouds from generated depth maps with either the color of class labels (row 3) or appearance (row 4). For better visualization, we colorize the depth map, where blue is close and red is far.

Fig. 1. Since the depth map provides accurate geometric relations, the 3D scene can be fully represented within this lower-dimensional space.

For this purpose, we first conducted depth generation by previous Conv-based conditional image generation models [25, 40, 47] yet receiving unsatisfied results which include incorrect depth interval or improper depth structure. The receptive field of the convolution architecture is limited to a local scope [37] and the feature aggregation is confined to pixels inside the scope. Hence, most existing Conv-based methods for depth generation cannot accurately predict the global geometric correlation between different objects which makes the generated depth incoherent in terms of visual perception. In addition, existing GAN-based conditional generative models adopt a simple manner of convolution layers and nonlinear activation to obtain the output image from generated high-resolution features. Since the depth maps have more structural regularity than color images, such a simple layer cannot fully model the depth distribution, leading to stretched or squeezed depth interval and unsmoothed depth maps.

Accordingly, to address the limitations above, we propose the DepthGAN, which redefines depth generation as a feature generation and depth synthesis task. In the feature generation part (Sec. 3.1), to better generate global features in the semantic-guided generation, we propose a semantic-aware transformer block with mixed attention and semantic-aware layer normalization, efficiently improving the consistency between the generated feature and the semantic input. Moreover, we replace the output layer in the previous generative methods with a semantic weighted depth synthesis module (Sec. 3.2) to generate an accurate depth map. We first predict the depth intervals within a scene and then synthesize the final depth map by a weighted combination to integrate local and global features with semantic information. Equipped with the above modules, the proposed DepthGAN achieves state-of-the-art results on both indoor and outdoor scene datasets, demonstrating the effectiveness of our approach on depth generation. Furthermore, our DepthGAN can perform scene manipulation with a simple modification to the input, as shown in Fig. 1. Meanwhile, we also generate the appearance by proposed semantic-aware transformer blocks together with depth generation task. Thus, we can enable the scene generation from a simply handcrafted semantic layout, as shown in our *video demos*.

Overall, our contributions are summarized as follows:

- We propose a novel generation-synthesis approach for the new depth generation task with only the semantic layout as input. It provides an effective and controllable solution for 3D scene generation.
- We present a semantic-aware transformer block with mixed attention and semantic-aware layer normalization to take advantage of rich global information of depth and semantic layout for generating depth features.
- A semantic weighted depth synthesis scheme is introduced to generate the final depth map with the input of generated depth features, which is superior both in terms of quantitative metrics and visual effects.

2. Related Work

2.1. GAN-based Semantic Image Synthesis.

Generative Adversarial Networks [22] have achieved impressive results in unconditional [5, 29, 16] and conditional [25, 43, 28] image generation. Semantic image synthesis is a task that takes the semantic layout as input, which provides pixel-level class labels, and outputs a natural image with semantic guidance.

Pix2Pix [25] first introduces an encoder-decoder architecture and a patch-based discriminator to handle this problem. SPADE proposed a method to modulate the activations in the normalization layers using the semantic input to guide the generation direction, which frees the encoder block, enabling a coarse-to-fine generation. Following works including [64, 46, 38, 45] learn the normalization layers using style, semantic, or instance input. CC-FPSE [34] predicts conditional separated convolution kernels from the input semantic layout, and introduces a feature pyramid semantic-embedding discriminator for semantic alignment. OASIS [44] re-designed the discriminator

with a semantic segmentation network for semantic alignment. LGGAN [48, 47] proposed a local class-specific and a global image-level generator to learn a local-global feature generation. SCGAN [53] learned semantic vectors to parameterize spatially conditional convolution and normalization. While depth generation requires more global feature awareness, thus we introduce a cascade of transformer-based blocks for a coarse-to-fine depth feature generation.

2.2. Monocular Depth Estimation.

A depth map measures the spatial structure of a scene, which is a low-dimensional but efficient representation of the 3D scene. Monocular depth estimation [17, 31, 19, 51], mainly focuses on regressing dense depth maps from images. Nevertheless, poor edge quality and the lack of global information are common problems of CNN-based depth estimation models. [65] explicitly introduced a pre-trained semantic segmentation network to guide depth boundaries due to the high quality of edges in the semantic map. In addition to CNN-based structures, generative models [1, 8] and vision transformers [41] have also been applied to depth estimation tasks. Recently, [3, 4] perform a global statistical analysis on depth bins to further predict the depth map in a classification-regression manner. In the task of depth estimation, the dense depth map can be further used to reconstruct 3D scenes, which inspires us a time-saving choice to generate a 3D scene by the depth map. Nevertheless, color images are required as input in the depth estimation task, obtaining 3D scenes in this way is difficult to meet the requirements of the simplicity of manipulation and variability for visual designers. To this end, we propose a new task of depth generation utilizing only the semantic layout as input, which is different from the depth estimation task with the input of color images.

2.3. Vision Transformers.

The seminal work [13] propose a pure transformer [7]-based architecture for discriminative vision tasks, which enables a global feature aggregation and extraction in images. Cvt [57] introduces convolutions into vision transformers to enhance the local attention. Swin transformer combines the local and global attention by calculating the attention in a local shifted window, leading to a huge improvement in vision transformers. Recently, researchers begin to explore the migration of using vision transformers in GANs for generating better global features in complex images. [32, 26] have improved rapidly in image generation tasks due to the superior global feature aggregation capability of multi-head self-attentions(MSAs). However, the generation quality of these methods is not proportional to the time consumption due to the quadric computing efficiency of the default vision transformers, which makes it difficult for a high-resolution generation. Recent works [12, 35, 50] propose to calculate MSAs in local windows, leading to the linear computational efficiency. [61] prove the feasibility of using block-wised attention for unconditional high-resolution image generation. In this work, we observe that exploiting vision transformers with more global information is suitable for the new conditional depth generation task.

3. Method

As illustrated in Fig. 2, we present a novel depth generation architecture, DepthGAN, which consists of a depth feature generation stage (Sec. 3.1) and a depth map synthesis stage (Sec. 3.2). Starting from a one-hot semantic layout $S \in \mathbb{N}^{H\times W\times C}$, we first adopt a cascade of semantic-aware transformer blocks to generate the depth feature $F \in \mathbb{R}^{H\times W\times E}$. Then we utilize F to generate the adaptive depth interval and apply a semantic-weighted com-

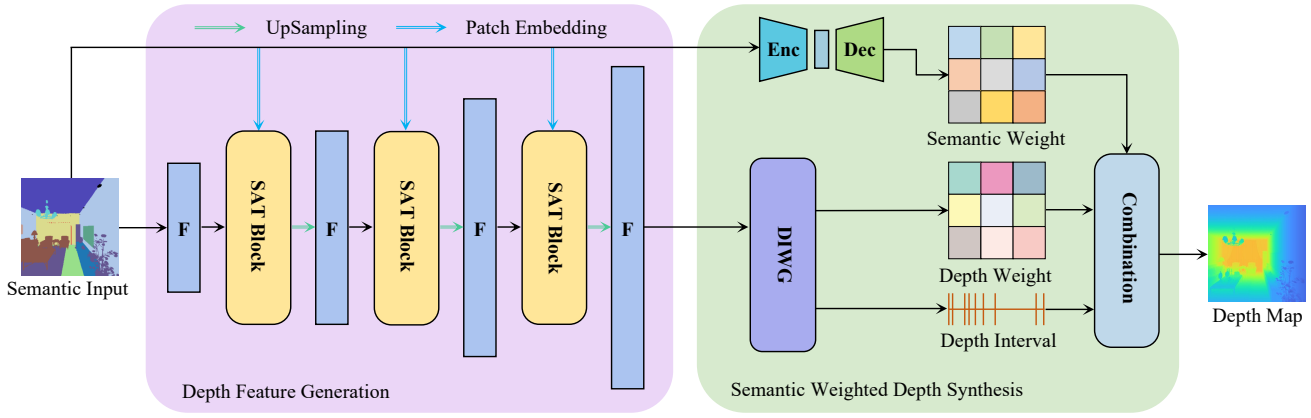


Figure 2. **Overview of our DepthGAN.** Our framework involves two stages: (1) depth feature generation and (2) semantic weighted depth synthesis. We first employ a cascade of semantic-aware transformer (SAT) blocks to generate depth features with semantic alignment. Then we adopt an encoder-decoder to generate the semantic weight map and a DIWG module to generate the depth interval and the depth weight map. Finally, the depth map is synthesized through a semantic weighted combination.

bination to obtain the final depth map $D \in \mathbb{R}^{H \times W}$, which is semantically aligned with the semantic layout S.

3.1. Depth Feature Generation

Unlike generating the appearance, depth map generation mainly focuses on global features, such as the geometric and spatial structure within the scene. In order to capture global information, we construct an architecture comprised of a series of Swin transformer [35] blocks as our baseline to better generate the global attention features. It takes the downsampled low-resolution semantic layout as the input of this stage and generates the depth feature with upsampling in a coarse-to-fine manner.

However, the baseline method cannot effectively align the generated features with the input semantic layout due to the lack of semantic constraints in the generation process. To address this issue, we propose a semantic-aware transformer (SAT) block, which introduces a semantic positional encoding (SPE), a mixed attention module, and a semantic-aware layer normalization (SALN) module to guide the direction of feature generation, as shown in Fig. 3 (a).

Semantic Positional Encoding. In the SAT block, we first aim to better let the layers know the semantic position information on each input scale. Thus, we utilize a learned semantic embedding from the semantic layout as a positional encoding, as shown in Fig. 3 (a).

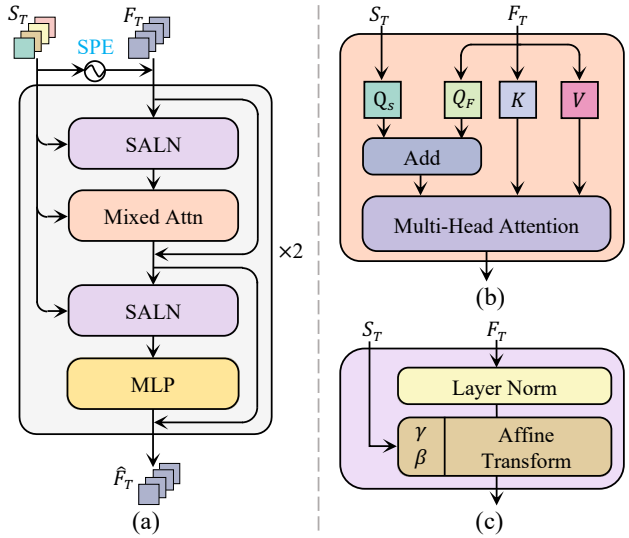


Figure 3. (a) The semantic-aware transformer (SAT) block. (b) The proposed mixed attention. (c) The proposed semantic-aware layer normalization (SALN).

For the input feature maps F^* of shape $\mathbb{R}^{H_F \times W_F \times E_F}$ in each input scale, where E_F, H_F, W_F are the spatial resolution, we embed the one-hot semantic input S of shape $\mathbb{N}^{H \times W \times C}$ to the same scale as F^* by learned *convolution kernels* with different stride parameters, denoted as S^*:

$$S^* = \text{Conv}(S). \tag{1}$$

Thus the SPE is adaptive for different feature scales. We then add the embedded semantic input S^* to the input feature maps F^*, enabling the SAT block to perceive the global semantic information for each pixel. Different from the learned positional encoding in the default transformer blocks which encodes the relative position of pixels, our SPE can involove the semantic information and further improve the feature generation quality and semantic alignment.

Mixed Attention. Although the baseline utilizes self-attention by calculating queries, keys, and values from the features, this method ignores the interaction between features and semantics. To this end, we propose a simple yet effective strategy, named Mixed Attention, as shown in Fig. 3 (b). Instead of calculating the attention between tokens of features, we adopt additional semantic queries:

$$\text{Mixed Attn} = \text{Softmax}(\frac{(Q_F + Q_S)K_F^T}{\sqrt{d_k}} + E)V_F, \tag{2}$$

where Q_F, K_F, V_F represent the query, key, and value matrices projected by the features, and Q_S is the query matrix from the semantic input. The relative positional encoding E is added as a bias term.

Compared to self-attention in Swin Transformer, our mixed attention enables the feature aggregation between features and semantics at the same time, leading to better semantic-aware generation.

Semantic-aware Layer Normalization. To better match semantic features and depth features in the SAT block, we propose the semantic-aware layer normalization to learn a parameterized affine transformation and fuse the semantic information to the features, as shown in Fig. 3 (c). Given the input feature tokens F_T, the output tokens $\hat{F}_T$ are calculated as:

$$\hat{F}_T = \frac{\gamma(S_T)}{\sigma} \odot (F_T - \mu) + \beta(S_T), \tag{3}$$

where γ, β are vectors learned by a simple MLP-ReLU-MLP architecture with the semantic tokens S_T. Here $\odot$ is the element-wise multiplication between two vectors. μ and σ denote the mean and standard deviation of F_T.

With learned scaling and bias vectors γ, β, the affine transformation is adaptive with the semantic input and varies with respect to different token positions, which facilitates the matching of different features without introducing unstable training.

3.2. Semantic Weighted Depth Synthesis

Especially, the depth map has more structural regularity in the feature distribution. However, the simple output manner in previous GAN-based image generation methods lacks

the capability to model the accurate depth map. Inspired by the Adabins [3] from the depth estimation task, we propose the semantic weighted depth synthesis (SWDS) stage, which generates the depth interval from the depth features of the previous stage and conducts a semantic weighted depth synthesis scheme to obtain the final depth map, as shown in Fig 2.

In this stage, we first design a depth interval and weight generation (DIWG) module to enable the depth interval and weight generation for the scene. Meanwhile, we propose to use an encoder-decoder architecture to compute the semantic weight map W from S and better utilize the semantic input. Finally, we utilize a semantic weighted combination module to fuse them and synthesize the final depth map.

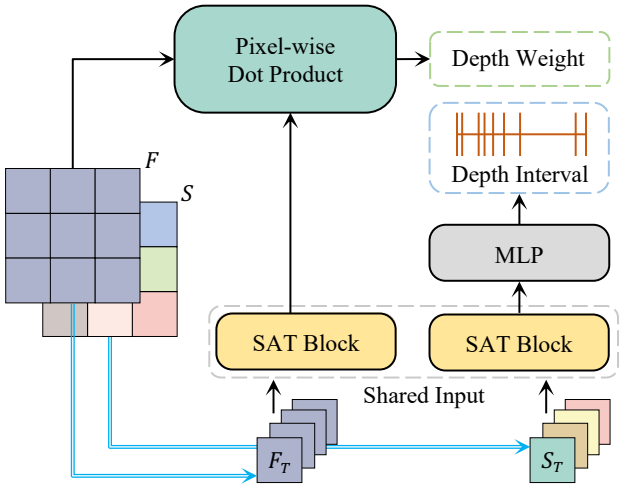


Figure 4. The proposed DIWG module. The module takes the feature map and semantic map as input and outputs the depth interval and depth weight map.

As shown in Fig 4, the DIWG module first embeds the input feature F and the semantic input S into patches, denoted as F_T and S_T. Then we adopt two SAT blocks to enable semantic-aware feature generation. Note that, we do not add a global positional encoding here since the window size in the SAT block is set to be the same as the embedded feature map, thus the relative positional encoding here can be regarded as a global one. The output embedding from one of SAT blocks is projected by a linear perceptron with *softmax* to yield an N bins length vector b. Like Adabins [3], the bin centers $c(b)$ are calculated via post-process:

$$c(b_i) = d_{min} + (d_{max} - d_{min})(\frac{b_i}{2} + \sum_{j=1}^{i-1} b_j), \quad (4)$$

where $c(b_i)$ is the center depth of the i^{th} bins. d_{max} and d_{min} are the maximum and the minimum depth values of the dataset.

Meanwhile, we obtain the depth weight map via a pixel-wise dot product between the generated feature embedding of another SAT block and the input feature F. Note that, the depth weight map contains rich local-global feature similarities while serving as a key-query process.

On the other hand, we compute the semantic weight map W with the encoder-decoder architecture. Then, we apply an element-wise multiplication between W and the depth weight map to obtain a semantic-aware depth weighted map, which aggregates the additional semantic information for a weighted generation. Next, the semantic-aware depth weighted map is converted to a weighted probability depth distribution map P^W via *softmax*. Finally, the depth value for each pixel is calculated from a weighted combination with the corresponding probability distribution: $\hat{d} = \sum_i c(b_i) p_i^W$.

In the SWDS stage, we fuse the semantic information to the depth map and disentangle the bins generation with the depth weight map generation by two separate SAT blocks, enabling more accurate and reasonable depth maps.

3.3. Loss Functions

The generator and the discriminator are trained alternatively, where we adopt the hinge loss in the discriminator for distinguishing real/fake. The generator is optimized by multiple losses, including the hinge-based adversarial loss, discriminator feature matching loss $L_{FM}(\hat{x}, x)$, and the perceptual loss $L_P(\hat{x}, x)$, following the previous works [27, 40, 52]:

$$\begin{aligned} L_D &= -\mathbb{E}_{x,S}[\mathrm{H(D(x, S))}] - \mathbb{E}_{\hat{x},S}[\mathrm{H}(D(\hat{x}, S))], \\ L_G &= -\mathbb{E}_{\hat{x},S}[D(\hat{x}, S)] + \lambda_P \mathbb{E}_{\hat{x},S} L_P(\hat{x}, x) \\ &\quad + \lambda_{FM} \mathbb{E}_{\hat{x},S} L_{FM}(\hat{x}, S), \end{aligned} \quad (5)$$

where x is a real depth map, $\hat{x}$ is a generated depth map, and S is the semantic layout. λ_P, λ_{FM} denote the weights for the perceptual loss and feature matching loss. H is the hinge function, $\lambda = 1$ if I is a real image and -1 if I is a generated image:

$$H(I) = \min(0, -1 + \lambda I). \quad (6)$$

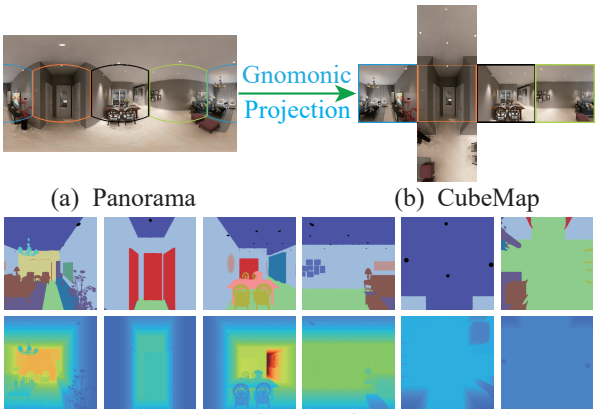
4. Experiments

4.1. Implementation

Datasets. We benchmark our approach over Structured3D [62], Stanford2D3D [2], and Visual KITTI (VKITTI) [18] datasets. Here we show details about the datasets following:

- **Structured3D** is rendered with synthetic scenes in panorama images. The geometric structure is distorted in the panorama images on the sphere grid, and accurate depth generation is difficult with conventional convolution kernels [10]. Therefore, we re-project the panorama images in Structured3D to perspective views by reverse gnomonic projection [49], as shown in Fig.

5. Following the official split, we choose scene 0 to scene 2999 for training, scene 3000 to scene 3249 for validation, and scene 3250 to scene 3499 for testing, resulting in 109494 training images and 10122 testing images of virtual indoor scenes.



(a) Panorama (b) CubeMap

(c) Semantic and Depth pairs of 6 perspective images

Figure 5. Explanation of the re-projection in Structured3D. (a) The panoramic image on the sphere grid. (b) Cubemap by reverse gnomonic projection. (c) Semantic layout and depth map pairs of 6 perspective views in the scene, viewed in color. Note that the same color box in (a) and (b) represents the same part from a spherical view to a perspective view.

- **Stanford2D3D** is scanned with RGB-D cameras in the real-world scene with both perspective and panorama images. Following the official split, we choose the perspective images in areas 1, 2, 3, 4, and 6 for training and area 5 for testing. Stanford2D3D contains 52093 training images and 17593 testing images of real-world indoor scenes.

- **VKITTI** is a photo-realistic synthetic video dataset designed to learn and evaluate computer vision models for several video understanding tasks: object detection and multi-object tracking, scene-level and instance-level semantic segmentation, optical flow, and depth estimation. We choose scenes 0, 2, 18, and 20 for training and scene 6 for testing. Thus, we get 18560 training images and 2700 testing images of outdoor scenes. The semantic labels are obtained from the provided instance labels by the color mapping in each scene provided in the dataset.

The minimum depth value is set to 0, and the maximum depth value is 655.35 meters for VKITTI while is 10 meters for the other indoor datasets.

Evaluation Metrics. We adopt Fréchet Inception Distance(FID) [23] to measure the Wasserstein-2 distance between the distribution of generated depth map and that of the ground truth depth. Moreover, seven standard evaluation metrics in depth estimation tasks [14, 3] are evaluated for depth accuracy, including mean absolute error (MAE), root mean square error (RMSE), absolute relative error (Abs Rel), square relative error (SqRel) and threshold percentage (δ^n):

- MAE $= \frac{1}{N}\sum_{i=1}^{N}|d_i - \hat{d}_i|$;
- RMSE $= \sqrt{\frac{1}{N}\sum_{i=1}^{N}|d_i - \hat{d}_i|^2}$;
- AbsRel $= \frac{1}{N}\sum_{i=1}^{N}|d_i - \hat{d}_i|/\hat{d}_i$;
- SqRel $= \frac{1}{N}\sum_{i=1}^{N}|d_i^2 - \hat{d}_i^2|/\hat{d}_i$;
- Threshold percentage δ_n is the percentage of pixels satisfying $max(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}) < 1.25^n, n \in \{1, 2, 3\}$,

where d and $\hat{d}$ are ground truth depth and generated depth respectively.

Additionally, we calculate the PSNR of generated depth maps:

$$\text{PSNR} = 20\log_{10}\frac{\text{MAX}_d}{\text{RMSE}}, \tag{7}$$

where MAX_d is the max depth value of the dataset for the depth generation.

Training and Testing Details. Different from the task of image generation, the depth values are stored as 32-bit float values, which are then normalized to [0, 255.0] by dividing the maximum depth value of the scenes. At the testing, the FID is calculated directly by the normalized depth values ranging from 0 to 255.0, while other metrics are calculated by re-scaling generated float depth values back to the original format of the dataset without losing accuracy. For the

Table 1. Detail architecture of our DepthGAN. Input size and Dim are the shapes of the input feature map and semantic embedding in the SAT block. SAT-8 means an SAT block with an input resolution of 8×8. In the SAT block, h is the number of heads in MSAs, d is the depth of SAT blocks, and w is the window size of mixed attention. In the SWDS module, p is the patch size of the patch embedding for both the feature map and the semantic layout, and MLP-256 is the 256-dimensional MLP for depth interval. We use bilinear upsampling for the upsampling layers.

-

Input size	Dim	Module	Architecture	Up
8×8	512	SAT-8	{h-16, d-2, w-8}	✓
16×16	512	SAT-16	{h-16, d-2, w-8}	✓
32×32	512	SAT-32	{h-16, d-2, w-8}	✓
64×64	256	SAT-64	{h-16, d-2, w-8}	✓
128×128	128	SAT-128	{h-8, d-2, w-8}	✓
256×256	64	SAT-256	{h-4, d-2, w-8}	
256×256	64	SWDS	p-16 SAT-16, MLP-256 SAT-16	

discriminator, we apply the Spectral Norm to all the layers. We adopt the Adam optimizer [30] with a learning rate 0.0001 for the generator and 0.0004 for the discriminator following TTUR [24], and set $\beta_1 = 0$ and $\beta_2 = 0.999$. The weight for the perceptual loss is 10. Our models are trained on 8 TITAN RTX GPUs, with a batch size of 32. The training and generated resolution is 256×256 for Structured3D and Stanford2D3D datasets, and 256×512 for the VKITTI dataset. All results presented are obtained after training 50 epochs.

Network Architecture. In this section, we provide the detailed model architecture for a 256×256 resolution depth generation, as shown in Tab. 1.

4.2. Comparison Experiments

In this sub-section, we provide quantitative and visual comparisons to prove the effectiveness of our DepthGAN. We compare with previous semantic image synthesis methods including Pix2pixHD [25], SPADE [40], CC-FPSE [34], LG-GAN [48], SEAN [64], OASIS [44], and SAFM [38] with the same training strategy. For OASIS, we use their default setting but remove the 3D noise part to avoid the randomness in the generated depth map for better accuracy.

Note that in the depth generation, we use a one-hot semantic layout as input for the accurate evaluation, unlike the input noise map commonly used in semantic image synthesis tasks for a multi-style generation. Thus we can learn a unique depth distribution from an input semantic layout without randomness. Meanwhile, using a semantic layout instead of noise as input enables a fully controllable depth generation. As shown in Fig. 1 and the video demo, the generated depth map only changes among the edited objects, while the remaining parts keep consistent.

Quantitative Comparison. Tab. 2 shows the performance of the depth maps generated by our approach and the competitors on the proposed new tasks. Our approach leads to decisive improvement and performs consistently better than previous approaches, which demonstrates the effectiveness of the proposed approach. With the generation-synthesis strategy, we generate depth maps with more accurate depth values and higher PSNR. In particular, our method improves MAE by around 20% and PSNR by around 5% for the average of three datasets over the second-best competitor, which means the proposed strategy is more capable of generating accurate depth than the convolution-nonlinear manner in the previous methods for this task. Moreover, our generated depth maps outperform the competitors by 28% on the FID

Table 2. Comparison of performance with previous approaches on multiple datasets. The better performances are in **bold**.

Dataset	Method	MAE ↓	AbsRel ↓	SqRel ↓	RMSE ↓	δ^1 ↑	δ^2 ↑	δ^3 ↑	PSNR ↑	FID ↓
Structured3D	Pix2pixHD	0.1587	0.1325	0.1162	0.2062	84.52	93.73	96.64	21.63	128.20
	SPADE	0.1366	0.1447	0.0781	0.1536	86.61	94.51	96.93	22.42	119.59
	CC-FPSE	0.0946	0.0903	0.0353	0.1297	91.46	97.68	99.04	27.19	87.62
	LGGAN	0.1362	0.1229	0.0893	0.1489	88.03	95.63	97.56	23.41	114.02
	SEAN	0.1037	0.0863	0.0481	0.1268	89.47	97.05	98.75	26.69	75.34
	OASIS	0.1199	0.1173	0.0532	0.1631	87.47	95.83	98.12	24.40	166.51
	SAFM	0.0981	0.0826	0.0419	0.1187	90.64	97.45	98.61	27.79	61.58
	Ours	**0.0613**	**0.0590**	**0.0228**	**0.0888**	**95.37**	**98.67**	**99.40**	**30.53**	**37.38**
Stanford2D3D	Pix2pixHD	0.5424	0.2985	0.5507	0.7801	69.36	84.97	90.97	17.25	335.98
	SPADE	0.5820	0.2981	0.3662	0.7910	60.54	83.07	92.10	19.37	201.53
	CC-FPSE	0.3662	0.1822	0.1466	0.5385	76.66	92.70	97.20	23.88	185.87
	LGGAN	0.3381	0.1866	0.1435	0.5637	77.65	92.92	97.25	22.23	254.65
	SEAN	0.4208	0.2068	0.1883	0.6057	72.28	90.99	96.56	21.67	157.37
	OASIS	0.4037	0.2441	0.2635	0.6252	71.23	89.79	97.43	22.79	172.55
	SAFM	0.3788	0.1927	0.1604	0.5559	74.17	91.86	96.98	22.19	238.06
	Ours	**0.2831**	**0.1380**	**0.1168**	**0.4898**	**83.90**	**95.21**	**98.19**	**23.95**	**130.45**
VKITTI	Pix2pixHD	24.689	0.3989	31.784	53.373	52.47	81.16	92.92	21.74	668.24
	SPADE	20.014	0.3384	13.675	38.607	55.64	80.95	91.49	24.60	510.08
	CC-FPSE	18.760	0.2869	11.559	35.376	64.63	84.90	92.67	25.40	764.08
	LGGAN	15.089	0.2605	12.331	34.091	67.45	88.34	94.26	25.70	470.11
	SEAN	18.719	0.2996	16.393	38.919	66.52	84.48	91.21	24.55	569.56
	OASIS	13.214	0.2657	8.439	30.263	64.40	86.71	93.17	26.68	493.30
	SAFM	15.220	0.2548	10.754	31.702	64.17	87.15	93.42	26.40	454.41
	Ours	**10.973**	**0.2315**	**7.181**	**26.388**	**69.47**	**89.16**	**94.55**	**27.02**	**291.78**

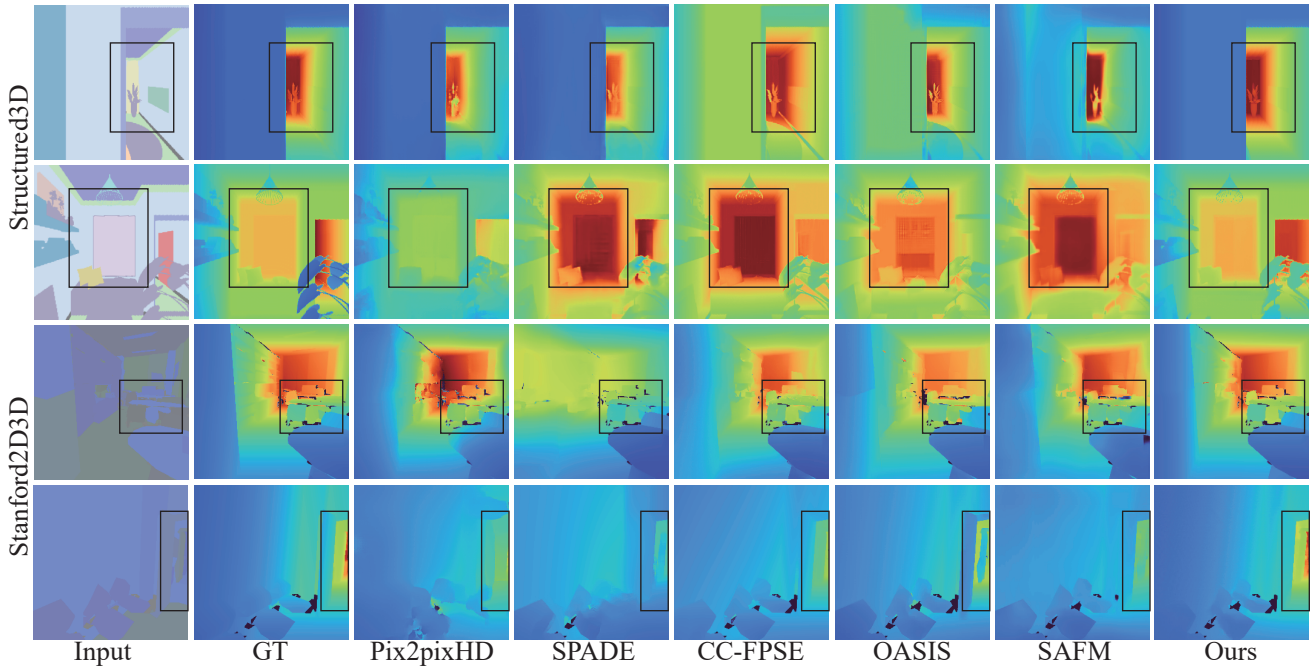


Figure 6. Depth maps comparison on Structured3D and Stanford2D3D. Blue is close and red is far.

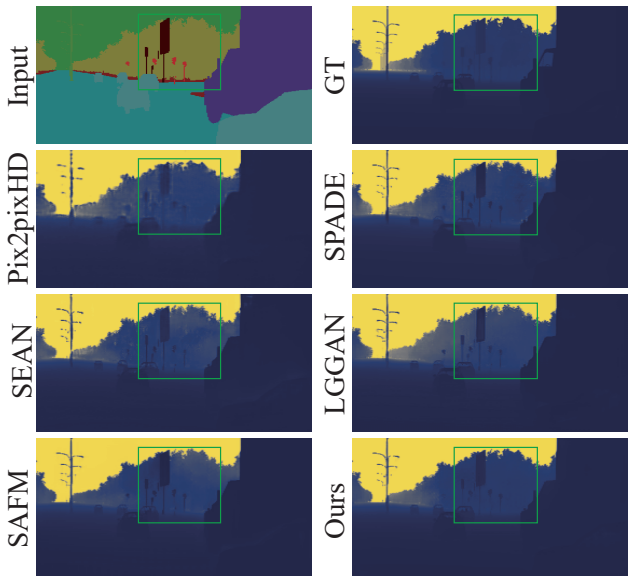


Figure 7. Depth maps comparison on VKITTI. Blue is close and yellow is far.

score, proving that the distribution of generated depth maps is closer to the ground truth distribution.

Visual Comparison. As shown in Fig. 6 and Fig. 7, our approach shows compelling quality in generating more accurate depth maps with reasonable structure correlation, better matching the ground truth depth distribution. With the global depth features generated by the SAT block, our generated depth maps can better represent the structure of complex scenes, such as chairs in rows 3 of Fig 6. Especially for small and far semantic regions, our approach can generate correct depth values, see the plants of row 1 and the door of row 4 in Fig. 6. Moreover, since we generate the depth interval for the scene and utilize a semantic-aware weighted combination, our generated depth maps show more accurate geometric correlation and can better capture the depth variation within the semantic regions, see row 2 in Fig. 6 and the depth of trees in Fig. 7.

4.3. Ablation Study

We conduct experiments on the Structured3D dataset to verify the effectiveness of each component in our method.

Table 3. Ablation study on the network architecture. Starting from the baseline architecture, we prove the effectiveness of each proposed component.

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
Baseline	0.1709	0.2086	17.65	307.16
+ SPE	0.1382	0.1749	20.81	226.94
+ SALN	0.0843	0.1015	26.90	77.24
+ Mixed Attn	0.0755	0.0826	29.58	50.11
+ SWDS (Ours)	**0.0613**	**0.0590**	**30.53**	**37.38**

Main Ablation. As shown in Tab. 3, starting from a cascade of Swin blocks as our baseline method, we gradually add each component to the framework. Compared with the baseline, adding a semantic position embedding (SPE) on each scale brings improvement because it encodes extra se-

mantic positions. The semantic-aware layer normalization (SALN) greatly improves the performance and training stability by matching semantic and depth features. Moreover, the mixed attention enables feature aggregation among different features at the same time. Finally, replacing the output layer with the proposed semantic weighted depth synthesis (SWDS) module makes the generated depth values more accurate.

Table 4. Comparison of performance with the respect to the choice of the discriminator.

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
Multiscale	**0.0613**	**0.0590**	**30.13**	**37.38**
FPSE	0.0736	0.0739	29.02	52.39
OASIS	0.9640	0.0878	26.59	66.41

Ablation on the Discriminator. In Tab. 4, we explore the discriminator choice for depth generation. We obtain better quantitative results by the multiscale design [25]. The FPSE [34] and OASIS [44] perform worse because the pixel-wised semantic alignment in the discriminator leads to clear semantic boundaries while introducing a drastic change in the depth of adjacent objects, which is also shown in row 1 of the 5 and 6 columns in Fig. 6.

Ablation on the SWDS. In Tab. 5, we replace the output layer in the semantic image synthesis approaches by our proposed SWDS. The improvements indicate the effectiveness of the SWDS module. Meanwhile, we observe that the worse performance the original approach does, the better improvement the SWDS achieves.

Table 5. Different methods with SWDS, denoted by †.

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
SPADE	0.1366	0.1447	22.42	119.59
SPADE†	0.1225	0.1208	24.96	108.35
OASIS	0.1199	0.1173	24.40	166.51
OASIS †	0.1084	0.1145	25.82	104.67
SEAN	0.1037	0.0863	26.69	75.34
SEAN†	0.0921	0.0789	28.46	62.57

Ablation on Adabins. We compare our proposed SWDS with the original adabins [3] design for the depth synthesis.

We note that there are two main differences. On the one hand, we disentangle the bins generation from the depth weight map generation. In detail, we utilize an SAT block and the following MLP to generate the depth interval and use another SAT block to generate the depth map. Adabins simply use a transformer to predict both depth bins and the weight, which leads to entanglement between different features. On the other hand, we propose to use a semantic weight map by the encoder-decoder architecture for the semantic weighted depth synthesis. The semantic weight map is especially suitable for the task of synthesizing depth maps using the semantic layout, which is not involved in Adabins.

Table 6. Comparison of performance with our SWDS and original adabins.

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
Adabins	0.0756	0.0671	29.69	48.24
SWDS	**0.0613**	**0.0590**	**30.53**	**37.38**

As a result, our generated depth maps obtain more accurate depth interval, enabling better performance in the depth evaluation metrics as shown in Tab. 6. Moreover, the proposed semantic weighted synthesis also enhances the quality of semantic alignment in the generated depth maps, leading to a better FID score.

Table 7. Comparison of performance with the respect to the choice of the loss function. Per, SSIM, and SI are perceptual loss, structural similarity loss, and scale-invariant loss respectively.

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
SSIM	0.0807	0.7781	28.69	45.89
Per	**0.0613**	**0.0590**	**30.53**	**37.78**
Per+SI	0.0781	0.0689	29.36	56.95

Ablation on Loss Functions. We conduct additional experiments on the loss functions commonly adopted in depth estimation tasks for depth generation, as shown in Tab. 7. Replacing the perceptual loss (Per) with the Structural Similarity loss (SSIM) [54] leads to worse results in our method since the local window size in the SSIM loss functions leads to block artifacts. Additionally, adding a Scare-Invariant (SI) loss [15] introduces a threadlike unsmooth issue during the adversarial training, which also reduces the performance.

4.4. MultiModel

Furthermore, we extend our method to generate both depth and appearance with smeantic input. Specifically, we first introduce appearance supervision to our approach and train a model with two branches for depth and appearance generation respectively. Then, we simply adopt the same design for the two branches with cascades SAT blocks and upsampling. As shown in Fig. 9, we share the SAT blocks in the low-resolution generation since the salient features such as edges are mainly generated at low resolutions. Finally, we use the SWDS module for the depth synthesis and a Conv-Tanh layer for the appearance generation. As shown in Fig. 8, we can generate 3D point cloud scene with different appearances using only a simple semantic layout as input, which better meets the visual designers' requirements. Moreover, the shared generation can supervise the

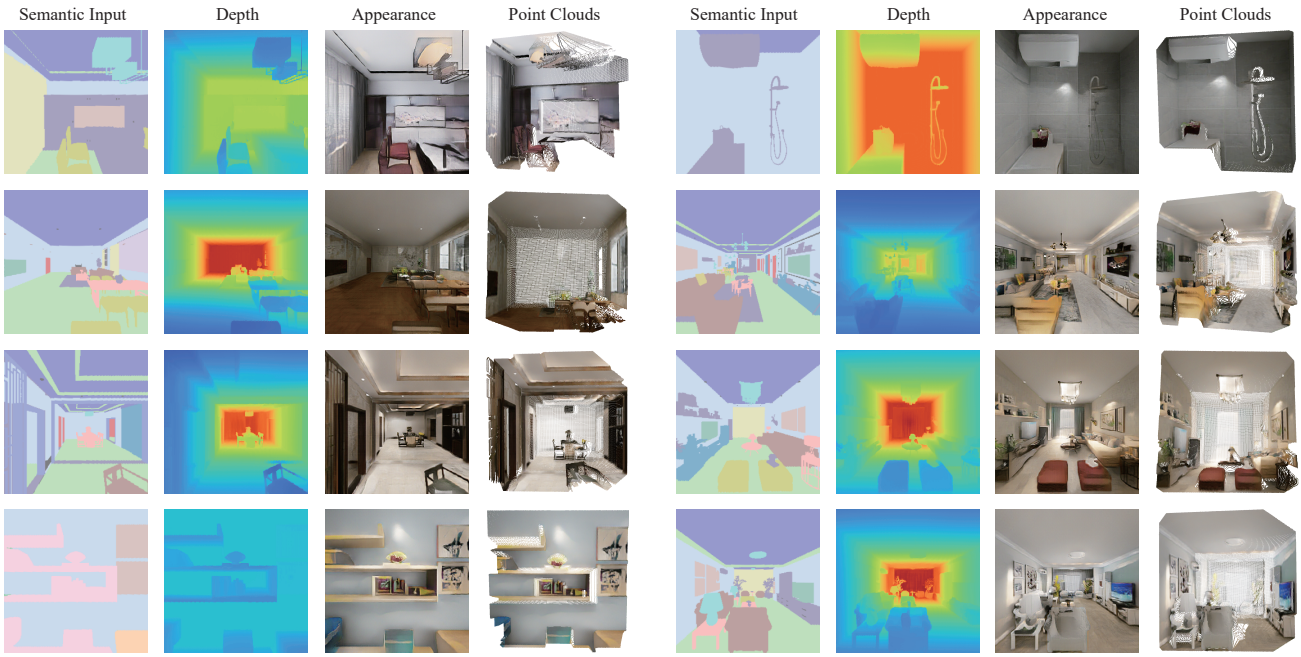


Figure 8. Samples by our two-branch model for generating appearance and depth at the same time. The point clouds are constructed with the generated appearance. For the depth map, blue is close and red is far.

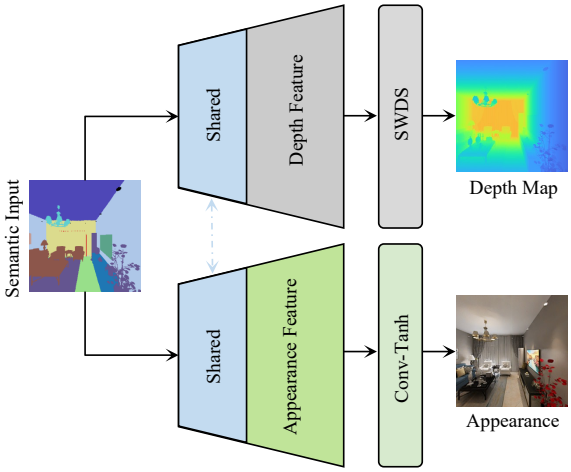


Figure 9. The depth-appearance generation architecture. We take as input the semantic layout and output the depth map and the appearance simultaneously.

depth features with appearance features, which also helps improve the depth generation quality (see Tab. 8). Please see more results with the handcrafted semantic layout as input in the supplementary video demos.

In addition, to further verify the influence of the appearance branch on the generated depth quality, we adopt different discriminators in the appearance branch, as shown in the Tab. 9. The multiscale discriminator in the appearance branch achieves better depth quality, while FPSE and OASIS discriminators perform similarly to our single-branch model even with the help of appearance, see Tab. 8. The reason is that the over-emphasis on semantic boundaries in the shared blocks will lead to a depth disparity at the object edges, see Tab. 4. Thus with the multiscale discriminator in the appearance branch, our depth generation achieves more continuous depth quality. The results also show the difference between depth generation and image generation.

Table 8. Comparison of performance between the depth generation with the depth-appearance generation, denoted by D and D-A respectively.

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
D	0.0613	0.0590	30.53	37.38
D-A	**0.0598**	**0.0582**	**31.01**	**32.35**

Table 9. Comparison of our appearance branch with different appearance discriminators.

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
FPSE	0.0614	0.0591	31.15	36.26
OASIS	0.0618	0.0596	**31.25**	36.47
Multiscale	**0.0598**	**0.0582**	31.01	**32.35**

4.5. 3D Scene Generation

Apart from generating depth maps, our method can generate the 3D scene point cloud to further demonstrate the effectiveness of our depth generation. Given the depth map

of a perspective view and a fixed camera intrinsic parameter, we can simply generate the point clouds through a *pinhole camera model* as shown in Fig. 1. For better visualization, we project the generated depth maps within a scene in the Structured3D dataset back to the panorama image and then construct the whole 360° scene with given camera parameters. As shown in Fig. 10, the accurate geometric details and the flatness of the wall show the compelling quality of our generated depth maps.

Figure 10. The constructed point clouds by our generated depth maps. We use the appearances from the dataset and crop the ceiling for visualization.

4.6. The difference with depth estimation

Here we explain the key differences between depth generation with depth estimation. Depth estimation models extract rich features of the input image and utilize these features to guide depth prediction. But with a semantic layout as input, we cannot extract enough features. Thus we need adversarial training to guide the model to generate the features. We tried to retrain depth estimation models, such as Adabins [3], Midas [42] and DPT [41] using a semantic layout as input, but do not receive satisfying output. Thus, depth generation is quite different from depth estimation, we can not use a depth estimation model to predict a dense depth map from a semantic layout. With the help of adversarial training, the depth generation model can generate a depth map from sparse features in the semantic layout in a coarse-to-fine manner.

4.7. Limitation

For one thing, with a 256×256 resolution depth generation, the constructed sparse point clouds contain 65536 points. Generating point clouds with higher resolution is time-consuming for training. For another thing, the depth map measures the distance between the surface of the objects and the camera. The regions that are not visible to the camera can not be constructed by our method, thus the point clouds are partial. To further address the occlusion issue for 3D modeling, an interesting future work is to further conduct an additional point cloud completion module from the partial point clouds generated by our method.

Another issue is for the users, the shape of the actual drawn object is sometimes not as standard as objects in the training set, especially for indoor objects. That will also lead to artifacts. Our future work is to increase the robustness of the model to irregular objects.

5. Conclusion

We propose a novel method dubbed DepthGAN for the proposed depth generation task with the input of semantic layout, which provides an effective and controllable solution for complex 3D scene generation for the first time. First, we build a cascade of semantic-aware transformer blocks with proposed semantic-aware layer normalization and mixed attention, enabling a semantic-based depth feature generation. The generated depth features are then utilized to synthesize the depth map using our proposed semantic weighted depth synthesis module. Extensive evaluations are verified on multiple datasets and both quantitative and qualitative results demonstrate that our approach achieves valid generation of the depth maps and 3D scenes. Furthermore, our method can perform scene manipulation by simply editing the layout input, which is crucial for visual designers. We will also explore generating more 3D representations such as meshes and implicit functions from a semantic layout in future works.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (U2003109, U21A20515, 62102393, 62202076), the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA23090304), China Postdoctoral Science Foundation (2021M703162), Natural Science Foundation of Chongqing (CSTB2022NSCQ-MSX0924), the Youth Innovation Promotion Association of the Chinese Academy of Sciences (Y201935), the State Key Laboratory of Robotics and Systems (HIT) (SKLRS-2022-KF-11), and the Fundamental Research Funds for the Central Universities.

References

[1] F. Aleotti, F. Tosi, M. Poggi, and S. Mattoccia. Generative adversarial networks for unsupervised monocular depth prediction. In *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part I*, volume 11129 of *Lecture Notes in Computer Science*, pages 337–354. Springer, 2018. 3

[2] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 5

[3] S. F. Bhat, I. Alhashim, and P. Wonka. Adabins: Depth estimation using adaptive bins. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 4009–4018. Computer Vision Foundation / IEEE, 2021. 3, 5, 6, 9, 11

[4] S. F. Bhat, I. Alhashim, and P. Wonka. Localbins: Improving depth estimation by learning local distributions. *arXiv preprint arXiv:2203.15132*, 2022. 3

[5] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 2

[6] K. Brodt and M. Bessmeltsev. Sketch2pose: estimating a 3d character pose from a bitmap sketch. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022. 1

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[8] P. Chakravarty, P. Narayanan, and T. Roussel. GEN-SLAM: generative modeling for monocular simultaneous localization and mapping. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pages 147–153. IEEE, 2019. 3

[9] W. Chen and J. Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, 2018. 1

[10] T. Cohen, M. Geiger, J. Köhler, and M. Welling. Convolutional networks for spherical signals. *arXiv preprint arXiv:1709.04893*, 2017. 5

[11] H. Dhamo, F. Manhardt, N. Navab, and F. Tombari. Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16352–16361, 2021. 1

[12] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *arXiv preprint arXiv:2107.00652*, 2021. 3

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 3

[14] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2366–2374, 2014. 6

[15] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 9

[16] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 12873–12883. Computer Vision Foundation / IEEE, 2021. 2

[17] J. M. Facil, B. Ummenhofer, H. Zhou, L. Montesano, T. Brox, and J. Civera. Cam-convs: Camera-aware multi-scale convolutions for single-view depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11826–11835, 2019. 3

[18] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 5

[19] R. Garg, V. K. Bg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016. 3

[20] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020. 1

[21] A. Ghosh, R. Zhang, P. K. Dokania, O. Wang, A. A. Efros, P. H. Torr, and E. Shechtman. Interactive sketch & fill: Multiclass sketch-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1171–1180, 2019. 1

[22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014. 2

[23] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[24] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7

[25] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 5967–5976. IEEE Computer Society, 2017. 1, 2, 7, 9

[26] Y. Jiang, S. Chang, and Z. Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, 34:14745–14758, 2021. 3

[27] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 5

[28] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018. 2

[29] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceed-*

ings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4401–4410, 2019. 2

[30] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[31] J. H. Lee, M. Han, D. W. Ko, and I. H. Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *CoRR*, abs/1907.10326, 2019. 3

[32] K. Lee, H. Chang, L. Jiang, H. Zhang, Z. Tu, and C. Liu. Vitgan: Training gans with vision transformers. *CoRR*, abs/2107.04589, 2021. 3

[33] R. Li, X. Li, K. Hui, and C. Fu. SP-GAN: sphere-guided 3d shape generation and manipulation. *ACM Trans. Graph.*, 40(4):151:1–151:12, 2021. 1

[34] X. Liu, G. Yin, J. Shao, X. Wang, and H. Li. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 568–578, 2019. 2, 7, 9

[35] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9992–10002. IEEE, 2021. 3, 4

[36] A. Luo, Z. Zhang, J. Wu, and J. B. Tenenbaum. End-to-end optimization of scene layout. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 3753–3762. Computer Vision Foundation / IEEE, 2020. 1

[37] W. Luo, Y. Li, R. Urtasun, and R. S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4898–4906, 2016. 2

[38] Z. Lv, X. Li, Z. Niu, B. Cao, and W. Zuo. Semantic-shape adaptive feature modulation for semantic image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11214–11223, 2022. 1, 2, 7

[39] P. Mittal, Y.-C. Cheng, M. Singh, and S. Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 306–315, 2022. 1

[40] T. Park, M. Liu, T. Wang, and J. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346. Computer Vision Foundation / IEEE, 2019. 1, 2, 5, 7

[41] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 3, 11

[42] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 11

[43] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016. 2

[44] E. Schönfeld, V. Sushko, D. Zhang, J. Gall, B. Schiele, and A. Khoreva. You only need adversarial supervision for semantic image synthesis. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 2, 7, 9

[45] Z. Tan, M. Chai, D. Chen, J. Liao, Q. Chu, B. Liu, G. Hua, and N. Yu. Diverse semantic image synthesis via probability distribution modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7962–7971, 2021. 2

[46] Z. Tan, D. Chen, Q. Chu, M. Chai, J. Liao, M. He, L. Yuan, G. Hua, and N. Yu. Efficient semantic image synthesis via class-adaptive normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[47] H. Tang, L. Shao, P. H. Torr, and N. Sebe. Local and global gans with semantic-aware upsampling for image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 3

[48] H. Tang, D. Xu, Y. Yan, P. H. Torr, and N. Sebe. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7870–7879, 2020. 3, 7

[49] K. Tateno, N. Navab, and F. Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 707–722, 2018. 5

[50] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021. 3

[51] R. Wang, S. M. Pizer, and J.-M. Frahm. Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5555–5564, 2019. 3

[52] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 5

[53] Y. Wang, L. Qi, Y.-C. Chen, X. Zhang, and J. Jia. Image synthesis via semantic composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13749–13758, 2021. 3

[54] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to struc-

tural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. 9

[55] X. Wei, Z. Chen, Y. Fu, Z. Cui, and Y. Zhang. Deep hybrid self-prior for full 3d mesh generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5805–5814, 2021. 1

[56] C. Wen, Y. Zhang, Z. Li, and Y. Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1042–1051, 2019. 1

[57] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021. 3

[58] J. Xie, Y. Xu, Z. Zheng, S. Zhu, and Y. N. Wu. Generative pointnet: Deep energy-based learning on unordered point sets for 3d generation, reconstruction and classification. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 14976–14985. Computer Vision Foundation / IEEE, 2021. 1

[59] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, volume 11212 of *Lecture Notes in Computer Science*, pages 785–801. Springer, 2018. 1

[60] W. Yin, J. Zhang, O. Wang, S. Niklaus, L. Mai, S. Chen, and C. Shen. Learning to recover 3d scene shape from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 204–213, 2021. 1

[61] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, and B. Guo. Styleswin: Transformer-based gan for high-resolution image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11304–11314, 2022. 3

[62] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, and Z. Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IX*, volume 12354 of *Lecture Notes in Computer Science*, pages 519–535. Springer, 2020. 5

[63] L. Zhou, Y. Du, and J. Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 1

[64] P. Zhu, R. Abdal, Y. Qin, and P. Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5104–5113, 2020. 2, 7

[65] S. Zhu, G. Brazil, and X. Liu. The edge of depth: Explicit constraints between segmentation and depth. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 13113–13122. Computer Vision Foundation / IEEE, 2020. 3